

注意：

本书为有版权图书，仅供 CCF 内部使用， 请勿外传！

在此感谢我的 LP 和刚满月的儿子对我的支持，不用我洗尿布使我得以有时间为给大家带来本书。

By 朝露@CCF



Word
排版艺术

The Art of Word - Write and Typeset Large Scale Documents

本书适用于 Word 97 以及后续的任何版本



电子工业出版社
PHEI
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

侯捷著

内 容 简 介

Word 是一种功能强大的文字处理软件，并已获得广泛的应用，但业内人士分析：大约 80% 的用户只使用了约 20% 的软件功能，排版软件只作为打字软件在用。这是因为使用者对软件大都是外行，需要有人帮他们了解、掌握和使用软件所具有的一些比较深入的、高级的功能，使 Word 的排版达到更高质更高效的程度。

本书就是为以上目的而编写的。作者侯捷先生是台湾著名的电脑图书作家，已发表著作、译作数十种，他撰写本书有三大优势：①他是电脑软件专家，对 Word 有深入了解。②他亲自用 Word 排版制作了近 30 种图书，对 Word 排版可谓了如指掌。③侯捷先生文笔功底深厚，表述流畅生动，向来深受读者称许。侯捷先生还在书中安排了许多令人忍俊不禁的插图，为本书增添了不少阅读的趣味。

想摆脱大量繁琐易错事务的写家，想大幅度提高自己的工作能力，把排版工作做得更高质、高效的录入排版工作者，若购买并经常翻阅此书，我们预言，您必将大有收获！

图书在版编目（CIP）数据

Word 排版艺术 / 侯捷著. —北京：电子工业出版社，2004.10

ISBN 7-121-00421-6

I.W... II.侯... III.文字处理系统，Word IV.TP391.12

中国版本图书馆 CIP 数据核字（2004）第 100672 号

责任编辑：周筠 郑兆昭

技术协作：陈硕 方舟

责任校对：张兴田

印 刷：北京智力达印刷有限公司

出版发行：电子工业出版社

经 销：北京市海淀区万寿路 173 信箱 邮编 100036

开 本：各地新华书店

印 张：787×980 1/16 印张：27 字数：500 千字

印 次：2004 年 10 月第 1 次印刷

印 数：8 000 册 定价：48.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：(010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

献给所有文字工作者

本书助您解脱排版梦魇
亲手完成作品最终形貌

本书助您理解 Word 写作平台的种种设施
助您专注于创作内容，不为劳役所苦

艺术是什么？

艺术和工程仿佛在光谱两端
但是艺术无处不在
不带 dirty work 的工程，就是一种艺术

“艺术和工程”是两个完全不同的概念。艺术是通过创作、表达和感受来实现的，而工程则是通过设计、制造和操作来实现的。艺术强调的是情感、创意和表现力，而工程则强调的是功能、效率和实用性。艺术往往需要花费大量的时间和精力，而工程则相对比较快捷和高效。艺术可以是任何形式的表现形式，而工程则通常是指物质和技术方面的活动。

“不带 dirty work 的工程，就是一种艺术。”这句话强调的是，好的工程应该是美观、实用且富有创意的。好的工程不仅仅是技术上的突破，更是对美的追求和表达。好的工程应该能够带给人们愉悦的感受，而不是仅仅满足于功能性的需求。

“艺术是什么？”这个问题没有标准答案，每个人都有自己的理解和诠释。但无论如何，艺术都是人类文明的重要组成部分，我们应该尊重和珍惜它。

自序

我是一个程序员（programmer），也是一名资讯教育工作者。我的主要工作是钻研软件技术，尤其是编程（programming）技术。我的主要产出并非程序（programs），而是著作、文章、译作和课程。除了课程，前三宗产品都需要大量文字。这些文字欲以悦目的形象出现，就需要良好的编排。

打从 1990 开始走上资讯教育的路，我便是一个十分重视文字呈现的人——不仅在内容上，也在表现形貌上。写作生涯的前四五年，我对排版又爱又恨。因为对版式有很高的要求，所以爱；因为自己没有能力操控，所以恨。一而再三和排版人员反复沟通的过程中，我觉得自己像一条摇尾乞怜的狗，涎着脸哈着腰，惟恐触怒右手握鼠标、左手敲键盘、以飞快速度掌控极端繁复系统的不可一世的女皇（多半是女皇）。每次书稿完成，我都要进驻出版社 2~3 天，全天候服侍女皇们。有时心中暗暗着急（怎不多用点心思？这个还要我说吗？就不知道举一隅以反三隅？）。尽管如此，嘴上手上的殷勤却不敢怠慢。

这种周而复始摇尾乞怜的日子，虽不致于冲淡出版的喜悦，多少还是让我有不小的精神压抑。终于 1995 年爆发了“大事情”：U 公司出版经理不顾红字处处圈批点点的首校件，直接下厂印刷装订出书。托天之幸这些瑕疵品只流出 100 本，其余都在出库前一刻追回来了，并在我的目光下销毁。但是，三颗脑袋我也想不出来那是什么样的出版思维！

终于我告诉自己：够了，我要亲自排版。我不要我的书的任何一个字一张图的大小型式颜色粗细位置……不在我的完全掌控之下。我要保证我能够预先、精准、完整地看到我的读者将来在纸件上看到的每一样东西。

不再涎脸哈腰，不再提心吊胆……我要自己排版！



1995年末我开始学习排版。基于 Microsoft Word 的亲和性和普及性，一开始我就没做其他选择。但是我所找到的相关书籍，总是从很基本、很琐屑的操作开始。**Bottom-up**（由下而上的）教学法使我看到了树，却看不到林。虽然无树不成林，基本操作总是必要的，但琐屑的操作解决不了我的疑团：“一本书难道是这样那样一字一句调校出来的吗？排版的轮廓是什么？排版的领域知识是什么？”



很幸运我遇到一位好老师（请容我把对他的感谢延至“致谢”中表达）。他告诉我样式的概念，使我对排版有了初步的认识；又根据我的需求为我量身订做了一些版式设置（纸张、版心、裁切线、页眉等等），使我能够快速掌握排版的根本作业方式。虽然未臻全面，也够我三弯两绕地解决问题。



1996至今，我就以此为基础，完成了近30本书的排版，再没有将任何一部作品假手他人。

兹后，基于对排版的兴趣和需要，我逐渐补齐全面战技，执行正规作业：目录和索引的制作、页眉的多样设置、页面视图+文档结构图、VBA 编程……我也逐渐发现，Word 不仅用于排版，对大型文档的创作过程也有非常大的帮助，包括章节标题（headings）和图表题注（captions）的自动化编号，批注、修订等多人协作工具……

就像乐于分享编程（programming）技术一样，我也满心欢喜地于今年拨出时间，整理出个人使用 Word 协助创作和排版的心得。本书讨论的焦点是以书籍规模（或近似规模）为目标之大型文档。第0章“导读”将对本书的定位以及读者的应用基础作详细的说明。

不论是书籍、论文或任何书面报告，写作过程中看着自己的思绪思维、观点想法，从无形到有形，从有形到美好，从原始文字和图片逐渐凝聚成一个以个人意志为依归的形貌，是一件很有趣也很有成就感的事——我已经乐此不疲了十数年。虽然我不知道有多少人需要撰写（或编排）书籍（或论文）之类的大规模文档，虽然我不知道有多少人需要这样一本论述 Word 高级议题的书，虽然我不知道真正需要这种书的人能否看到这一本书……管它呢，生命中留下足迹，不枉一遭！



侯捷 2003/12/01 于台湾新竹

jjhou@jjhou.com

<http://www.jjhou.com> (繁体) <http://jjhou.csdn.net> (简体)

目录

章节目录

自序.....	v
目录.....	vii
章节目录.....	vii
表目录.....	xx
图目录.....	xxi
VBA 程序目录.....	xxvii
致谢.....	xxix
导读.....	1
Word 术语，英中繁简对照表	1
缘起.....	9
本书定位.....	11
读者基础.....	11
需要循序阅读吗？	11
为什么选择 Word + Acrobat	12
本书使用的软件版本.....	12
本书内容与软件版本相依吗？	13
本书未履踏之处.....	13
本书文字表达习惯.....	14
视窗控件之称谓：视窗、菜单、项目、图标、工具栏.....	16
文档组成之称谓：扉页、序言、目录、正文、附录、索引.....	17
页面元素之称谓：页眉、页脚、页码、正文、标题、版心.....	18
排版要件之称谓.....	20

1. 问答 101.....	23
关于出版 and 书籍制作流程	23
○ 出版流程是什么	23
○ 制版有哪几种做法？对品质有何影响？	23
○ 单色和多色印刷，制版上有什么差别？	24
○ 印刷版材有哪些种类？如何印刷？	25
○ 印刷版材的材质、保存期限和耗损因素为何？	25
○ 单色和多色，在印刷上有什么差别？	26
○ 印张是什么？对开版是什么意思？	26
○ 书帖是什么？	26
○ 纸厂供应的纸有多大？	26
○ 开本（16开、18开、32开……）指的是什么？	26
○ 各种开本的一块对开版可拼凑多少张书页？	27
○ 书籍尺寸与原始（排版系统中的）设置是否绝对准确相等？	27
○ 装订时裁剩的纸干什么用了？	27
○ 拼版是什么？	27
○ 出血是什么？	28
○ 重印和再版指的是什么？绝版是指什么？	28
关于 Word 写作平台	29
○ 写作平台（Writing Platform）是什么意义？	29
○ 何谓“在写作期间就带来帮助”？	29
○ 页次自动编号，Word 做得到吗？	29
○ 两个编页系统（i,ii,iii…和1,2,3…）共存于一份文档，可以吗？	29
○ 图表自动编号，Word 做得到吗？	29
○ 章节标题（heading）自动编号，Word 做得到吗？	30
○ 脚注（footnotes）自动编号，Word 做得到吗？	30
○ 还有什么自动编号是 Word 可以提供的？	30
○ 自动编号的东西一旦在文档中被移动位置，会自动重新编号吗？	30
○ 交叉引用（cross reference）怎么做？	30
○ 目录可以在写作过程中随时被看到并更新吗？	30
○ 索引可以在写作过程中随时被看到并更新吗？	30
○ 索引很难搞吗？	30
○ 文档各单元（例如各章）从奇数页开始，容易办到吗？	31
○ 还有哪些 Word 功能可以辅助写作？	31

关于 Word 排版概念与环境设置	31
○ 文字处理 (word processing) 和排版 (typesetting) 如何区分?	31
○ Word 是专业排版软件吗? 可独立实现一整本书的排版吗?	31
○ 彩色文档适合以 Word 完成吗?	32
○ 试比较各排版软件如 Word, PageMaker, Publisher, Quark XPress	32
○ 排版是某种 dirty work 吗? 是某种繁琐而重复劳动的工作吗?	32
○ Word 的【视图(V)】菜单下有许多视图, 做什么用?	33
○ 该把一整份文档置于同一文件, 或是另有其他选择?	33
○ 是否该为了加快 Word 反应速度而使用普通视图或大纲视图?	33
○ PDF 文档可透过导览窗格内的书签快速进入对应页面。Word 也可以 吗?	33
○ 长时间排版或阅读, 白底文档容易使眼睛疲劳。是否有其他选择?	34
○ 文字设为白色有什么后果? 可藉此设置某些文字不被打印出来吗?	34
○ 我可以获得文档的统计数据吗?	34
○ Word 有不少自动进行的字符串替换功能, 可否取消?	34
○ Word 有不少自动套用格式功能, 可否取消?	35
○ Word 的版本兼容性如何?	36
○ Word 文档结构图有时候表现失常!	36
○ Word 自动化章节标题编号有时候表现失常, 乱编!	36
○ 我的文件体积莫名其妙地暴涨!	37
○ 谈谈 T _E X 排版软件.....	37
关于 Word 排版实务	38
页面设置 (Page Setup) 与系统设置	38
○ 排版的第一件事是什么?	38
○ 节 (Sections) 是什么?	38
○ 模板 (Templates) 是什么? .dot 是什么?	39
○ normal.dot 是什么?	39
○ 页眉 页脚是什么? 天头地脚是什么?	39
○ 可否让每一章 (或说每一单元) 一定从奇数页开始?	40
○ 页码会自动流水排列吗, 即使任意安插或删除页面?	40
○ 文前是指什么? 正文 (内文) 是指什么? 文后是指什么?	40
○ 对文前采 i,ii,iii…编页, 对正文和文后采 1,2,3…编页, 可以吗?	41
○ 如何以单张 A4 纸打印两页, 对折后打钉成为一本小书?	41
文字与段落 (Texts & Paragraphs)	42
○ 格式 (Formats) 是什么?	42

○ 样式 (Styles) 是什么?	42
○ 好看的样式如何设计?	42
○ 有内建样式 (Built-in Styles) 可用吗?	43
○ 那么内建样式有何价值?	43
○ 我已自定许多样式 现在才知道内建样式的好处, 如何补救?	43
○ 可以删除内建样式吗?	44
○ 如何使用别人建好的样式?	44
○ 文字段落的缩进和制表位总是令我困惑!	44
○ 我可以看到 Word 为文档所加的分隔符号吗?	44
○ 目录 (Table of Contents) 可由 Word 自动制作吗?	44
○ 章目录可由 Word 自动制作吗?	45
○ 索引 (Indexes) 可以由 Word 自动制作吗?	45
○ 域 (Fields) 是什么?	45
图片 (Figures)	46
○ 图片如何进入文档之中?	46
○ 嵌入 (Embedded) 和链接 (Linked) 是怎么回事?	46
○ 图片无法随着文字移动, 怎么办!	46
○ 我无法精准定位图片!	46
○ 可否裁剪过大的图? 将来可复原吗?	46
○ 过多的图片会不会急速膨胀文件的体积, 拖垮 Word 的速度?	47
○ Word 会对图片自动进行压缩吗?	47
○ 可以制作图目录 (Table of Figures) 吗?	47
表格 (Tables)	47
○ 文字可以很方便地被转化为表格吗? 反之如何?	47
○ 我可以让表格的表头跨页重复吗?	47
○ 可以制作“非方阵、非对称”的表格吗?	47
○ 可以制作“表格目录”吗?	47
关于 Word 程序可控性	47
○ 宏 (Macros) 是什么?	47
○ 宏可以做些什么事?	48
○ 如何制作宏?	48
○ 宏如何启动? 可制成按钮便利使用吗?	48
○ 我写了个宏, 换一台电脑后却消失了!	48
○ VBA (Visual Basic for Application) 是什么?	48

○ VBA 很重要吗?	48
○ Word 对象模型 (Object Model) 又是什么?	48
关于 Word 电子文档	49
○ 有人使用 Word 电子文档来传布文档吗?	49
○ 书签 (Bookmarks) 是什么? 如何插入文档内?	49
○ 超链接 (Hyperlinks) 是什么? 如何插入文档内?	49
○ 批注 (Comments) 是什么? 如何插入文档内?	49
○ 追踪修订 (Track Changes) 是什么?	50
○ Word 是否支持多人协作? 是否提供“各种修订的历史记录”?	50
关于 PDF 便携式文档	50
○ PDF 是什么?	50
○ PDF 文档该如何展示? 如何阅读?	50
○ PDF 文档如何制作?	50
○ <i>PDFWriter</i> 是什么?	50
○ <i>PDF Distiller</i> 是什么?	50
○ <i>Adobe Acrobat</i> 是什么?	51
○ 透过 <i>Adobe Distiller</i> 产出 PDF 文档, 其形貌是否与原件相同?	51
○ <i>PDF Reader</i> 或 <i>Adobe Acrobat</i> 的导览窗格内的书签如何设置?	51
○ PDF 是便携式文档, 馈给网片输出机直接制版最好了。可以吗?	51
本书所有版式与样式细节	51
2. 总论	59
2.1. 排版, 被个人电脑完全取代的行业	59
我的第三文书中心经验 (1984-1985)	59
我的 90s 年代出书经验 (1991-1995)	60
我的 21 世纪出版作业模式 (2003-date)	62
2.2. 个人出版思潮	67
2.3. 排版基本概念	68
何谓专业排版	68
何谓便携式电子文档	68
何谓样式	68
何谓创作平台	70
2.3.1. 最佳作业模式	70
2.3.2. 文档结构视图	71

2.3.3. 页面视图.....	72
文档结构图 + 页面视图 = 创作者的好帮手	72
2.3.4. 大纲视图.....	74
2.3.5. Web 视图	76
2.3.6. 主控文档视图.....	77
2.3.7. 细说页面元素.....	79
2.3.8. 细说排版要件.....	83
2.3.9. 心得偶拾.....	84
一致性有多重要?	84
文档 (document) 与文件 (files)	85
页码最好采用流水号.....	86
尽量让页面信息丰富.....	86
快捷键的取舍.....	86
样式命名考虑.....	87
不要有洁癖.....	88
2.4. 版面美学之我见.....	88
2.4.1. 字号、字距、行距.....	88
2.4.2. 字体、色彩不要过于丰富.....	89
2.4.3. 斜体中文字的杀伤力.....	90
2.4.4. 中英文字体的调和搭配.....	91
2.4.5. 图片的大小.....	92
2.4.6. 不要每一页都出血.....	92
3. 页面设置.....	95
3.1. 纸张大小.....	95
3.2. 版心的大小和位置.....	96
3.3. 页眉与页脚的大小和位置.....	98
3.4. 节 (sections) 的概念.....	99
3.4.1. 单元永远始于奇数页.....	99
3.4.2. 多重页码系统.....	101
3.4.3. 奇偶页的页眉/页脚不同调	102
3.4.4. 自动编列行号.....	102
3.5. 出血.....	103
4. 页眉与页脚.....	105

4.1. 页眉/页脚的大小和位置	105
4.2. 页眉/页脚的内容设置	105
4.2.1. 奇数页页眉内侧的标题名.....	106
4.2.2. 偶数页页眉内侧的章名.....	107
4.2.3. 页脚右侧的书名.....	108
4.2.4. 页眉/页脚可放置交叉引用吗?	109
4.3. 插入页码.....	109
4.3.1. 同一份文档的不同页码格式.....	110
4.3.2. 页眉文字下的横线.....	110
5. 版心之内.....	113
5.1. 样式 (styles)	113
修改既有样式.....	114
样式 (Style) 涵盖的各种格式 (Format)	115
样式格式之一：字体.....	116
样式格式之二：段落.....	117
样式格式之三：制表位.....	119
样式格式之四：边框.....	121
样式格式之五：语言.....	123
样式格式之六：图文框.....	123
样式格式之七：编号.....	124
5.1.1. 新增段落样式.....	127
5.1.2. 新增字符样式.....	128
5.1.3. 内建样式 (Built-in Styles)	128
5.1.4. 样式与快捷键.....	130
5.1.5. 将样式套用于既有文字.....	130
5.2. 模板 (Templates)	132
关于模板 (Templates) 的一个实验.....	134
关于模板 (Templates) 的第二实验.....	135
5.2.1. 各种设置的栖身规则.....	136
5.2.2. Word 内建的模板	138
5.2.3. 万范之本：Normal.dot	140
5.2.4. 全局模板 (Global Templates)	140
5.2.5. 模板 (Templates) 的管理.....	142

5.3. 脚注 (Footnotes)	143
5.3.1. 插入脚注.....	143
5.3.2. 插入尾注.....	145
5.3.3. 脚注引用.....	145
5.3.4. 脚注文本.....	145
5.4. 多重栏位 (multi-columns)	146
5.5. 分隔设置.....	147
5.6. 项目 (清单; 列表; list)	149
5.7. 表格.....	150
5.7.1. 工具栏上的“插入表格”钮.....	150
5.7.2. 菜单上的“插入表格”项.....	151
5.7.3. 非“单纯方阵”表格.....	155
5.7.4. 手绘表格.....	156
5.8. 图片	157
5.9. 图文框.....	157
5.10. 题注 (Captions)	157
题注样式.....	160
图目录/表目录.....	160
修改题注标签.....	160
题注和标签的关系.....	161
5.11. 多层次样式/自动化章节编号	162
5.11.1. 初体验.....	163
5.11.2. 尝苦果.....	165
5.11.3. 山不转路转.....	166
5.11.4. 忍辱负重.....	168
5.11.5. 柳暗花明.....	169
5.12. 交叉引用 (Cross References)	172
5.13. 书签 (Bookmarks)	176
6. 图文结合	179
6.1. 绪论	179
各种图档格式的分类和特性.....	180
6.2. 插入图片	181
6.2.1. 剪贴画.....	182

6.2.2. 来自文件	184
6.2.3. 自选图形与几何绘图	184
6.2.4. 艺术字	185
6.2.5. 图表	186
6.3. 插入对象	187
链接 (Linking) 对象	188
6.4. 图片与对象的剪裁	188
6.5. 图片与对象的精准定位	190
6.6. 图片与对象的大小	197
6.7. 插入图文框	197
6.8. 文绕图	200
6.9. 覆盖于文档上的图片/对象	200
6.10. 图片对文档体积的影响——效率总检讨	201
7. 目录自动化	205
7.1. 藉由样式自动化产生目录	205
7.1.1. 自定目录形貌	207
7.2. 手工加入目录项	208
7.3. 更新目录	210
7.4. 切断目录链接，成为独立文字	210
7.5. 单章目录的制作	211
7.6. 图表目录的制作	214
8. 索引自动化	217
8.1. 索引相关概念	217
8.1.1. 索引项 (Index Entries)	218
XE 域的选项 (Options)	219
8.1.2. 索引主项 (索引第一层)	220
8.1.3. 索引次项 (索引第二层、第三层……)	220
8.2. 以“索引词条文件”批量标记索引项目	220
8.2.1. 索引词条文件的建立	221
8.2.2. 让 Word 自动化建立索引	222
重复建立索引，可以吗？	223
一份文档带有两份索引，可以吗？	224

8.2.3. 自定索引形貌.....	224
8.3. 逐一手工标记索引项.....	225
8.4. 索引佳例示范与讨论.....	227
8.5. 本书索引制作过程.....	229
9. 域.....	235
9.1. 所谓域 (Fields)	235
域的代码、结果、底纹.....	235
Word 域总览.....	239
9.2. 域的插入与更新.....	242
9.2.1. 插入域.....	242
9.2.2. 更新域.....	242
9.2.3. 显示或隐藏域代码.....	243
9.3. 本书使用的域.....	244
9.3.1. Page 域：目前页次.....	244
9.3.2. Section 域：目前节次.....	245
9.3.3. NumPages 域：文档总页数	245
9.3.4. NumChars 域：文档总字符数	245
9.3.5. NumWords 域：文档总字数.....	246
9.3.6. TOC 域：目录.....	246
9.3.7. TC 域：目录项	246
9.3.8. Index 域：索引	247
9.3.9. XE 域：索引项目	247
9.3.10. StyleRef 域.....	247
9.3.11. PageRef 域.....	247
9.3.12. Ref 域.....	248
9.3.13. Seq 域.....	248
9.4. StyleRef 域选项 (options)	249
10. 宏、VBA、Word 对象.....	253
10.1. 所谓宏 (macro)	253
10.1.1. 制作宏.....	254
宏的录制 (Record)	255
宏的新建 (Create)	257

Visual Basic 编辑器	258
宏、模块、工程、文档、模板.....	259
10.1.2. 播放宏.....	259
10.1.3. 宏与文档及模板的关系.....	260
制作宏之前先指定贴附（储存）处所.....	260
宏制作之后改变贴附处所（管理器）	261
10.1.4. 将宏制成按钮.....	261
10.1.5. 制作一个测试环境.....	264
10.2. 所谓 VBA	265
10.2.1. VBA 语法简介	265
注释（Comments）	265
子程序（SubRoutine）	265
函数（Function）	267
变量（Variable）	267
数据型别（Data Type）	267
流程控制（Flow Control）	270
开启文件（Open File）	271
读取文件内容（Read File）	272
关闭文件（Close File）	272
With、MsgBox()、CStr()、Chr().....	272
10.2.2. 自学 VBA Programming 的好办法	273
10.3. Word 对象（Objects）	275
10.3.1. 面向对象（Object Oriented）	275
10.3.2. Word 对象模型（Object Model）	279
With 语句（statement）	280
MsgBox 函数	280
CStr 函数	281
Chr 函数.....	281
10.3.3. 重要而常用的 Word 对象	281
Application.....	283
Documents/Document	284
Paragraphs/Paragraph	286
Sentences	289
Words	289

Characters	290
Sections/Section	291
Styles/Style	293
TablesOfContents/ TableOfContents	294
TablesOfFigures/ TableOfFigures	296
Range	297
Selection	297
Templates/Template	300
ListTemplates/ListTemplate	301
Find	303
Replacement	303
10.3.4. 自学 Word Programming 的好办法	305
10.4. 实例一：程序代码自动插入行号	305
10.4.1. 全自动加行号.....	305
10.4.2. 半自动加行号.....	307
10.4.3. 改变注释颜色.....	309
10.5. 实例二：两岸术语转换.....	311
10.5.1. 文本文件格式设计.....	313
10.5.2. 读取文本文件内容.....	315
10.5.3. 字符串处理.....	316
10.5.4. 以 查找和替换 对话框为师	319
10.5.5. 需要脑袋瓜的地方	319
10.6. 实例三：列出文档使用的所有字体	320
10.6.1. 关于 Collection.....	320
10.6.2. 搜寻所有 Style 所用的字体	324
10.6.3. 搜寻所有 Character 所用的字体	325
10.6.4. 后记：字体发生处	327
10.6.5. 列出所有样式的说明文字	328
11. Word 电子文档	331
11.1. 绪论	331
11.2. 超链接（Hyperlink）	332
超链接 vs. 交叉引用	334
11.3. 书签（Bookmark）	335
11.4. 批注（Comment）	338

11.5. 追踪修订 (Track Changes)	339
11.6. 电子文档的保护.....	343
11.6.1. 批注保护.....	344
11.6.2. 修订保护.....	345
11.6.3. 开启保护.....	345
11.6.4. 防写保护/只读保护	346
12. PDF 电子文档	349
12.1. 绪论.....	349
12.2. 便携式的条件.....	350
12.3. PDF 简介	351
12.4. PDF Reader.....	352
12.5. PDF Writer.....	352
12.6. PDF Distiller (蒸馏器)	352
12.6.1. 首选项.....	354
12.6.2. 字体位置.....	355
12.6.3. 作业选项：分辨率、压缩、字体、颜色.....	355
12.6.4. 安全性设置.....	357
12.6.5. 水印设置.....	358
12.6.6. 保留 Word 文档的书签、超链接、交叉引用	360
12.6.7. 一只臭虫.....	360
12.7. Adobe Acrobat.....	360
12.7.1. 安全性设置.....	362
12.7.2. 页面操作.....	362
插入页面.....	363
提取页面.....	364
取代页面.....	364
删除页面.....	364
裁剪页面.....	365
12.7.3. 编排页码.....	366
12.7.4. 书签+导览窗格	369
手工制作书签.....	369
自动化制作书签.....	372
更动页面会影响书签吗？	372

12.8. PDF Maker.....	373
A 参考资料	377
B 粗粮的细作	379
索引.....	385

表目录

表 0-1 Word 用语之英中繁简对照——按菜单排列.....	1
表 0-2 Word 用语之英中繁简对照——按字母次序排列.....	6
表 0-3 本书文字表现习惯	15
表 1-1 本书各种组成描述	51
表 1-2 本书页面设置与样式细目	53
表 2-1 细说页面元素	80
表 2-2 细说排版要件	83
表 2-3 斜体英文范例	90
表 2-4 斜体中文范例	91
表 5-1 若干 Word 内建样式的名称及用途	129
表 5-2 交叉引用 中的各种选项.....	175
表 6-1 图形文件格式与特性说明	180
表 6-2 图文相对关系的说明	191
表 9-1 Word 域一览表	239
表 9-2 域 StyleRef 的 \l,\r 选项意义	249
表 10-1 VBA 数据型别 (Data Types)	268
表 10-2 VarType 函数的返回值及其意义.....	269
表 10-3 VBA 字符串处理函数一览.....	316
表 11-1 超链接 vs.交叉引用	334
表 11-2 URL 与书签的相互关系	335
表 11-3 书签 vs.题注和章节编号	337
表 11-4 批注和追踪修订的比较	341
表 12-1 Acrobat Distiller 的作业选项 (Job Options)	357

图目录

本书这份“图目录”我个人认为实用价值不大；大概很少人会从这里作为找图的起点。摆出来只是为了示范 Word 的图目录制作能力。

图 0-1 Microsoft Windows 视窗元素	16
图 0-2 大型文档之各种组成单元	17
图 0-3 Word 所提供的页面元素；每一个都是 Word 的内建样式	18
图 0-4 Word 抽取标题成为目录，并以 目录 1~9 样式呈现之	19
图 0-5 Word 抽取索引项成为索引，并以 索引 1~9 样式呈现之	19
图 1-1 出版流程	24
图 1-2 制版流程	25
图 1-3 拼版示意图	28
图 1-4 出血示意图	28
图 1-5 页面布局：天头地脚、页眉、页脚、页码、订口、切口、版心	32
图 1-6 文档与视图的关系示意图	33
图 1-7 Word 的 字数统计 画面	34
图 1-8 Word 文字自动更正与替换（例如自动替换笑脸）	35
图 1-9 Word 文字自动套用格式（括号正确匹配、引号替换……）	36
图 1-10 宏和样式可储存于 (1)文档 (2)模板 (3) normal.dot	39
图 1-11 节 (sections) 示意图	40
图 1-12 令 A 样式继承 B 样式	43
图 1-13 列出所有样式（含内建样式）	44
图 2-1 文档结构图 + 页面视图：Word 最佳作业模式	70
图 2-2 文档结构图的右键（快捷）菜单	71
图 2-3 文件引导框 + 大纲视图	75
图 2-4 Word97 大纲工具栏，细部说明	75
图 2-5 文件引导框 + Web 视图	77
图 2-6 Word 2000 的大纲工具栏。右侧 7 个按钮给主控视图使用	77
图 2-7 文档结构图 + 主控文档视图 (in Word 2000)	78
图 2-8 被主控文档视图储存起来的主控文档和子文档	79
图 3-1 纸型（纸张大小）的设置	96
图 3-2 页边距的设置（Word2002 有更多功能，见 p.41）	96

图 3-3 页边距设置的数据意义	97
图 3-4 页眉实例画面	99
图 3-5 页脚实例画面	99
图 3-6 页眉和页脚 工具栏	99
图 3-7 页面设置 + 版式	100
图 3-8 分隔符 : 有换行、分页、分栏、分节四种	100
图 4-1 奇偶页页眉不同的情况	105
图 4-2 为页眉插入一个域	106
图 4-3 选择域的选项	107
图 5-1 样式 对话框	113
图 5-2 修改样式 。共有七种格式可供修改	114
图 5-3 【 格式 → 样式(S) 】 + 更改(M) + 字体(F) + 字体(N)	116
图 5-4 【 格式 → 样式(S) 】 + 更改(M) + 字体 + 字符间距	117
图 5-5 【 格式 → 样式(S) 】 + 更改(M) + 段落 + 缩进和间距	117
图 5-6 段落的间距和行高	118
图 5-7 缩进数据	118
图 5-8 【 格式 → 样式(S) 】 + 更改(M) + 制表位	119
图 5-9 【 格式 → 样式(S) 】 + 更改(M) + 边框和底纹 + 边框(B)	121
图 5-10 【 格式 → 样式(S) 】 + 更改(M) + 边框和底纹 + 底纹(S)	122
图 5-11 【 格式 → 样式(S) 】 + 更改(M) + 语言	123
图 5-12 【 格式 → 样式(S) 】 + 更改(M) + 图文框	123
图 5-13 【 格式 → 样式(S) 】 + 更改(M) + 项目符号和编号 + 项目符号(B)	124
图 5-14 【 格式 → 样式(S) 】 + 更改(M) + 项目符号和编号 + 编号(N)	125
图 5-15 新建一个段落样式	127
图 5-16 新增一个字符样式	128
图 5-17 自定义键盘	130
图 5-18 将.doc 另存为.dot	133
图 5-19 模板 (templates) 与文档 (documents) 的关系	134
图 5-20 模板 (Template) 实验之一	135
图 5-21 模板 (Template) 实验之二	136

图 5-22 模板集中地，在 Word 中观察.....	139
图 5-23 模板集中地，以资源管理器观察.....	139
图 5-24 模板和加载项 (Templates and Add-ins)	141
图 5-25 管理器可以管理模板 (templates) 内的各种设置	142
图 5-26 脚注和尾注 之一.....	143
图 5-27 脚注和尾注 之二.....	144
图 5-28 注释选项 + 所有脚注(F).....	144
图 5-29 注释选项 + 所有尾注(E).....	144
图 5-30 分隔符.....	147
图 5-31 表格自动设置 提供许多现成的表格形式.....	151
图 5-32 【表格】菜单提供有许多表格调整功能.....	152
图 5-33 将文字转换成表格.....	153
图 5-34 拆分单元格.....	155
图 5-35 插入题注.....	157
图 5-36 题注标签、题注数字、题注文字.....	158
图 5-37 Word 要求用户建立一个“链接至标题样式的多级列表”.....	159
图 5-38 题注与章节（标题）自动编号的关系.....	163
图 5-39 选择一个多级符号型式，令 Word 做出多级 章节 编号	164
图 5-40 自定义多级符号列表	164
图 5-41 页眉没有出现自动化编号（这不理想）	166
图 5-42 章序号	168
图 5-43 交叉引用，对创作过程有极大帮助	174
图 5-44 用户可选择的交叉引用类型（内建或自定义均可）	174
图 5-45 可选择插入题注所含的信息及其所在页码.....	174
图 6-1 Microsoft 剪辑库	182
图 6-2 多媒体项目（剪贴画、图片、声音、影片）属性	182
图 6-3 插入图片	184
图 6-4 【插入→对象(O)】+ 新建(C)	187
图 6-5 【插入→对象(O)】+ 由文件创建(F)	188
图 6-6 裁剪图片/对象	190

图 6-7 设置自选图形格式 + [位置] 可精确定位一张浮贴图	193
图 6-8 设置对象格式 + [位置]	194
图 6-9 调整图片（对象）的大小	197
图 6-10 [图文框] 对话框	199
图 6-11 设置图片格式 + [环绕]	201
图 7-1 [索引和目录] + [目录]	206
图 7-2 [目录选项]，默认的“用以制成目录”的样式	206
图 7-3 [目录选项]，改设另一些“用以制成目录”的样式	207
图 7-4 [目录 1~目录 9] 样式，观察与更改	207
图 7-5 插入 TC 域	208
图 7-6 更新域	210
图 7-7 Word 询问以何种方式更新目录	210
图 7-8 章目录的制作示意	213
图 8-1 索引和目录	222
图 8-2 索引项目（XE）的标记（mark）	226
图 9-1 选择显现域的代码或结果，以及底纹	236
图 9-2 域代码（部分）。不分大小写	236
图 9-3 域结果	237
图 9-4 【文件→属性(I)】 + [摘要信息]	238
图 9-5 【文件→属性(I)】 + [统计信息]	238
图 9-6 【工具→选项(O)】 + [用户信息]	239
图 9-7 【插入→域】	244
图 9-8 域 PageRef、Ref、Seq 的用途	248
图 9-9 域 StyleRef 的 6 个选项	249
图 9-10 本书运用 StyleRef/r 于奇数页页眉	250
图 10-1 【工具→宏(M)】	255
图 10-2 宏录制器	255
图 10-3 新建一个宏	257
图 10-4 VBA 程序编辑视窗	257
图 10-5 VBA 编辑器	258
图 10-6 宏可被置于三个地方（共同模板、文档模板、文档）	259

图 10-7 执行某个宏	260
图 10-8 管理器。可管理样式、自动图文集、工具栏、宏	261
图 10-9 自定义一个工具栏，步骤 1	262
图 10-10 自定义一个工具栏，步骤 2	262
图 10-11 自定义 工具栏 和 命令	263
图 10-12 将自定义工具栏中的“文字按钮”改为“图片按钮”	264
图 10-13 查找和替换 对话框	274
图 10-14 类 (classes) 与对象 (objects) 的关系	276
图 10-15 VBA 和应用程序 (applications) 的关系	279
图 10-16 Word 对象的逻辑组织	282
图 10-17 Application 对象的 properties (局部列出)	283
图 10-18 Document 对象的 properties (局部列出)	285
图 10-19 Paragraph 对象的 properties (局部列出)	286
图 10-20 解释 paragraphs, sentences, words, characters	287
图 10-21 Sentences 对象的 properties	289
图 10-22 words 对象的 properties	289
图 10-23 Characters 对象的 properties	290
图 10-24 Section 对象的 properties (局部列出)	293
图 10-25 Style 对象的 properties (局部列出)	294
图 10-26 TableOfContens 对象的 properties	295
图 10-27 TableOfFigures 对象的 properties	297
图 10-28 Range 对象的 properties (局部列出)	298
图 10-29 Selection 对象的 properties (局部列出)	298
图 10-30 Template 对象的 properties (局部列出)	300
图 10-31 ListTemplate 对象的 properties	303
图 10-32 Find 对象的 properties (局部列出)	304
图 10-33 Replacement 对象的 properties (局部列出)	304
图 10-34 找出文档内的每一整块! 程序代码	306
图 10-35 重新设置一个 Range 对象的涵盖范围	310
图 10-36 简繁中文互转	313
图 10-37 中英繁简术语对照表 文件格式一	314
图 10-38 中英繁简术语对照表 文件格式二	314
图 10-39 遍历所有 Style 对象，获得的所有字体名称	325

图 10-40 遍历所有 Character 对象，获得的所有字体名称	327
图 11-1 超链接自动调用 email 软件，并填妥正确的收件者	333
图 11-2 插入超级链接（hyperlink）	333
图 11-3 图表题注其实就是一种书签	336
图 11-4 书签 对话框	336
图 11-5 批注视窗出现了（下侧视窗）	338
图 11-6 带有批注的文句，以黄底纹展现	339
图 11-7 追踪修订实例画面	340
图 11-8 追踪修订 对话框	340
图 11-9 接受或拒绝修订 对话框	341
图 11-10 设置个人的追踪修订记录显现风格	342
图 11-11 Word 文档版本比较	343
图 11-12 文档保护：修订保护、批注保护、窗体保护	344
图 11-13 文档保护之密码确认	344
图 11-14 文档保护的解除	345
图 11-15 【文件→另存为】+ 工具(L) +常规选项	346
图 11-16 文档之开启密码	346
图 11-17 防写（只读）模式之解除	346
图 12-1 Acrobat Distiller 主画面	353
图 12-2 Acrobat Distiller 首选项	354
图 12-3 Distiller 的【设置(S)】菜单	355
图 12-4 作业选项 分为 一般 、 压缩 、 字体 、 颜色 、 高级 五大类	356
图 12-5 Acrobat Distiller 的 安全性 对话框	357
图 12-6 在 Acrobat Distiller 打印机中设置水印	359
图 12-7 新水印	359
图 12-8 Acrobat 主画面，左为导览窗格	361
图 12-9 Acrobat 主画面，左为小型缩图	361
图 12-10 Acrobat 的 标准安全 对话框	362
图 12-11 Acrobat 裁剪功能所造成的裁剪线	367
图 12-12 Acrobat 的 编排页码 对话框	368
图 12-13 图左全部是阿拉伯数字编页，图右前 10 页为罗马数字编页	368

图 12-14 PDF 书签之手动形成.....	370
图 12-15 添加新书签，例一	371
图 12-16 添加新书签，例二	371
图 12-17 删除某个书签	371
图 12-18 Acrobat 在 Word 身上加装的菜单，for PDFMaker.....	373
图 12-19 PDFMaker for Microsoft Office, Office	374
图 12-20 PDFMaker for Microsoft Office, 书签	375
图 12-21 本书 PDF 文档画面一瞥.....	376

VBA 程序目录

VBA10-1 宏，将文档中的所有 "软体" 替换为 "软件"	275
VBA10-2 Application 对象.....	284
VBA10-3 Document 对象	286
VBA10-4 Paragraph 对象	287
VBA10-5 剖析 paragraphs、sentences、words 和 characters.....	288
VBA10-6 观察 Section 对象和 PageSetup 对象	291
VBA10-7 剖析 Section 对象（执行结果写于注释内）	292
VBA10-8 列出文档的所有样式.....	293
VBA10-9 剖析 TableOfContents 对象.....	295
VBA10-10 剖析 TableOfFigures 对象.....	296
VBA10-11 剖析 Template 对象	300
VBA10-12 剖析 ListTemplate 对象	302
VBA10-13 示范 Find 对象和 Replacement 对象	303
VBA10-14 找出文档内的每一区块！ 程序代码	306
VBA10-15 加行号，错误示范.....	307
VBA10-16 加行号，正确示范.....	309
VBA10-17 加行号，并将注释设为蓝色.....	311
VBA10-18 将读入的一整行文字切分为三个语汇	319
VBA10-19 判断某物是否存在于 Collection 之中	324
VBA10-20 遍历文档内含的所有 Style 对象，取其 Font 对象	325
VBA10-21 遍历文档内含的所有 Characters 对象，取其 Font 对象.....	326
VBA10-22 找出字体发生地点，并跳至该处.....	328
VBA10-23 将所有使用样式 (inUse Style) 的细目写入文档最尾端	329



致谢

首先要感谢老朋友张钧杰。他是第一位告诉我排版怎么回事的人，并且在我还不很熟手熟脚的时候，为我做了一些模板（又称样式库），让我在还没有全面掌握 Word 排版精髓的时候，就能够上战场，使我有机会开创多年来被读者昵称“侯式风格”的计算机书籍页面版式。同时免除了我对书籍后制作不确定性的焦虑，解我于倒悬。

其次要感谢老朋友薛晓岚。她撰写《Word 的使用艺术》三个版本，教导我不少操作手法。晓岚历练了年轻人罕见的经历，展现了年轻人罕见的胆识和作为。如今远走英国，令我怀念。

再其次要感谢老朋友黄俊尧。他告诉我 Word 域的设置，开启了我对 Word 的另一层认识和其后的许多相关乐趣。虽然在他只是随口一提，或许早就忘了，但我没忘！

感谢合作多年的暮峰图书公司。我提出这个出人意表的写作计划后，他们并没有额头冒汗脸上三条线，而是很爽快地同意出版，并允诺双色印刷（我不需要四色，双色就够了），一如多年来对我的信任和纵容。暮峰老伙伴 hanson、tony、daphne、cathy、lay、novia……为我提供了书籍后制和作者庶务的完美服务，使我欢喜地将“与出版公司打交道”感受栏上的记录从“畏途”改为“享受”。有一个干练而融洽的后勤支持，实在是创作路上的一大幸福。我还记得那个由暮峰专门为我安排的印刷之旅，让我真正有机会看到打样、输出、制版、印刷、裁切、装订、仓储等种种过程，得以亲身感受一本书籍的后制工序是多么繁复。

感谢 Word 随附的 *On-line Help*（联机帮助文档）。当我钻研一些深入的、不常见的主题时，*On-line Help* 几乎是我惟一的帮助——尽管它的 *hyperlink*（超级链接）组织方式不若书籍那般严谨易读，不少高级议题的说明相当不够，用词也过于艰硬，我必须反复推敲再三实验。但“惟一”毕竟珍贵，有线索胜过没头绪。我还是由衷感谢制作那么庞大说明文档的一群人。

感谢好友凤娇作为我的印刷知识来源。本书第1章谈到的分色、输出机、网片、印刷版、印刷机、纸张、开本……，全都是她悉心指导下的结果。

感谢好友百莉帮我截取Word英文版菜单画面，使我得以确认Word各种用语的英文说法（我的机器上并没有安装英文版Word）。

感谢好友陈硕帮我截取Word简体中文版菜单画面，使我得以确认Word各种用语的大陆说法。陈硕并就若干高级议题与我做了讨论，带给我相当的启示和帮助。

感谢好友大伟告诉我人体肌肉的学名。大伟是ENT（耳鼻喉）主任，但我猜想他更喜欢做个电脑专家或作家。人体肌肉和Word有什么关系吗？没有关系，只是为了叙述一段个人历史。猜猜这些肌肉学名出现在本书什么地方？☺

感谢侄儿冠宇。他从12岁起便只身从台北搭车来新竹跟着我学习C++编程语言，如今已能以饶富创意的思考来解决问题，每每令我惊艳。本书第10章（宏和VBA程序）得力于他的启发甚多。

感谢电子工业出版社/博文视点出版团队在本书简体版后制作过程中积极、不畏难、不落窠臼的工作态度。在每次任务中追求突破和提升，是优秀团队的表征。我的伙伴们呈现了这个特征。

感谢试读本书的众多好友，您们给了我许多宝贵意见。感谢来信询问排版问题的众多网友，您们给了我许多写作灵感。

最后，感谢内人美静，见证我的十数年写译历程，与我共同经历这条路上的每一分喜乐与哀愁。除了《无责任书评》，这本《Word排版艺术》是她惟一看得懂的我的作品。很公平，我也看不懂她的豆芽菜！本书首先让美静看懂然后出版，这使我对本书“妇孺皆解”充满了信心。看完本书（电子稿）后她信心满满地要帮我排版，但是…呃…我不需要了，经过这番磨练，加上先前的功力，我已经能够将创作和排版轻松融于一炉，毕全功于一役，让创作成为我人生路上的绝大享受☺



导读.....	1
Word 术语, 英中繁简对照表	1
缘起.....	9
本书定位.....	11
读者基础.....	11
需要循序阅读吗?	11
为什么选择 Word + Acrobat	12
本书使用的软件版本.....	12
本书内容与软件版本相依吗?	13
本书未履踏之处.....	13
本书文字表达习惯.....	14
视窗控件之称谓: 视窗、菜单、项目、图标、工具栏.....	16
文档组成之称谓: 靡页、序言、目录、正文、附录、索引.....	17
页面元素之称谓: 页眉、页脚、页码、正文、标题、版心.....	18
排版要件之称谓.....	20

O

导读

Word 术语，英中繁简对照表

这个对照表有没有制作的必要？我想有必要！有些日常用语如目录、索引、章节、正文、段落……，大家都知道意义。有些 Word 用语如页眉、页脚、脚注……，大致还可以猜而知之。但您知道什么是样式、格式、题注、标签、交叉引用、目录项、尾注、制表位……的精确意义吗？英文或许有益于您从另一个方向来理解。

这个对照表有没有必要放在这里？我想有必要！它（如果说有的话）通常会被放在附录，但谁看书先看附录呢？我希望读者第一眼就能先看看这个表。

本书出版繁简两式，所以我把两岸的 Word 用语一并整理下来。两岸交往日益频繁，不定哪一天您就需要与对岸朋友沟通②。

这个词汇对照表将以两种方式呈现：(1) Word 菜单次序（表 0-1）和 (2) 英文字母排列次序（表 0-2）。

* 以下列出 Word 菜单（某些与本书无关的菜单项并未列出，例如邮件、信封打印等等）。由其引发的对话框内的词汇亦列于其下（内缩形式）

表0-1 Word 用语之英中繁简对照——按菜单排列

Word 英文版用语	Word 繁体版用语	Word 简体版用语
File	档案	文件
New	开新档案	新建
Open	开启旧档	打开
Close	关闭	关闭
Save	储存档案	保存
Save As	另存新档	另存为

Save As Web page	另存成 Web 画面	另存为 Web 页
Versions	版本	版本
Web Page Preview	Web 画面预览	Web 页预览
Page Setup	版面设定	页面设置
Margins	边界	页边距
Pages	页数	页码范围
Multiple pages	多页	多页 (only in Word2002+)
Mirror Margins	左右对称	对称页边距
2 Pages per sheet	单面双页	拼页
Book fold	书籍对页	书籍折页
Reverse book fold	反向书籍折页	反向书籍折页
Layout	配置	版式
Paper Size	纸张大小	纸型
Portrait	纵向	纵向
Landscape	横向	横向
Paper Source	纸张来源	纸张来源
Print Preview	列印预览	打印预览
Print	列印	打印
Send To	传送到	发送
Properties	属性	属性
Exit	结束	退出

Edit	编辑	编辑
Undo Typing	复原键入	撤消键入
Repeat Typing	重复键入	重复键入
Cut	剪下	剪切
Copy	复制	复制
Paste	贴上	粘贴
Paste Special	选择性贴上	选择性粘贴
Paste as Hyperlink	贴上超连结	粘贴为超级链接
Clear	清除	清除
Select All	全选	全选
Find	寻找	查找
Replace	取代	替换
Go To	到	定位
Update IME Dictionary	重新组字	汉字重组
Links	连结	链接
Object	物件	对象

View	检视	视图
Normal	标准模式	普通
Web Layout	Web 版面配置	Web 版式
Print Layout	整页模式	页面
Outline	大纲模式	大纲
ToolBars	工具列	工具栏
Ruler	尺规	标尺

Show Paragraph Marks	显示段落标记	显示段落标记
Gridlines	格线	网格线
Document Map	文件引导模式	文档结构图
Header and Footer	页首/页尾	页眉和页脚
Footnotes	注脚	脚注
Markup	注解	批注
Full Screen	全萤幕	全屏显示
Zoom	显示比例	显示比例

Insert	插入	插入
Break	分隔设定	分隔符
Page Numbers	页码	页码
Date and Time	日期及时间	日期和时间
AutoText	自动图文集	自动图文集
Field	功能变数	域
Symbol	符号	符号
Comment	注解	批注
Number	数字	数字
Footnote	注脚	脚注
Endnote	章节附注	尾注
Caption	标号	题注
Label	标签	标签
Cross reference	交互参照	交叉引用
Index and Table	索引及目录	索引和目录
Mark Entry	标设项目	标记索引项
Auto Mark	自动标设	自动标记
Indented	阶层	缩进式
Run-in	等阶	接排式
Table Of Contents	目录	目录
Table Of Figures	图表目录	图表目录
Picture	图片	图片
ClipArt	美工图案	剪贴画
From file	从档案	来自文件
Link to file	连结到档案	链接到文件
Auto Shapes	快取图案	自选图形
WordArt	文字艺术师	艺术字
Chart	图表	图表
Text Box	文字方块	文本框
File	档案	文件
Object	物件	对象
Display as icon	以图示显示	显示为图标
Bookmark	书签	书签
Hyperlink	超连结	超级链接

Format	格式	格式
Font	字型	字体
Character Spacing	字元间距	字符间距
Underline	底线	下划线
Regular	标准	常规
Bold	粗体	加粗
Italic	斜体	倾斜
Superscript	上标	上标
Subscript	下标	下标
Strikethrough	删除线	删除线
Outline	外框字	空心
Emboss	浮凸	阳文
Engrave	雕刻	阴文
Shadow	阴影	阴影
Paragraph	段落	段落
Indents & Spacing	缩排及间距	缩进和间距
Line and Page Breaks	行与分隔设定	换行和分页
Alignment	对齐方式	对齐方式
Outline level	大纲阶层	大纲级别
Indentation Left	左边缩排	缩进, 左
Indentation Right	右边缩排	缩进, 右
Line Spacing	行距	行距
Bullets and Numbering	项目符号及编号	项目符号和编号
Bulleted	项目符号	项目符号
Numbered	编号	编号
Outline Numbered	大纲编号	多级符号
	自订大纲编号方式	自定义多级符号列表
	将层次连结至样式	将级别链接到样式
	法律样式编号	正规形式编号
	ListNum 栏位清单名称	ListNum 域列表名
Borders and Shading	框线及网底	边框和底纹
Columns	栏	分栏
Tabs	定位点	制表位
Drop Cap	首字放大	首字下沉
Text Direction	直书/横书	文字方向
Change Case	大小写转换	更改大小写
Background	背景	背景
Theme	主题	主题
Frames	框架	框架
AutoFormat	自动格式设定	自动套用格式
Styles	样式	样式
Default	预设	默认
Shortcut Key	快速键	快捷键
	组合管理	管理器
Object	物件	对象

Tool	工具	工具
Spelling and Grammar	拼字及文法检查	拼写和语法
Language	语言	语言
Word Count	字数统计	字数统计
Auto Summarize	自动摘要	自动编写摘要
Track Changes	追踪修订	修订
Protect Document	保护文件	保护文档
Online Collaboration	线上共同作业	联机协作
Macro	巨集	宏
Record New Macro	录制新巨集	录制新宏
Visual Basic Editor	Visual Basic 编辑器	Visual Basic 编辑器
Templates and Add-Ins	范本与增益集	模板和加载项
Customize	自订	自定义
Options	选项	选项

Table	表格	表格
Draw Table	手绘表格	绘制表格
Insert	插入表格	插入
Column	栏	列
Row	列	行
Delete	删除储存格	删除
Select	选取	选定
Merge Cells	合并储存格	合并单元格
Split Cells	分割储存格	拆分单元格
Split Table	分割表格	拆分表格
Table AutoFormat	表格自动设定	表格自动套用格式
AutoFit	最适列高	自动调整
Heading Rows Repeat	跨页标题重复	标题行重复
Convert	转换	转换
(Word97: 文字转换为表格)		
Sort	排序	排序
Formula	公式	公式
Hide Gridlines	隐藏格线	隐藏虚框
Table Properties	表格内容	表格属性

Window	视窗	窗口
New Window	开新视窗	新建窗口
Arrange All	并排显示	全部重排
Split	分割	拆分

Help	说明	帮助
Online Help	线上说明	联机帮助

※ 以下列出 Word 词汇。按字母次序排列。

表0-2 Word 用语之英中繁简对照——按字母次序排列

Alignment	对齐方式	对齐方式
Arrange All	并排显示	全部重排
Auto Mark	自动标设	自动标记
Auto Shapes	快取图案	自选图形
Auto Summarize	自动摘要	自动编写摘要
AutoFit	最适列高	自动调整
AutoFormat	自动格式设定	自动套用格式
AutoText	自动图文集	自动图文集
Background	背景	背景
Bold	粗体	加粗
Bookmark	书签	书签
Borders and Shading	框线及网底	边框和底纹
Break	分隔设定	分隔符
Built-in	内建	内置
Bulleted	项目符号	项目符号
Bullets and Numbering	项目符号及编号	项目符号和编号
Caption	标题	题注
Change Case	大小写转换	更改大小写
Character Spacing	字元间距	字符间距
Chart	图表	图表
Clear	清除	清除
ClipArt	美工图案	剪贴画
Close	关闭	关闭
Column	栏	列
Columns	栏	分栏
Comment	注解	批注
Convert	转换	转换
Copy	复制	复制
Cross reference	交互参照	交叉引用
Customize	自订	自定义
Cut	剪下	剪切
Date and Time	日期及时间	日期和时间
Default	预设	默认
Delete	删除储存格	删除
Display as icon	以图示显示	显示为图标
Document Map	文件引导模式	文档结构图
Draw Table	手绘表格	绘制表格
Drop Cap	首字放大	首字下沉
Edit	编辑	编辑
Emboss	浮凸	阳文
Endnote	章节附注	尾注
Engrave	雕刻	阴文

Exit	结束	退出
Field	功能变数	域
File	档案	文件
Find	寻找	查找
Font	字型	字体
Footnote	注脚	脚注
Format	格式	格式
Formula	公式	公式
Frames	框架	框架
From File	从档案	来自文件
Full Screen	全萤幕	全屏显示
Go To	到	定位
Gridlines	格线	网格线
Header and Footer	页首/页尾	页眉和页脚
Heading Rows Repeat	跨页标题重复	标题行重复
Help	说明	帮助
Hide Gridlines	隐藏格线	隐藏虚框
Hyperlink	超连结	超级链接
Indentation Left	左边缩排	缩进 左
Indentation Right	右边缩排	缩进 右
Indented	阶层	缩进式
Indents & Spacing	缩排及间距	缩进和间距
Index and Table	索引及目录	索引和目录
Insert	插入	插入
Insert	插入表格	插入
Italic	斜体	倾斜
Label	标签	标签
Landscape	横向	横向
Language	语言	语言
Layout	配置	版式
Letter Wizard	信件精灵	信函向导
Line and Page Breaks	行与分隔设定	换行和分页
Line Spacing	行距	行距
Link to file	连结到档案	链接到文件
Links	连结	链接
Macro	巨集	宏
Mail Merge Wizard	邮件合并精灵	中文信封向导
Margins	边界	页边距
Mark Entry	标设项目	标记索引项
mark	标设	标记
Markup	注解	批注
Merge Cells	合并储存格	合并单元格
New	开新档案	新建
New Window	开新视窗	新建窗口
Normal	标准模式	普通
Number	数字	数字

Numbered	编号	编号
Object	物件	对象
Online Collaboration	线上共同作业	联机协作
Online Help	线上说明	联机帮助
Open	开启旧档	打开
Options	选项	选项
Outline	外框字	空心
Outline	大纲模式	大纲
Outline level	大纲阶层	大纲级别
Outline Numbered	大纲编号	多级符号
Page Numbers	页码	页码
Page Setup	版面设定	页面设置
Paper Size	纸张大小	纸型
Paper Source	纸张来源	纸张来源
Paragraph	段落	段落
Paste	贴上	粘贴
Paste as Hyperlink	贴上超连结	粘贴为超级链接
Paste Special	选择性贴上	选择性粘贴
Picture	图片	图片
Portrait	纵向	纵向
Print	列印	打印
Print Layout	整页模式	页面
Print Preview	列印预览	打印预览
Properties	属性	属性
Protect Document	保护文件	保护文档
Raw	列	行
Regular (Font)	标准	常规
Repeat Typing	重复键入	重复键入
Replace	取代	替换
Ruler	尺规	标尺
Run-in	等阶	接排式
Save	储存档案	保存
Save As	另存新档	另存为
Save As Web page	另存成 Web 画面	另存为 Web 页
Select	选取	选定
Select All	全选	全选
Send To	传送到	发送
Shadow	阴影	阴影
Shortcut Key	快速键	快捷键
Show Paragraph Marks	显示段落标记	显示段落标记
Sort	排序	排序
Spelling and Grammar	拼字及文法检查	拼写和语法
Split	分割	拆分
Split Cells	分割储存格	拆分单元格
Split Table	分割表格	拆分表格
Strikethrough	删除线	删除线

Styles	样式	样式
Subscript	下标	下标
Superscript	上标	上标
Symbol	符号	符号
Table	表格	表格
Table AutoFormat	表格自动设定	表格自动套用格式
Table Of Contents	目录	目录
Table Of Figures	图表目录	图表目录
Table Properties	表格内容	表格属性
Tabs	定位点	制表位
Templates and Add-Ins	范本与增益集	模板和加载项
Text Box	文字方块	文本框
Text Direction	直书/横书	文字方向
Theme	主题	主题
Tool	工具	工具
ToolBars	工具列	工具栏
Track Changes	追踪修订	修订
Underline	底线	下划线
Undo Typing	复原键入	撤消键入
Update IME Dictionary	重新组字	汉字重组
Versions	版本	版本
View	检视	视图
Web Layout	Web 版面配置	Web 版式
Web Page Preview	Web 画面预览	Web 页预览
Window	视窗	窗口
WordArt	文字艺术师	艺术字
Word Count	字数统计	字数统计
Zoom	显示比例	显示比例

缘起

过去七八年来我以 Microsoft Word 为工具，排版了数百篇文章、数十本书籍，并搭配 Acrobat Distiller + Acrobat 制作了几乎相同册数的完整 PDF 电子书（有些开放，有些是私房品）。近年来写作和排版同步进行，又运用 Word 的文档结构图、自动化图表题注、交叉引用、域、宏……，为写作过程引入架构与管理上的协助，降低琐屑事务。这些美好的经验使我亟思整理成书，与大家分享，祈愿对于“已有若干 Word 运用基础，希望以 Word 完成大型文档排版，或使用 Word 作为创作平台”的人带来帮助。

大部分 Word 书籍皆以 **bottom-up**（由下而上）的方式进行教学：在各种基本而繁琐的操作，以及为吸引初学者而添加的各种轻松话语和插科打诨之后，篇幅轻松凌驾 500 页。（这是目的吗？）其中真正的排版（而非文字处理）篇幅往往寥寥。寥寥的篇幅要交待系统性、架构性甚强的排版观念和操作手法，以及创作平台的种种贴心设计，极为困难。有些 Word 书籍的专业排版篇幅比例稍微高些，但因采取 bottom-up 教学，排版的本质作业方式很容易被掩盖于枝枝节节之中。

当然，并不是每个人都需要专业排版或大型文档创作，也不是每个人都愿意接受“根本性的排版作业方式”。如果您每次工作份量仅只 15 页（有些伟人的容忍度是 30 页），那么就算创作过程天翻地覆，排版过程逐字沧桑，咬咬牙，dirty work 一样完成任务，并非定得先学会“专业排版”和“根本作业方式”才能展开工作。

但如果您的文档超过 15 页（或 30 页），又需要正经八百的章节目录、图表目录、索引、页眉、脚注……那么即使一生一次（例如毕业论文），我想这本书还是对您有很大价值。

本书定位在“两天内让您完全认识排版领域知识，以及 Word 的能与不能”。如果您因为知道过去的工作多么 dirty and mess，或因为知道未来的创作及排版多么 clean and neat，因而痛哭流涕，不论是感伤的泪还是感激的泪，我都会感到骄傲。

话说回来，不论这本书被多少人看了、读了、理解了，我都会因为写作的单纯快乐与发表的美好回忆，而开心，而荣耀。

本书定位

这是一本谈论长文档（书籍、论文、大型文书报告）之创作平台与排版的书籍。我把它当做一份心得整理，并不企图成为繁多问题的解答。我不敢说我能够告诉您，欲达成某种效果，Word 提供的所有方法或最佳做法，我只能说我可以提供多年实际经验。这些经验容或有所局限（well, 谁知道呢），但够我完成 30 本书籍的排版，包括您手上这一本！

四种人可以受益于本书：一是需要进行大型文档排版的人，二是需要进行大型文档创作的人（这一类需求常被忽略），三是需要团队协同写作（例如论文或技术文档或产品文档）的人（这一类需求也常被忽略），四是需要制作便携式电子文档的人。

本书定位为 Word 排版而非 Word 文字处理，采取 **top-down** 教学方式，一开始让您看到广袤的森林和大穿越走法。此外，我希望本书尽量轻薄短小，主题专注而焦点集中，这意味着阅读本书的您需要若干基础。

读者基础

本书不谈 Word 文字处理，更不谈 Windows 基本操作，谈的是排版与创作平台。作为读者，您必须已经能够在 Word 之中操控鼠标、知道鼠标左右键的用意、知道 click（单击 / 点击 / 按下）和 double click（双击）和 select / highlight（圈选）的作用、知道如何开关文件、保存文件、打印文档，知道如何圈选一段文字、如何搜寻或替换文字、如何让一段文字变粗 or 变斜 or 加下划线 or 换颜色、如何让整段文字左齐 or 右齐 or 置中。



我将在这些文字处理的更高层次上，带您以大局观方式学习大型文档的创作平台和专业排版——这才是本书的目标。

需要循序阅读吗？

最好如此！排版行业有许多术语和概念，视窗操作也有许多术语，您我之间最好先建立共识，因此循序阅读比较理想。本书有流畅的动线，相信我！书很薄，您可以快速扫描一遍。一定要先完整看过第 0 章“导读”、第 1 章“问答 101”和第 2 章“总论”。

为什么选择 Word + Acrobat

众所周知，Microsoft Word 并不是一个最高级的排版软件。在数学界和计算机科学界，最权威的排版软件和文件格式或许是 TeX。在其他领域，PageMaker、FrameMaker、Publisher、QuarkXPress……各领风骚。

我选择 Word 的原因只有一个：它最普及，而且够好。也许有人不把 Word 视为排版软件，但它的确是！所有对 Word 的嗤之以鼻和冷嘲热讽，都源于对 Word 的不够理解。

Word 排版能力对绝大多数需求而言，是充分而足够的。它可以很方便做出大型文档所需的每一项页面元素，它的易用性几乎满足所有人。Word 的 VBA 宏带来的二次开发能力，更拉开与其他同类软件间的巨大差距。Word 文档的后制（网片输出与制版）支持相当丰富，您很难在台湾找到一家无法接受 Word 文档的制版厂——这样的厂商等于对自己的商机设下了极大的限制。

我选择 Adobe Acrobat 作为便携式电子文档（第 12 章）制作工具，是基于相同的原因：普及度高，而且够好。PDF 格式被大量团体和个人采用、制作过程极其简单、具备基本应有的保护功能和压缩功能；阅读器到处可见，而且免费下载！

本书使用的软件版本

本书谈的是 Word 排版和创作平台，所谈的每一项主题在书籍本身皆有实践，这些实践发生在 Microsoft Word 97 和 Acrobat 4.0 身上，并在更新版本中做了确认。Microsoft Word 97 和 Acrobat 4.0 都不是各自的最新版本；执意在旧平台上实践本书，是为了让您知道，**本书不受版本限制**。

这样一来，书中的屏幕截图（硬拷图, hardcopy）就有了点麻烦。在旧版本上作业，理所当然使用旧版本截图，但不少朋友认为可能会对读者带来心理障碍。此话不无道理，我对此挣扎许久。其实各版本的菜单、对话框、工具栏的画面和位置安排都是大同小异。如果我现在执着于屏幕截图的版本问题，我就必须永远执着于它，而我正是非常不喜欢随着版本更动的人。有些道理可用十数年，为什么我却要让自己每年变动，逐波其上呢？

我最后对自己的内心说：顺其自然吧！本书繁体版在 Word 97 完成，那就用 Word 97 画面；本书简体版在 Word 2000 重新抓图（考虑责任编辑的电脑环境），

那就使用 Word 2000 的画面。如果读者非得要找一本与您的电脑画面完全一致的 Word 书籍不可，因而与本书错身，我将轻轻一笑：我们的缘分未到。

总之，本书适用于 **Word 97, Word 2000, Word 2002, Word 2003**，以及将来的更新版本（除非 Word 不再向下兼容）。本书也适用于 **Acrobat 4.0, 5.0, 6.0**，以及将来的更新版本（除非 Acrobat 不再向下兼容）。



我还是留意了若干在不同版本中变异较大的菜单、对话框、工具栏，并在它们出现时特别提醒您。

本书内容与软件版本相依吗？

要看从哪一个角度来回答。

我已经将本书所谈内容在“截稿日期为止，存在于市面上的 Word + Acrobat 最新版本”做了测试。因此就功能性而言，本书内容与软件版本不相依。

但是同一软件的不同版本的菜单（menu）或工具栏（tool bars）的排列方式，甚或菜单项（menu items）名称，都可能有一点点小差异。因此本书的屏幕截图有可能和软件新版的画面略有出入。这种情况不多见，即使有，变异也极微小。只要稍加留意，您便可以在新版软件的屏幕画面上找到书中所说的东西。

就 2.3 节的排版概念而言，本书所谈的内容甚至和 Word 不相依。换言之，当您学会了这些排版概念和作业方式，一旦换用其他排版软件，概念和操作大同小异。

本书未履踏之处

没有一本书能够上天下地涵盖所有范围。“一次买全”的心态肯定会让您失望。

本书谈的是高级的排版与创作平台，不是低级的文字处理。当然，排版作业需要文字处理的基础，否则无法设置样式内的一大堆格式（5.1 节）。但是本书不谈文字处理！市面上已经有很多书籍很详细很丰富地谈了 Word 的文字处理。

本书也不谈论无关乎排版或写作平台的周边议题，诸如打印机、信封打印等等。

举个例子。本书讨论了图片插入、图片剪裁、图片精准定位等操作，但是不讨

论图片的类型、压缩、分辨率、色彩——这些也影响排版效率和效果，但是本书不谈它们（适当时候谈点概念，还是有的）。

又例如，如果您的成品并不打算大量印刷（例如书籍；采用印刷机），而只是个人作坊式的小量输出（例如毕业论文；采用打印机），那么打印机当然会影响您的输出效果。但本书不讨论打印机。



本书名为《Word 排版艺术》。艺术一词可从各种角度解释，我对艺术的解释是：在工程中实现简洁与高效，扫除 dirty works，也是一种艺术！系统化、组织化、甚至相当程度自动化地完成大型文档，便是一种艺术。

本书文字表达习惯

我不打算把话说成下面这样：（以下双横线杠掉的文字在本书中将不会出现）



请点击“格式”菜单下的“项目符号和编号(N)”项目，在其所呈现的“项目符号和编号”对话框中点击“多级符号”索引标签，选择其中某种多级符号后，单击“自定义”按钮，填妥（点击）其中的“级别(V)”清单、“编号位置”下拉框和“将级别链接到样式(K):”下拉框，最后单击“确定”按钮。

文字搞成这个样子，写和读都累。为了清晰明快地描述 Word 操作程序，表 0-3 列出本书的文字表现习惯。以表 0-3 为依据，上述那一段话我会写成这样：

请点击【格式→项目符号和编号(N)】，在其所呈现的【项目符号和编号】中点击【多级符号】，选择其中某种多级符号后单击【自定义】，填妥（点击）其中的【级别(V)】、【编号位置】和【将级别链接到样式(K):】，最后单击【确定】。

表0-3 本书文字表现习惯

种类	实例	特征	说明/助忆
菜单	【格式→边框和底纹】	中括号+箭头	有时会有三层
按钮	确定、取消、自定义	边框+阴影+灰底	助忆：看似可按
对话框	边框和底纹、样式	双边框+灰底	助忆：有点像对话框的标题栏
索引标签/页签	页面边框、边框、底纹、多级符号	虚边框+阴影	助忆：有点像属性页上端小纸签
对话框内的控件或栏目(名称)	应用于、预览、填满	灰底	出现于对话框上的各种东西
Word术语	文档结构图、页眉、页脚、标题、样式、域、目录、索引、脚注、交叉引用、节……	蓝色字 + 黑体	Word提供的功能或概念
样式名称	标题1、目录2、索引3、正文、!大标、!dialog	灰底 + 隶书体(中) Lucida Sans(英)	
醒目效果	视窗控件、页面元素、排版要件、多级符号、字体名称	灰色字 + 黑体	只要有助阅读，我都可能使用这种字体
域名称(英)	Page、Index、TOC、TC、XE、StyleRef……	Lucida Sans字体(英)	
域内容	{ PAGE [* 格式选项] }	小字+灰色边框 Courier New字体(英)	

视窗控件之称谓：视窗、菜单、项目、图标、工具栏……

为求与读者有所共识，以下列出各种视窗控件¹（视窗上各种组成）的称谓。这些名称都以 Microsoft 公布之词汇表为依据。

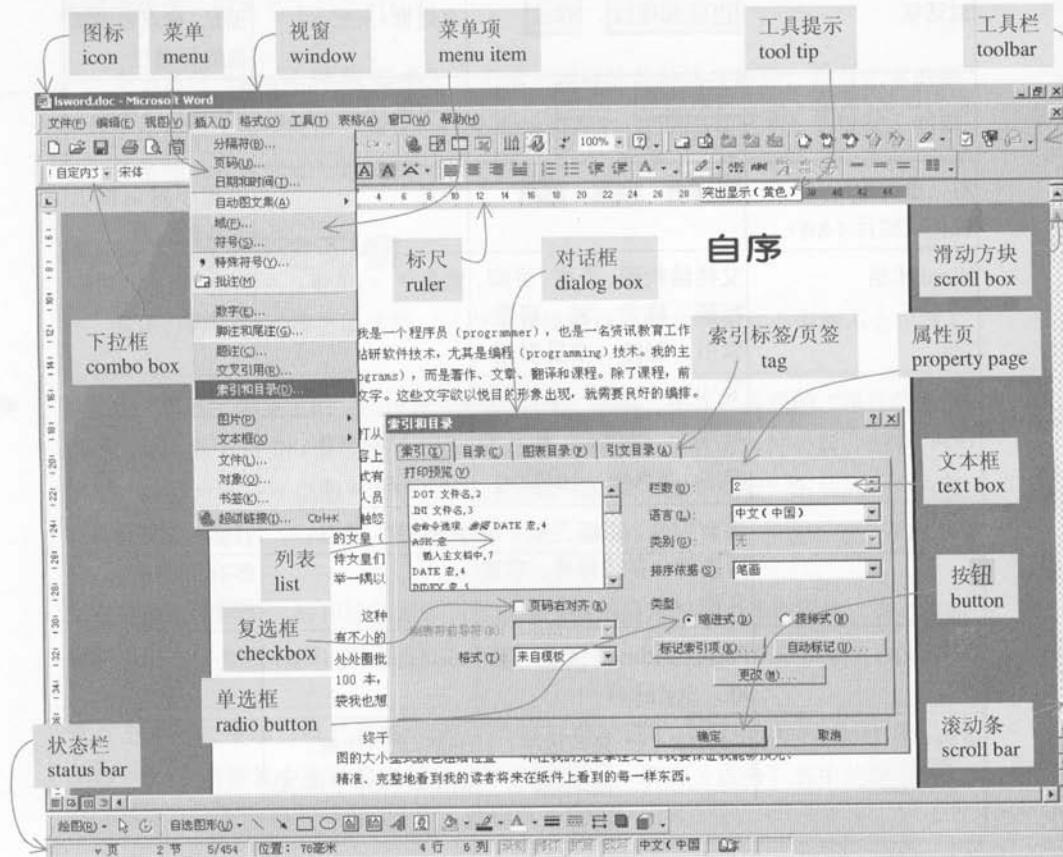


图0-1 Microsoft Windows 视窗元素

¹ 由于这些“东西”英文称之为 controls，所以普遍译为“控件”——控制元件的意思。也有人习惯称为元件。

文档组成之称谓：扉页、序言、目录、正文、附录、索引……

大型文档（书籍或论文等）的组成如下。虚线框起的部分代表页眉与页脚（也有人将它们合称为页眉）的种种变化。为获得页眉上的巨大弹性，排版者必须为每一单元（前言、目录、致谢、各章……）设立一个节²。下图右侧说明 Word 的目录自动化（第 7 章）、索引自动化（第 8 章）和脚注自动化（5.3 节）。

书籍、论文、大型报告

单元	奇数页 起始	奇偶页 页眉内侧	页码 数字	页码 编号	文字 编排
扉页	是	无内容	无	起始	本图未能呈现脚注。随页式的脚注编号和脚注文字可随着“被注文字”的移动而移动。脚注编号可自动调整
题献	是	同	无	续前	
序言	是	同	i,ii,iii...	续前	
目录	是	不一定	i,ii,iii...	续前	所有单元中的特定 样式 可被自动收集成为缩进式目录（最多7层）。默认以“标题1”，“标题2”，“标题3”等 样式 为收集对象，亦可由用户指定。收集得到的目录将以“目录1”，“目录2”，“目录3”等 样式 呈现形貌。
前言	是	异	i,ii,iii...	续前	
致谢	是	同	i,ii,iii...	续前	
第1章	是	异	1,2,3...	起始	所有组织单元都以各种 样式 进行文字编排
第2章	是	异	1,2,3...	续前	
...	是	异	1,2,3...	续前	
附录	是	不一定	1,2,3...	续前	所有单元之中凡是被用户设置为索引项目（index entry, XE，一种域）之文字，皆可被自动收集成为缩进式索引（最多7层）。用户可批量设定 XE ，非常方便。收集得到的索引将以“索引1”，“索引2”，“索引3”等 样式 呈现形貌。
索引	是	同	1,2,3...	续前	

图0-2 大型文档之各种组成单元

² 节这个字眼，在Word英文版中是“Section”。但不论中文或英文，都和我们平常说的“第1章第2节”的“节”互相混淆。我个人认为如果使用segment或session比较好些，中文译为“区”也比较好些。本书将遵循官方称谓：节。

页面元素之称谓：页眉、页脚、页码、正文、标题、版心……

下图最外缘之粗框，请视为一本书或文档的某一页边界。

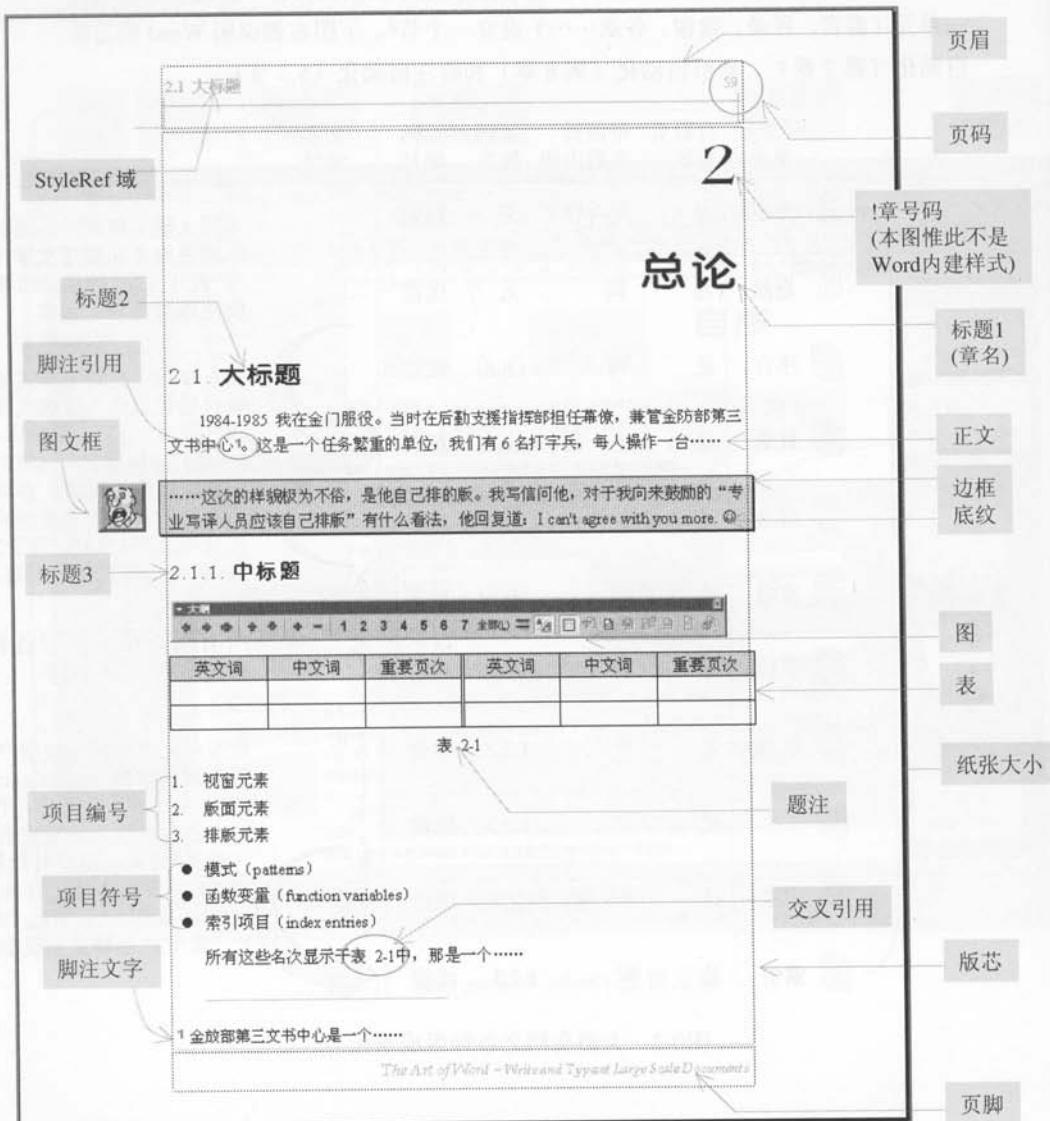


图0-3 Word 所提供的页面元素；每一个都是 Word 的内建样式

构成页面的所有元素，本书称之为页面元素。这些元素也就是排版者必须制造出来的东西，例如页眉、页脚、页码、标题、正文、脚注、图、图编号、表、表编号、网底（底纹）、目录、索引……上页所列的是各种页面元素在 Word 中的称谓，本书所列都以 *Word On-line Help* 为依据。各种页面元素的说明详见 p.83，表 2-1。

并非所有排版效果都可以从图 0-3 显示出来。例如：奇偶页页眉内容不同、文前和正文的页码采用不同编页方式、每章从奇数页开始……就没有显示。此外 Word 还具有自动化抽取标题成为目录、自动化抽取索引项成为索引，自动化变更图题注和表题注，并自动化抽取图题注成为图目录、自动化抽取表题注成为表目录的能力。当然啦，所谓自动化，不是自动/主动，您必须下（很简单的）命令。



图0-4 Word 抽取标题成为目录，并以**目录 1~9**……样式呈现之

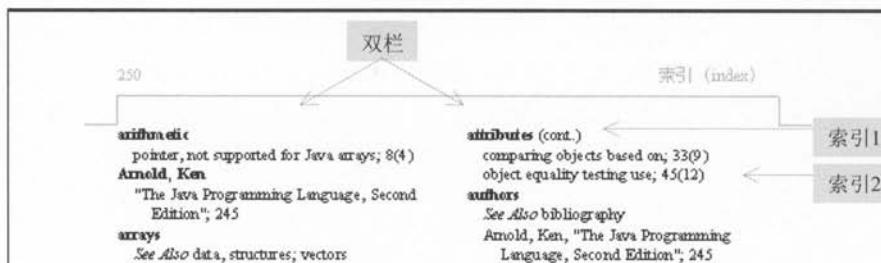


图0-5 Word 抽取索引项成为索引，并以**索引 1~9**……样式呈现之

排版要件之称谓

构筑各种页面元素时，排版者所需工具我称之为排版要件。这些要件也就是排版者操控的对象。

图 0-3 所列的页面元素，几乎都有同名的排版要件。此外另有一些不可视的排版要件，如样式、模板（样式库）、域、分隔符号……详见 p.83，表 2-2。



附录 A 内容索引

1. 问答 101	23
关于出版 and 书籍制作流程	23
关于 Word 写作平台	29
关于 Word 排版概念与环境设置	31
关于 Word 排版实务	38
页面设置 (Page Setup) 与系统设置	38
文字与段落 (Texts & Paragraphs)	42
图片 (Figures)	46
表格 (Tables)	47
关于 Word 程序可控性	47
关于 Word 电子文档	49
关于 PDF 便携式文档	50
本书所有版式与样式细节	51

1

问答 101

这里收录的问答，涵盖写作平台、排版、制版、印刷、出版流程中的常见疑惑。有些问题并非本书焦点，但作为上下游，或许也是本书读者关心或感兴趣的话题，所以一并整理进来。提问以 ○ 为记，回答以 ➤ 为记。

关于出版 and 书籍制作流程

- 出版流程是什么？
 - 首先是内容确定（作者及编辑定稿），然后是排版（排版人员或作者完成），然后是制版（由制版厂完成），之后是印刷（由印刷厂完成），然后装订（由装订厂完成），然后是入库，最后是发行（由渠道商完成）。书籍封面也需经过设计、定稿、印刷的过程。完整流程如图 1-1。
- 制版有哪几种做法？对品质有何影响？
 - 早期使用照相制版：完稿（排版完成后之稿件）后先打印到优良的纸品上（通常是以激光打印机输出至雪铜纸），再经照相，晒制成所谓的蛋白版。这个过程受到照相因素的影响，品质不易有绝对掌控。近来使用软片输出：将电子文档 (.doc 或 .pdf 或其他) 送入照排机，输出软片（一种胶片），再晒制成 PS 版。或甚至跳过软片直接输出 PS 版。完整流程如图 1-2。使用照排机，过程中没有任何外在因素参与，因此品质稳定良好，对于底纹效果（← 您现在所看到的灰色底纹）亦有很好的表现。

○ 单色和多色印刷，制版上有什么差别？

- 单色（黑白灰）印刷只需一块印版。多色印刷需先分色处理：依完稿文件的设置，进行文件转换，分成CMYK四个印刷基色，并打印“数字化打样”进行校对，校对完毕后输出软片并晒版。多色印刷则需四块印版（或直接输出PS版）。



图1-1 出版流程

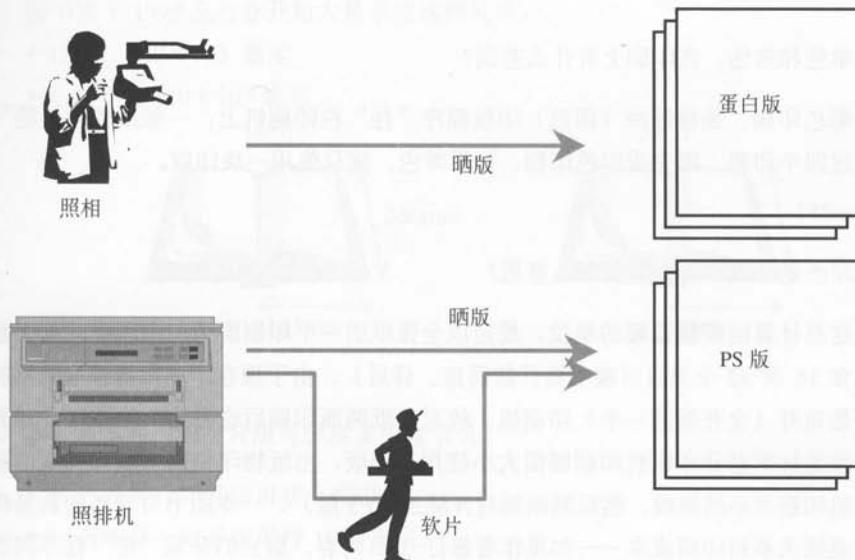


图1-2 制版流程

- 印刷版材有哪些种类？如何印刷？
 - 印刷版材有早期惯用的蛋白版和近期普遍使用的 PS 版（Positive Plate）。印版上了印刷机后，即可高速而大量地印刷。目前高级印刷机都已由电脑控制，甚至由印前系统直接连接印刷机，自动完成制版、印刷。两个名词值得认识：**CTP**（Computer to Plate）和 **CIP**（Cooperation for Integration of Prepress, Presss, and Postpress）。

- 印刷版材的材质、保存期限和耗损因素为何？
 - 蛋白版是早期胶印印版的一种。以金属版为版基，版面涂布蛋白质与重铬酸盐的混合液，经紫外线曝光而生成不溶于水的亲油性硬膜。图文的基础就是建立在感光硬化的膜上。因蛋白版耐酸耐磨性能较差，耐印率较低，且网点容易变形，故目前已很少使用。PS 版是目前使用较广泛的一种优良版材。它是一种涂布树脂与重氮化合物调配成感光液的铝制备用版材，一般每版可印 10 万印左右，印刷效果较精美，保存期限约（0.5~1）年，取决于气候、温湿度造成的氧化程度以及印刷时的磨损程度。一旦过了保存期限，若要再印刷必须重新制版。另有中间产物软片，是一种类似于大型的胶片底片，保存期限约 3~4 年，视气候及温湿度造成的氧化程度而定。

- 单色和多色，在印刷上有什么差别?
 - 彩色印刷，是将四块（四色）印版顺序“挂”在印刷机上，一张纸依序“跑”过四个印版，即完成四色印刷。如果单色，就只使用一块印版。

- 印张是什么？对开版是什么意思?
 - 这是计算出版物篇幅的单位。规定以全张纸的一个印刷面为一个印张。其上包含 16 或 32 个书页（视书籍开数而定，详后）。由于现在印刷市场普遍采用的是对开（全开纸的一半）印刷机，故对开纸两面印刷后也称为一个印张。对开版是针对对开印刷机印刷幅面大小使用的印版。出版物印刷前将软片拼成印刷机印版大小的原版，然后晒制到对开版上（PS 版）。一本图书对开版的数量将直接关系到印刷成本——如果作者修订 3 页内容，而它们分别“落”在不同的印版上，重新制版的成本就是 3 块对开版；如果运气好落在 1 张印版上，重新制版的成本就是 1 块对开版。

- 书帖是什么?
 - 按页码顺序折叠成帖的书籍印张。是组成书籍的基本单位。

- 纸厂供应的纸有多大?
 - 目前纸厂供应的常用全开纸张幅面规格：787*1092 毫米、787*980 毫米、850*1168 毫米等，而 890*1240 毫米、900*1280 毫米、1000*1400 毫米的纸张则必须请纸厂**特殊生产**。

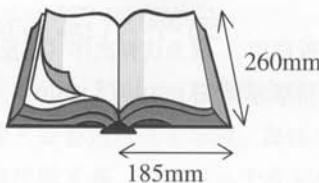
- 开本（16 开、18 开、32 开……）指的是什么?
 - 开本是图书幅面的大小。把全张纸平均切成多少同等尺寸的小张，就称多少开。如全张纸经 4 次对折后，其幅面为全张纸的 1/16，这样幅面大小的开本就称 16 开本。以下是各开本的实际尺寸：
 - ◆ 16 开：185×260 毫米
 - ◆ 大 16 开：210×297 毫米
 - ◆ 大 18 开（国际 18 开）：185×230 毫米。目前英文电脑书惯用尺寸。台湾电

脑书籍于 1995 左右亦开始大量采用这种尺寸。

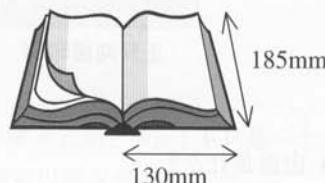
◆ 32 开：130×185 毫米

◆ 大 32 开：140×203 毫米

16 开



32 开



○ 各种开本的一块对开版可拼凑多少张书页？

➤ ◆ 16 开本是一对开版可拼 8 面书页。

◆ 32 开本是一对开版可拼 16 面书页。

○ 书籍尺寸与原始（排版系统中的）设置是否绝对准确相等？

➤ 不，都有误差存在。书籍实物是裁切而来，裁切机没有那么高的精准度，但误差也不至于太多。

○ 装订时裁剩的纸干什么用了？

➤ 由于印版上有图文的部分要小于纸张的幅面，书刊装订时要将多余的部分裁去，那么裁剩的纸对于出版商来说就是废料了。因为大多数纸厂并没有提供太多种纸张尺寸供选择（刚才曾举出六种：三种普遍规格，三种特别规格），所以如果书籍的开数选择不当，会影响成本。通常开数由出版社决定（作者也可能参与意见），涉及环节很多，包括装箱时是否符合既有箱子的尺寸、上架时是否符合既有书柜的尺寸、上架效果如何等等。

○ 拼版是什么？

➤ 把书刊的多页胶片根据印刷条件、版式规格的不同（如，印刷机印刷幅面的大小：全开、对开；装订方式：开本、横开、竖开的不同等），在片基上拼制成立版用的原版。因此原版上的书页排列方式就不该是 1,2,3,4… 流水排列，必须顺应将来书帖的对折程序，这个动作就称为拼版。请参考图 1-3。

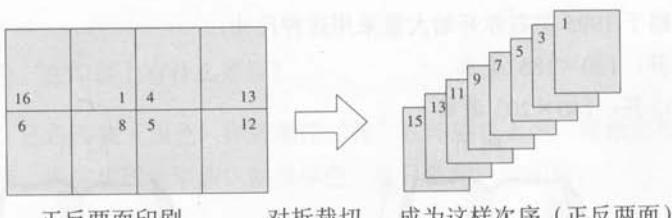


图1-3 拼版示意图

○ 出血是什么？

➤ 怪可怕的名词。其实没有血，亦不见红。当底纹出现于页边，不能只是刚好涂到页边为止，应该更多一些（至少3毫米），才不会因为折纸裁切而使页边“露馅儿”。此谓出血。请参考图1-4。如果图片（底纹）尺寸做得刚刚好，没有出血，成书时裁切边上会出现白色边（纸的颜色）。

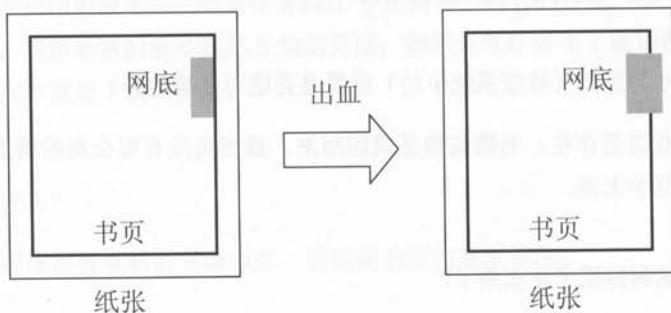


图1-4 出血示意图

○ 重印和再版指的是什么？绝版是指什么？

➤ 重印是图书第一次印刷后，没有改动或改动甚少的印制。由于对排版的软片不作改动或仅作少量改动，重印可以加快出书速度并使生产成本大大降低。图书重印时，开本、书号、版式均不变，但封面、扉页可以重新设计。如《深入浅出 MFC》（第二版）第一次重印，标“第二版第二次印刷”。再版是对原书作较大修改后重新排版印刷。再版虽以原书为基础但要进行较大的修改，有时甚至是脱胎换骨的改造，所以操作时完全按照生产新品种的程序进行，制作成本并不低于或稍稍低于初版书。再版的书，有的在书名后加括号标明“修订本”，有的不标明。版本记录中对第一次再版标“第二版”，再版的图书，不可再用原书号，必须用新的书号；除书名外，其他如封面装祯、开本、版式等都可以

重新设计。一旦出版社不再印制某本书，我们便说它绝版。如果重新制版，或软片并未被销毁，绝版书当然还可以重印——书籍毕竟不是邮票或钱币或版画，是完全由市场供需而决定是否加印的一种物品。

关于 Word 写作平台

○ 写作平台（Writing Platform）是什么意义？

➤ 不知道其他人是否用过这个字眼。我使用这个字眼来说明：Word 不只是一个文档定稿后的排版工具，而且是一个在写作期间就可以带来帮助的平台。

○ 何谓“在写作期间就带来帮助”？

➤ 大型文档的创作过程十分繁复，又有许多琐屑细节要注意。由于写作是一种不断修改、增删、搬移文字的过程，而科学工程类书又有许多需要交叉引用（cross reference）的地方，例如“请参考图 3-2”或“请参考 2.4 节”或“请参考第 125 页”，但那些被参考点在写作过程中一直是变动的。这就会带来很大的麻烦，不但影响写作思路，也大量增加后期整理时间。Word 可以解决这个问题，自动化地为它们编号，并于它们有所变动时同步（但非即时）修正所有引用点。

○ 页次自动编号，Word 做得到吗？

➤ 毫无疑问。详见 4.3 节。

○ 两个编页系统（i,ii,iii…和 1,2,3…）共存于一份文档，可以吗？

➤ 毫无疑问。利用 Word 提供的节（sections），n 个编页系统也做得到。详见 4.3.1 节。

○ 图表自动编号，Word 做得到吗？

➤ 毫无疑问。“图 2-1”、“表 3-4”这种东西称为题注（captions），其中的“图”或“表”称为标签（labels）。您可以自定义标签，例如“小秘诀”、“方程式”、“程序列表”等等。详见 5.10 节。

- 章节标题（heading）自动编号，Word 做得到吗？
 - 毫无问题。但 Word 有些无伤大雅（可以回避）的 bugs 存在。详见 5.11 节。
- 脚注（footnotes）自动编号，Word 做得到吗？
 - 毫无问题。而且您可以决定脚注的显现风貌。详见 5.3 节。
- 还有什么自动编号是 Word 可以提供的？
 - 书签（bookmarks）。详见 11.3 节。书签并不是一种页面可视元素，而是排版的一个幕后英雄。
- 自动编号的东西一旦在文档中被移动位置，会自动重新编号吗？
 - 会自动化，但不会自动——需要您做一道非常简单的手续。详见 9.2.2 节。
- 交叉引用（cross reference）怎么做？
 - 别在文档中直接写“请参考图 x-x”，应该写“请参考 |”（| 表示输入点），然后点击【插入 ➔ 交叉引用(R)】，再选择您想引用的东西。所有被 Word 自动编号的东西，例如图表题注（包括标签、编号、文字、页码）、章节编号（包括编号、文字、页码）、书签（包括文字、页码）、脚注（包括编号、页码），都可以被做成交叉引用。被引用物一旦有任何变化，引用点也会随之改变。详见 5.12 节。
- 目录可以在写作过程中随时被看到并更新吗？
 - 可以，前提是文档的编排必须使用约定样式（styles），不能任性而为。目录可由 Word 自动化制作，但不会自动更新——您必须办一道非常简单的手续。详见第 7 章。
- 索引可以在写作过程中随时被看到并更新吗？
 - 可以，但是不建议。索引应该在文档全部底定之后再来考虑，详见第 8 章。
- 索引很难搞吗？
 - 索引的制作需要作者协助，不可能由排版者单方完成。一般认为索引项必须逐一插入文档内，但 Word 也提供批量作业方式，不但系统化，并有助于留下索引制作轨迹，利于日后的追踪修改。详见第 8 章。

- 文档各单元（例如各章）从奇数页开始，容易办到吗？
 - 简单极了。只要为每个单元设置自己的节，即可办到。详见 3.4.1 节。
- 还有哪些 Word 功能可以辅助写作？
 - Word 文档结构视图会在视窗左侧显示一个文档结构图，即时呈现最新目录（还可以折合开展），有助于创作者掌控大局，详见 2.3.1 节。文档结构图内的章节条目，连同先前所说由 Word 自动编号的各种题注（captions）、书签（bookmarks），都是超链接（11.2 节），有助于创作者在创作过程中的择点跳跃。Word 宏允许创作者发挥各种创意，写出各种功能来辅助 Word，例如写出“文字转换”批量作业功能，进行两岸用语自动转换等，详见 10.4.3 节。

关于 Word 排版概念与环境设置

- 文字处理（word processing）和排版（typesetting）如何区分？
 - 这两个字眼在某些范围可能略有重叠。通常文字处理是指对文字本身的处理（字体、大小、颜色、字距、行距、特殊效果……），扩大一些还可包括文字段落的设置（段落前后间距、首行缩进、居中或靠左或靠右排列……）或图表的制作。排版则涵盖更大范围的版式设置，例如页边距、天头地脚（详后）、页眉、页码、尾注、目录、索引……请参考图 1-5。
- Word 是专业排版软件吗？可独力实现一整本书的排版吗？
 - 早期的 Word 排版功能或许不强（唔，我不知道），再加上其命名，往往被认为只是一种文字处理软件。但以我个人经验，至少从 Word97 开始它就是一个功能相当齐全的专业排版软件，绝对可以胜任书籍或论文的排版需求。本书从头到脚完全以 Word 完成。Word 的弱点在于其分色能力，以及页面尺寸（A3 以上恐怕就没戏了）。

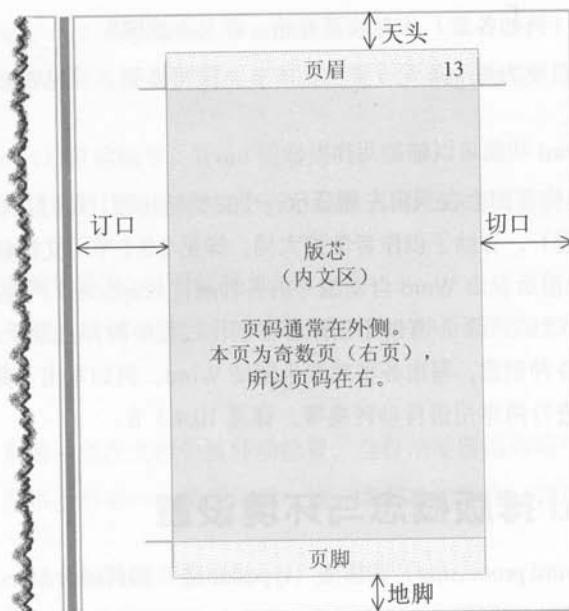


图1-5 页面布局：天头地脚、页眉、页脚、页码、订口、切口、版心

- 彩色文档适合以 Word 完成吗？
 - Word 的色彩模式为 RGB，适合屏幕显示，因此制作彩色电子文档是很适合的。但印刷用的色彩模式是 CMYK。虽然目前的输出机可将 RGB 三色转换为 CMYK 四色进行印刷，但颜色较不饱和，容易出现偏差。所以 Word 不适合作为印刷用途的彩色排版。
- 试比较各排版软件如 Word, PageMaker, Publisher, Quark XPress
 - Word 以外的上述其他三个排版软件都是组页式排版，文字处理能力较弱，但具备极强的分色能力，并可应付变化多端的版面，页面大小也比较有弹性。Word 恰好相反，文字处理能力极强（这是很大的优点，也是易用性的强大保证），但分色能力薄弱，而且页面尺寸稍大一些就挂了。但其宏提供了二次开发能力。
- 排版是某种 dirty work 吗？是某种繁琐而重复劳动的工作吗？
 - 如果猛冲猛撞，暴虎冯河，率性而为，排版绝对是件 dirty work。但如果设好各种样式，认清根本的作业方式，知道什么该做，什么先做，什么后做，什么不做，那么排版是件轻松愉快的活儿。正规排版作业的系统性/工程性甚强，并不需要繁琐而重复的劳动——这话只对具备良好素养的排版人员才成立。

○ Word 的【视图(V)】菜单下有许多视图，做什么用？

➤ 除了主控文档视图（2.3.6 节），其余的普通、Web 版式、页面、大纲、文档结构图等五种视图（统称视图模式，详见 2.3.1~2.3.5 节）均是以某种角度（方式）观察同一份文档，可说是文档的一体多面（如图 1-6）。此相当于《*Design Patterns*》一书所言 MVC（Model-View -Controller）模式（范式）的实现，也相当于 MFC 程序设计的 Document-View 概念：Document 就是文档本身，View 就是视图。

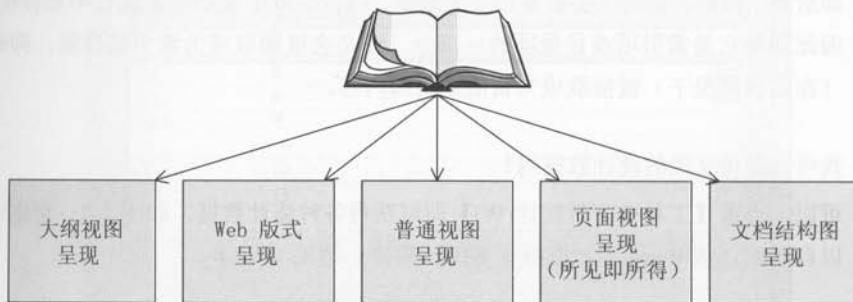


图1-6 文档与视图的关系示意图

○ 该把一份文档置于同一文件，或是另有其他选择？

➤ 如果文档体积很大，整份文档置于单一文件会对 Word 的反应速度有影响。但目前电脑速度很快、内存基本配备很高，相当程度地允许大量内容集中于同一个文件（p.72 的脚注 11 有我的一些个人经验）。我个人觉得置于单一文件比较直观而且方便。但您亦可选用主控文档视图将整份文档分散至数个文件（本书仅在 2.3.6 节大略讨论主控文档视图）。

○ 是否该为了加快 Word 反应速度而使用普通视图或大纲视图？

➤ 普通视图、Web 视图或大纲视图的确是最阳春的文档查看方式，屏幕（页面）展示效率也因此最高。但最满足排版人或创作人需求的是 "What you see is what you get"（所见即所得），也就是页面视图（2.3.3 节）。就绝大多数的文档大小和电脑配备而言，使用页面视图应该不会令作业迟缓（p.72 的脚注 11 有我的一些个人经验）。

○ PDF 文档可透过导览窗格内的书签快速进入对应页面。Word 也可以吗？

➤ 是的，Word 有相同设计，称为文档结构图（2.3.2 节）。这不仅有益阅读，也有益写作——作者再不至于迷失于庞大的书稿结构中。对于辅助作者深植架构轮廓有很大的帮助。

- 长时间排版或阅读，白底文档容易使眼睛疲劳。是否有其他选择？
 - 深底色可以让眼睛舒服一些。请点击【工具→选项(O)】，再在对话框中点击 **常规+蓝底白字**。

- 文字设为白色有什么后果？可藉此设置某些文字不被打印出来吗？
 - 白色白底文字不会显示于屏幕（除非您改用蓝底白字）。根据页面视图“所见即所得”原则，它也不会被输出（于纸张、网片、PDF 文档）。但它毕竟存在，因此如果它是索引项或目录项的一部分，还是会被抽取成为索引或目录，抑或（在适当情况下）被抽取成为页眉或页脚内容。

- 我可以获得文档的统计数据吗？
 - 可以。点击【工具→字数统计(W)】即可获得各种统计数据，如图 1-7。您也可以自制宏，从 Word 手上取得更多统计数据，例见 10.6 节。

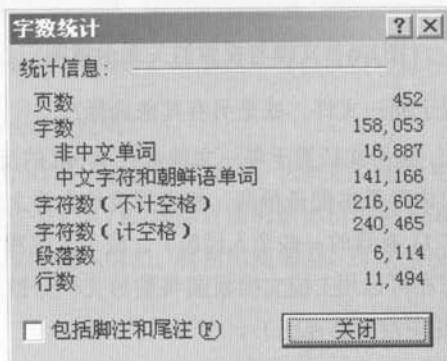


图1-7 Word 的**字数统计**画面

- Word 有不少自动进行的字符串替换功能，可否取消？
 - 可以。Word 会自动将 :) 变成②，将 :(变成③，将 ==> 变成 ➤，将 <== 变成 ←，将 <=> 变成 ⇄，将 (tm) 变成™，将 (c) 变成©。如果嫌这些举动过于“鸡婆”，可点击【工具→自动更正(U)】如图 1-8，取消勾选**键入时自动替换**。如果不打算全部取消，那么就从图中清单一一挑出欲取消的“替换字符串”。如果没有取消它们，输入文字时便会出现上述的自动替换字符串，此时亦可在它们一出现时按一次 **BackSpace** 键令现出原形。

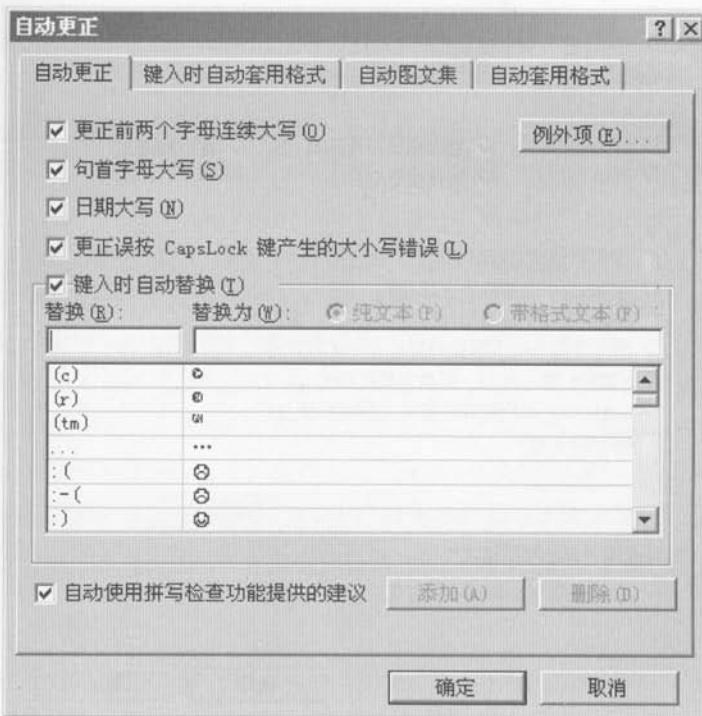


图1-8 Word 文字自动更正与替换（例如自动替换笑脸）

- Word 有不少自动套用格式功能，可否取消？
- 可以。Word 会自动将直引号 “xxx” 变成弯引号 “xxx”、将括号自动正确匹配：“[” 配对 “]”，“(” 配对 “)”，“{” 配对 “}”。又自动将 “1st” 变成 “1st”。如果嫌多此一举，可点击【工具→自动更正(A)】，再点击【自动套用格式】或【键入时自动套用格式】（如图 1-9），勾选其中想要的，取消不想要的。这些功能的原始构想虽然很体贴人意，但有时候确实多此一举。例如我撰写 STL（一种 C++ 程序库）相关书籍和文章时，常为了 Word 自动为我配对“正确的”括号而头痛不已。STL 的数据区间是前闭后开区间，表现应如 [a,b,c)，如果 Word 老强迫转为 [1,2,3] 就头大了——我将因此总是必须回头把 ‘[’ 改为 ‘)’。更不幸的是如果将这样的文档交给他人，而对方的 Word 并没有取消“自动匹配正确括号”，将是灾难一场。我曾经遭受过这种困扰。

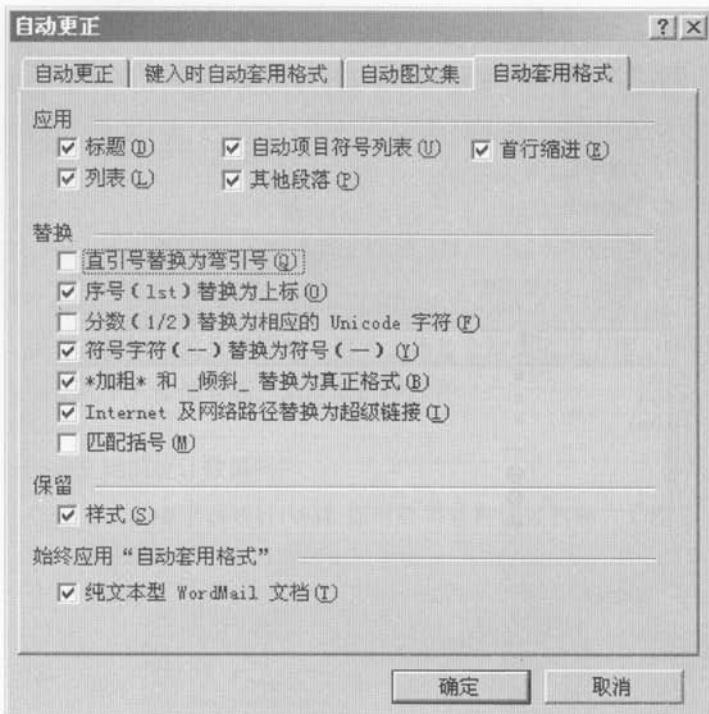


图1-9 Word 文字自动套用格式（括号正确匹配、引号替换……）

- Word 的版本兼容性如何？
 - 一般公认 Word 向下兼容的能力很不错。本书所谈都是 Word97 以来就有的功能，适用于其后所有更新版本（已经过测试）。

- Word 文档结构图有时候表现失常！
 - 就我的经验，您所遇的问题大概有两种。一是它没有正确抓出样式为 **标题 1**、**标题 2**、**标题 3**……的文字加以显现，另一是它没有抓出“标题自动化编号”所得的标题数字（例如 “3.2.3 节”的那个 “3.2.3”）。前一个问题令人心脏麻痹，幸好救心药很便宜：多展开折叠几次就会正常，后一问题是个小 bug，在某种情况下才会出现，可以回避。详见 5.11 节。

- Word 自动化章节标题编号有时候表现失常，乱编！
 - 大概是因为您给它太重的负担了。更新所有域（详见 9.2.2 节），就可以让 Word 恢复神志。

○ 我的文件体积莫名其妙地暴涨！

➤ 的确会如此。有时我确定已把 24-bit 全彩点阵图换为 16 色，文件大小却没有减少。有时只是做了一些简单动作，体积却暴涨两倍。这是因为 Word 支持所谓的快速保存能力：许多时候当您保存文档，Word 并不把整份文档按照内容前后次序储存一遍，而是以（如果可以的话）添加方式将某些东西附加于原文件尾端（其中可能有对原文档内容的修改指示）。因此虽然删除（或替换）某张图片，说不定 Word 只是把新图片加进来，并对原图片做一个“逻辑删除”动作（做一个记号而已）。这便可以解释诡异的“不减反增”现象。如果您确定您的作为不会造成体积暴涨，别慌，Word 迟早要还一个公道——也许是在多次 **保存(S)** 之后。您也可以要求 Word 立刻把垃圾清干净：点击 **另存为(A)**，并选择同名文件。这个动作在刚保存完毕的文档上是无效的，因为 **保存(S)** 后如果未对文档做任何修改而又点击 **另存为(A)**，Word 97 将不会对您有任何理睬。您也可以点击 **【工具→选项】+保存**，取消 **快速保存**

○ 谈谈 T_EX 排版软件

最近闲时研究 T_EX 排版系统。老实说这套由 Knuth 发明又有众多高手不断发展和使用的软件，实非我等凡夫俗子能轻易上手。如果从头 DIY，单是成功令其正确处理中文就要大费周章；编写一份稍复杂的专业文档都得记忆一大堆命令和符号；如果文档带有格式错误，系统给的提示信息也可能令初学者莫名其妙。但 T_EX 有个重要的理念极具借鉴意义，那就是 WYTIWYG：What You Think Is What You Get。T_EX 并不鼓励（但也不限制）用户直接指定字体、颜色、缩进、间距、段落格式等属性，而是希望用户专注于自己想表达的内容，通过相应的命令说明作品的逻辑关系。用户需要告诉 T_EX 的是“这是一章开始”、“这个单词应该强调”、“这里是一段诗”……而不是告诉它“这是第 3 章，应该用黑体三号字，开头有一个双 S”、“这个单词用斜体楷书 9 磅”、“左右缩进各 1 英寸，右边不对齐，换用小一磅字体”……这样，即使丝毫不懂排版艺术，也可以用 T_EX 输出专业文档，甚至直接付印。使用 Word 这一类“文字处理”软件，如果没有专门指导，普通用户排版出来的文档可能就中看不中用，或者根本就不堪入目。当然 Word 也提供很多机制如域（fields）、模板（templates）、样式（styles）、题注（captions）等等，让用户能够更专注于文档的结构和内容，但这部分功能似乎没有得到应有的强调。所以从一开始听到《Word 排版艺术》出版的消息，我就非常期待。换个角度看，T_EX 这样的系统可以看作是将专家经验数字化，使普通用户得以共享的平台。如果能够把这种共享以更人性化，用户体验更舒适的方式提供出来，应该可以作为 Office 软

件的一个发展方向。

- 莫把人性化、亲和性看成是一件轻松可达的事，那其实是一个成功软件（在市场上、在普及度上、在贡献度上）之所以成功的极关键因素。Microsoft Windows 和 Microsoft Office 开发团队投注在人因工程上面的时间和人力非常惊人，这些软件的成功自有其功能性及市场竞争手段以外的因素。我认为您所描述的 T_{EX} "What You Think Is What You Get"，在 Word（或其他排版软件）正是体现在样式（styles）身上。用户想指明“这是某一章开始”，那就选择**章名**（或**标题 1**）样式；想指明“这个单词应该强调”，那就指定**强调粗体** 样式；想指明“这里是一段诗”，那就指定**诗体** 样式……一般文档所需样式 Word 都已备妥，特殊样式（例如上述**诗体**）则当然必须由用户自己定义出来。不论一般（内建）样式或特殊样式，都可能需要调整或新设，这时候就是软件方便性和亲和性显优势的地方。当一个软件“实非我等凡夫俗子能轻易上手”时，它的命运将落在孤单的社群内，毕竟人类社会 99.99% 由凡夫俗子构成。Word 的众多功能“没有得到应有的强调”（尤其是写作平台相关功能诸如域 fields、模板 templates、宏 macros、题注 captions、多级大纲编号 multi-level outline numbering、交叉引用 cross reference、追踪修订 track changes、批注 comments、书签 bookmarks 等等），正是我写作《Word 排版艺术》的动机。任何软件用户在晋升 power user 层级时几乎都需要专门指导（从书籍、从文献），而 T_{EX} 用户所需的专门指导，我相信远远高于 Word 用户所需要的专门指导（后者也许只需要这本书就够了，I hope so ☺）。任何一种排版软件都有其诉求，Word 之于一般以文字为主、非美工专业、非彩色印刷的需求绝对是够好而易用的，这是本书乐意传达并证实的一点。

关于 Word 排版实务

页面设置（Page Setup）与系统设置

- 排版的第一件事是什么？

- 言人人殊。我认为是【文件→页面设置(U)】，先确立纸张大小和边界（详见 3.1 节），这就把尺寸、版心位置、天头地脚等等都先定了下来，然后是节的设置（详见 3.4 节）以及各种样式设置（详见 5.1 节）。

- 节（Sections）是什么？

- 这是很被 Word 用户忽略的一个排版概念。节（详见 3.4 节）用来设置一个个排版单元，使不同的单元得以拥有不同的大局设置。您可以在独立的节中设置专

属的编页方式、起始页次（奇页或偶页）、页眉/页脚内容……如图 1-12。如果忽略节的设置，整份文档就是单一的节。通常我会为每一文档单元（序言、目录、致谢、第 1 章、第 2 章……）各设独立的节，然后便可令它们都由奇数页开始，并令文前（正文前的序言、目录、致谢等等）采用罗马数字编页，正文由数字 1 开始采用阿拉伯数字编页。

○ 模板（Templates）是什么？.dot 是什么

- 模板（templates）又称样式库，是一群样式的集合，并包含版面设置（纸张、边宽、页眉和页脚位置、天头地脚等等）。一旦开始新文档，您可以载入某个模板（详见 5.2 节），这便获得其中所有的样式设置，便可将这些样式套用于文档身上。模板的后缀名是.dot。

“模板”和“样式库”两个字眼，前者出现在【格式→样式库(G)】中，后者出现在打开对话框下方的文件类型清单内。当然，还有其他出现场所。

○ normal.dot 是什么？

- 后缀名为.dot 者便是模板（样式库）。Word 用户的所有作为，其影响范围可分三个层次：(1) 文档自身，(2) 相同模板所衍生的所有文档，(3) 所有 Word 文档。如果您的样式（或宏）储存于 normal.dot 中，便属于第三类。Word 一被执行便会载入 normal.dot，将其中的设置雨露均沾于所有后继被开启的任何文档（详见 5.2.3 节）。

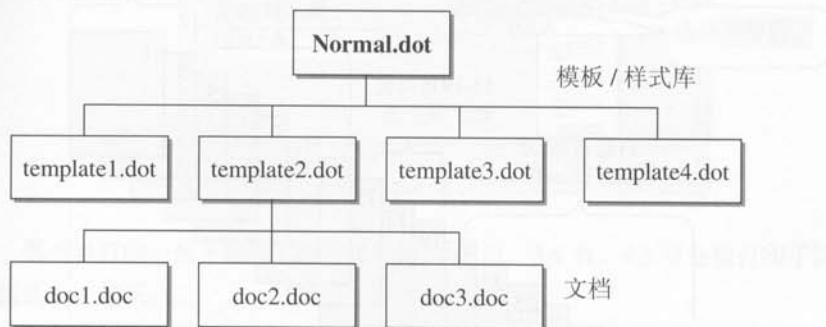


图 1-10 宏和样式可储存于 (1) 文档 (2) 模板 (3) normal.dot

○ 页眉、页脚是什么？天头地脚是什么？

- 页眉是指版心以上部分，通常用来放置章名或节名或页码。页脚是指版心以下部分；有人喜欢把页码放在页脚外侧。天头地脚是排版术语，天头意指页眉以

上的留白，地脚意指页脚以下的留白。通常天头大于地脚视觉效果比较好。如果天头地脚（留白）太小，会予人不舒服的拥挤感。请参考图 1-5。在台湾，页眉是页首和页尾的合称，繁体版 Word 并无页眉这一称呼，简体版 Word 所说的页眉则是繁体版所说的页首。

- 可否让每一章（或说每一单元）一定从奇数页开始？
 - 可以。只需为每一单元设置独立的节，并令它从奇数页开始即可（详见 3.4.1 节）。如此一来，假设前一单元结束于 p.31，下一单元将从 p.33 开始，虽然您不会在 Word 文档中看到空白页 p.32，但打印机的确会输出一张空白页，Word 打印预览也会出现一张空白页，PDF Distiller 输出的 PDF 文档也会有一张空白页夹在 p.31 和 p.33 之间。
- 页码会自动流水排列吗，即使任意安插或删除页面？
 - 可以。Word 随时为每一页自动重排页码，完全无需用户操心。
- 文前是指什么？正文（内文）是指什么？文后是指什么？
 - 文前、正文和文后都是排版术语。通常第一章以前的篇幅（可能包括扉页、序言、目录、前言、致谢）称为文前，各章称为正文，附录、索引等称为文后。

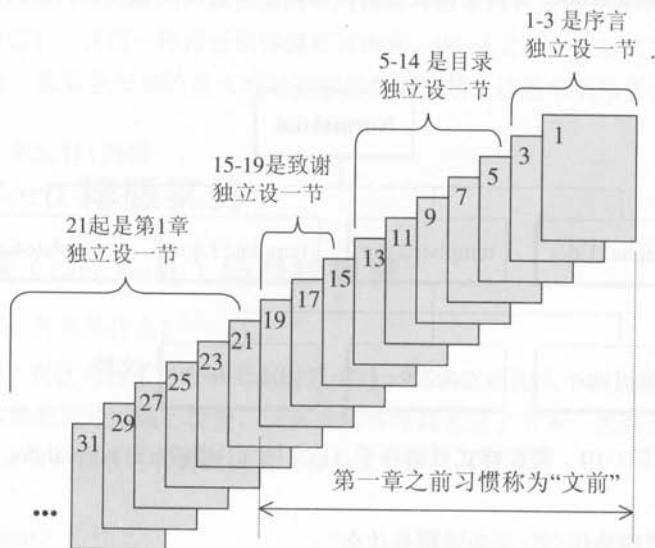
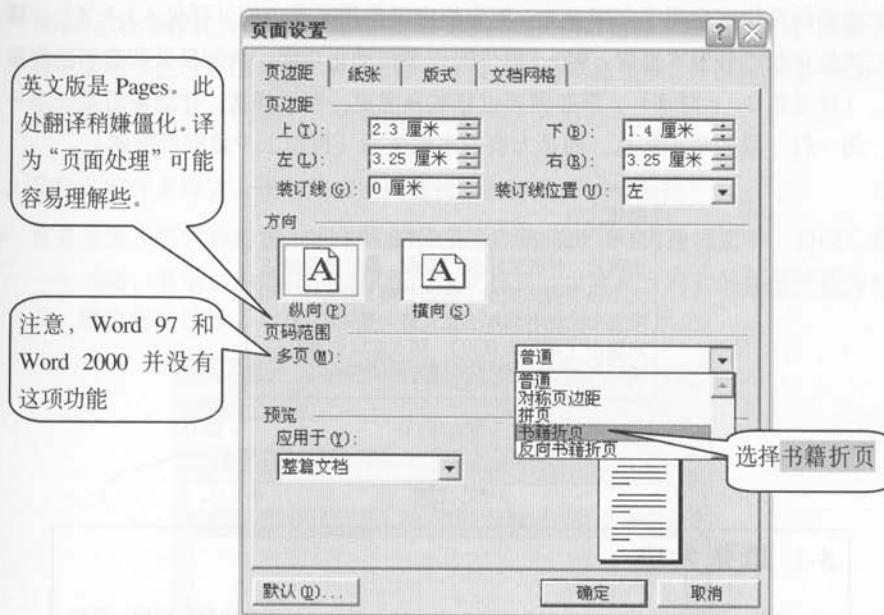


图 1-11 节 (sections) 示意图

- 对文前采 i,ii,iii…编页，对正文和文后采 1,2,3…编页，可以吗？
- 是的，可以。几乎所有书籍都如此做。Word 可对文前和正文采取不同的编页方式——只要您为各单元设置独立的节（详见 3.4 节）。

- 如何以单张 A4 纸打印两页，对折后打钉成为一本小书？
- 您的意思是希望在一张纸上单面打印两页（或双面打印四页），对折后于对折处打钉成为一本小书，那么在 Word 之中该如何排列文档页次，是吗？这个问题有点类似先前说过的“落版”（但比印刷上的落版单纯多了）。可在 Word 2002+ 中点击【文件→页面设置】+【页边距】，在【页码范围】/【多页(M)】栏目中选择【书籍折页】：



那么当打印一份 8 页文档时，1,8 页、2,7 页、3,6 页、4,5 页会被打印于同一张纸上，如下：

8	1	2	7	6	3	4	5

如果再安排双页打印，使 1,8 页和 2,7 页打印于同一张纸的两面，那么上述页面将全部塞入两张纸中，再往中间一折一订，就是一本小册子了◎

文字与段落 (Texts & Paragraphs)

○ 格式 (Formats) 是什么?

➤ “格式”是很普通的用词，即使 Word 也在不同场合用上这个字眼。通常 Word 所说的格式是指对文字字体、段落、制表位、边框、图文框、编号方式的设置。

○ 样式 (Styles) 是什么?

➤ 样式是某种“特定身份的文字”的呈现风格。文档中特定身份的文字（例如大标题、小标题、正文、程序代码、图、表……）必然需要特定的呈现风格（包括字体、字体大小、字距、行距、颜色、前后段距离……）并在整份文档中一以贯之。Word 允许用户将这样的设置储存起来并赋予名称（详见 5.1.1 节），将来即可快速套用于文字身上；配合快捷键使用更是方便（详见 5.1.4 节）。样式是排版的最根本要素，Word 提供的许多自动化功能，例如目录和索引的收集（详见第 7、8 两章），都必须透过样式来完成。有了样式，排版不再是一字一句一行一段的辛苦爬梳，而是大块山水地泼洒（再加上少量细部微调）。

纸张大小

通常，如果您的文档将来要拿到输出中心输出网片，并送至印刷厂印刷，最好是将纸张大小设为将来成品（例如书籍）的大小。如果您的文档并非那么正式或大量，只以打印机输出（再配合复印），那么应该依据打印机所用的纸张来进行设置。

↑ 原始内容



3.1. 纸张大小

通常，如果您的文档将来要拿到输出中心输出网片，并送至印刷厂印刷，最好是将纸张大小设为将来成品（例如书籍）的大小。如果您的文档并非那么正式或大量，只以打印机输出（再配合复印），那么应该依据打印机所用的纸张来进行设置。

↑ 分别套用**标题 2** 样式和**正文**样式后，呈现的风貌

○ 好看的样式如何设计?

➤ “好看”是主观意识。不过相同文化下的审美眼光不会相差太远。我认为好看主要在于视觉舒服，舒服主要在于协调：空间协调、字体协调（详见 2.4 节）。值得注意的是，我的惯用版式 10 年来从没变过，但读者很喜欢。这或许可以给

争妍斗艳、非来点奇怪不可的年轻排版者一些省思。当然我也同意，不同的产品（for 电脑、for 文艺、for 广告、for 宗教、for 哲学……）有不同的诉求。

○ 有内建样式（Built-in Styles）可用吗？

➤ 有，但通常不适用于您（否则就不必排版了），您必须修改其中设置。每个样式都有字体、段落、制表位、边框、语言、图文框、编号方式等 7 种设置，见 5.1 节。

○ 那么内建样式有何价值？

➤ 内建样式（详见 5.1.3 节）极有价值也极重要。虽然其中设置多半不适合用户，需动手修改，但内建样式的名称却是 Word 进行高级排版作业时的依据。举例来说，Word 会收集样式为**标题 1~标题 9** 的文字内容，组成目录，并以**目录 1~目录 9** 样式表现多达 9 层的目录。Word 也会收集文档内的索引项，并以**索引 1~索引 9** 样式表现多达 9 层的索引。诸如此类的应用，不胜枚举。

○ 我已自定许多样式，现在才知道内建样式的好处，如何补救？

➤ 意思是您不想烦琐地逐一将您的自定样式的设置抄录到内建样式中，是吗？有一个办法，就是令内建样式继承您的自定样式，也就是令前者的基准样式为后者。假设您已设计了一个**大标题**，现令**标题 2** 样式继承之：

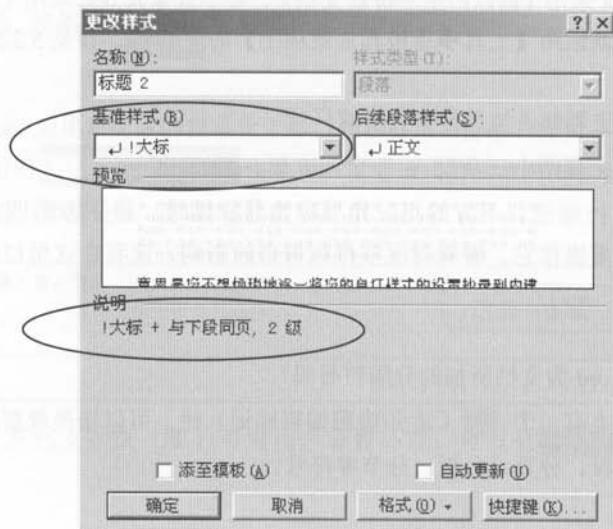


图1-12 令 A 样式继承 B 样式

注意：点击【格式→样式(S)】所得的对话框内，不一样会出现所有内建样

式。您还必须点击所有样式才行：

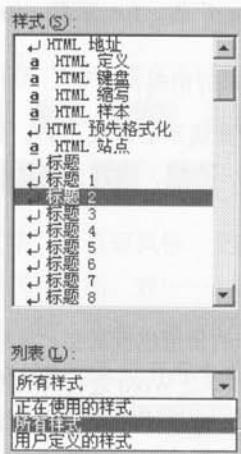


图1-13 列出所有样式（含内建样式）

- 可以删除内建样式吗？
 - 可以在您自己的模板 (.dot) 中删除之，但恐怕不会带来任何好处。

- 如何使用别人建好的样式？
 - 要不就拿其.dot 来用（据以产生一份新文档），要不就拿其.doc 来用（删除其内文字），要不就运用【工具→模板和加载项(I)】的管理器。详见 5.2.5 节。

- 文字段落的缩进和制表位总是令我困惑！
 - 关键在于标尺上那两个三角型 + 一个四方形：  。上方的倒三角型称为首行缩进，下方的正三角型称为悬挂缩进，最下方的四方块称为左缩进。您试着操作它，看看对文字排列带来的影响，比我在里口角春风要好。

- 我可以看到 Word 为文档所加的分隔符号吗？
 - 可以。工具栏上有一个  （显示/隐藏编辑标记）钮，可以切换观察不可见的控制符号如分段、分页、分栏、分节等符号。

- 目录（Table of Contents）可由 Word 自动制作吗？
 - 可以自动化，但无法自动制作。用户可要求 Word 抓取数种样式的文字，做成层

次目录（详见 7.1 节）。本书目录就是以这种方法完成的。

○ 章目录可由 Word 自动制作吗？

➤ 有些书籍会贴心地在各章之前增加一份章目录，对该章提纲挈领一番。制作章目录可以是自动化行为，但无法自动。本书就有章目录，7.5 节有一些经验谈。

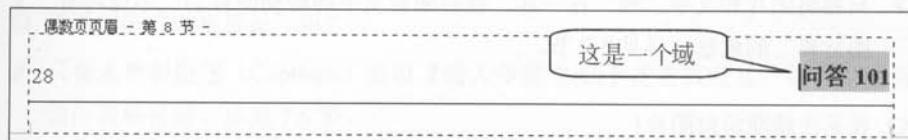
○ 索引（Indexes）可以由 Word 自动制作吗？

➤ 可以自动化，但无法自动。用户可逐一或批量设置所谓的索引项（一种 Word 域），再以简单的动作要求 Word 建立带层次的索引（详见第 8 章）。索引可以是中文（按笔画多寡排列），也可以是英文（按 a,b,c,d… 排列），完全视索引项而定。本书索引就是以这种方法完成的。

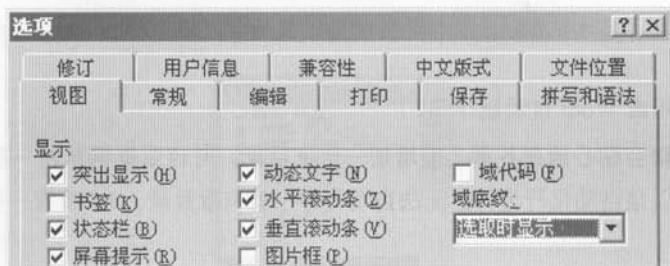
○ 域（Fields）是什么？

➤ Word 可被视为由“核心”和“外壳”构成：外壳时时向核心下命令。Word 核心接受许多 VBA 指令（详见 10.3.1 节）。当我们操作 Word，种种操作就相当于向核心下达指令（甚至可以把各操作所下达的指令记录起来，成为宏（详见 10.1.1 节）。Word 核心对 VBA 指令或宏所做的反应，有时直接呈现于屏幕，有时将运算结果填入某些“空间”。这些空间以编程（programming）术语而言就是变量（variables, fields），Word 称之为域。例如，用户在页眉或页脚所设的页码，就是个域：Word 重新组页后会将每一页的编码填入 Page 域（详见 9.3.1 节）。

凡 Word 为用户所做的自动化工作，诸如页码、目录、索引、文档统计、题注、交叉引用……，都需借助域来完成。试着双击页眉或页脚，您会发现其中有灰色字块，那就是域（的内容显现）：



这些灰色文字块（域）可由用户设置外观：可以是选取时显示、显示、或不显示：



图片 (Figures)

- 图片如何进入文档之中?
 - 两种方式：【插入→对象(O)】和【插入→图片(P)】。前者把图片当做对象对待，并区分为嵌入和链接两种形式。后者有 5 种来源：
-
- 嵌入 (Embedded) 和链接 (Linked) 是怎么回事?
 - 嵌入是将对象或图片整个放进文档内，链接则还放进其他线索（详见 6.3 节）。Word 相关文献上对这两种情况下的压缩和更新有一些描述，但我的实验显示出不同的结果。详见 6.10 节。
 - 图片无法随着文字移动，怎么办!
 - 只要将图片和文字“锁”在一起，就会随着文字的移动而移动，不会再有“文图分离”的尴尬。详见 6.9 节。
 - 我无法精准定位图片!
 - 图片分为“类字符图”和“浮贴图”两种，关系到图片的位置、加框、加阴影的做法。首先您应该彻底了解这两种图的不同。详见 6.9 节。
 - 可否裁剪过大的图？将来可复原吗?
 - 两者的答案都是 yes。详见 6.4 节。

- 过多的图片会不会急速膨胀文件的体积，拖垮 Word 的速度？
 - 非常有可能，尤其是全彩点阵图（bitmap）。因此请慎选图片格式。如果是点阵图，应慎选色彩的多寡——屏幕截图（screen hardcopy）通常只需 256 色或 16 色即有良好表现。本书所用屏幕截图除了极少数例外，都是 16 色点阵图（但操作上需要一些小小注意，详见 6.2 节）。
- Word 会对图片自动进行压缩吗？
 - 如果是点阵图就会！Word 2000+比 Word 97 的压缩能力更强得多。详见 6.10 节。
- 可以制作图目录（Table of Figures）吗？
 - 只要图的题注（Captions）是以【插入→题注(C)】的方式完成，就可以轻松做出图目录。详见 7.6 节。

表格（Tables）

- 文字可以很方便地被转化为表格吗？反之如何？
 - 两者的答案都是 yes。详见 5.7 节。
- 我可以让表格的表头跨页重复吗？
 - 非常简单。圈选表头，然后点击【表格→标题行重复(H)】即可。
- 可以制作“非方阵、非对称”的表格吗？
 - 非常简单。Word 提供极强大的表格制作能力，而且几乎以鼠标便可完成。详见 5.7.4 节。
- 可以制作“表格目录”吗？
 - 只要表格的题注（Captions）是以【插入→题注(C)】的方式完成，就可以轻松做出表格目录。详见 7.6 节。

关于 Word 程序可控性

- 宏（Macros）是什么？
 - 从程序的角度来说，宏是“一些 VBA 指令”的集合。从另一个意义来说，宏是

“吾人操作 Word”的行为集。事实上，Word 用户的各种操作本来就是“对 Word 核心下达 VBA 指令”。

○ 宏可以做些什么事？

- 超乎一般想象！本书举了三个实例：(1) 两岸术语转换(2) 为程序代码加行号(3) 列出文档所用字体，详见 10.4 节~10.6 节。宏是一般用户和超级用户（Power Users）的分水岭。宏还可以用来写病毒（练功可以，千万别害人害己）。

○ 如何制作宏？

- 可录制而得，也可以撰写 VBA 程序而得（详见 10.1.1 节）。

○ 宏如何启动？可制成按钮便利使用吗？

- 宏完成后可透过【工具→宏(M)→宏(M)】+运行(R)启动执行。更方便的做法是将它制成一个按钮，置于 Word 工具栏上（详见 10.1.4 节）。

○ 我写了个宏，换一台电脑后却消失了！

- 您把宏储存在什么地方？文档之中？文档所属的模板（.dot）之中？normal.dot 之中？如果是后两者，您得将相关的.dot 拷贝到另一台电脑，宏才会出现。（详见 10.1.3 节）

○ VBA（Visual Basic for Application）是什么？

- 一种语言，几乎就是 VB（Visual Basic）的翻版。VBA 是 Microsoft Office 软件的可编程接口（Programming Interface）。只要理解 VBA 语法及为数众多的 Word 对象，便可以写 VBA 程序来操控 Word。

○ VBA 很重要吗？

- 取决于您想做什么事。当 Word 提供的现成能力无法满足您的需求，而您又希望解决某些 dirty works（例如稍早说的两岸术语转换、为程序代码加行号等等），恐怕 VBA 是惟一系望。对具备面向对象编程经验的人来说，VBA 很容易上手，只需认识 Word 对象模型即可对 Word 进行程序操控。但对从无编程经验的人来说，VBA 有相当程度的困难。

○ Word 对象模型（Object Model）又是什么？

- Word 是一个以面向对象（Object Oriented）概念建立起来的软件。其核心以对

象 (objects) 的形式开放出来，不仅供给其外壳程序（也就是您所看到的“那个 Word 程序”）使用，也供应给终端用户 (End Users) 使用。这些对象形成一个阶层体系(hierarchy)，其中包罗万象，有 Application, ActiveDocument, Selection, Ranges/Range, Styles/Style, Paragraphs/Paragraph, Words, Characters, Sentences, Fonts/Font, Templates/Template……每一个对象又有自己的 properties (属性) 和 methods (动作)。光会 VBA 并无法对 Word 写程序，还必须熟悉上述这些 Word 对象才行。这整个体系便是所谓的 Word 对象模型。

关于 Word 电子文档

- 有人使用 Word 电子文档来传布文档吗?
 - 不能说没有，尤其在公司或组织内部。因为 Word 不但提供电子文档基本应有的功能（密码、统计数据、书签、超链接），还有很好的团队合作模式，包括批注、追踪修订，很适合用来发展需要集体协作的文档。但是，文档定案之后通常不会以 Word 格式发布出去，一来它没有压缩（虽然它对其内的点阵图有压缩），二来它并非便携式（可移植），保不准在什么电脑环境上会稍微走样。通常最后对外发布时会选择便携式文档格式如 PDF(第 12 章)。第 11 章对 Word 电子文档有详细介绍。
- 书签 (Bookmarks) 是什么？如何插入文档内?
 - 就像真实世界一样，书签是文档内的一个定位点，可从其他地点直接跳过去。点击【插入→书签(K)】即可在当前输入点完成插入书签的动作。详见 11.3 节。
- 超链接 (Hyperlinks) 是什么？如何插入文档内?
 - 超链接的功能近似书签，但跳跃的对象是 URL (Universal Resource Locator)：可能是个网址，可能是个电子邮箱。点击【插入→超级链接(I)】即可在当前输入点完成插入超链接的动作。详见 11.2 节。
- 批注 (Comments) 是什么？如何插入文档内?
 - 批注是团队合作过程中，文档检阅者提出的意见。点击【插入→批注(M)】即可在当前输入点完成插入批注的动作。详见 11.4 节。

○ 追踪修订 (Track Changes) 是什么?

- 顾名思义，就是追踪“文档的修改”。【工具→修订(T)】和【工具→选项(O)+**修订**】都是相关操作。详见 11.5 节。

○ Word 是否支持多人协作？是否提供“各种修订的历史记录”？

- Word 多人协作功能详见 11.4 节“批注”、11.5 节“追踪修订”和 11.6 节“电子文档的保护”。Word 并未提供“各种修订的历史记录”，但提供了不同文档之间的比对功能，详见图 11-11。

关于 PDF 便携式文档

○ PDF 是什么？

- 便携式文档格式：全名 Portable Document Format。“便携式”意指该文档在不同平台上有着完全相同的表现（屏幕呈现）。PDF 是 Adobe 公司的 Acrobat 应用软件的文档格式，具有基本的压缩和保护（密码）能力。

○ PDF 文档该如何展示？如何阅读？

- PDF Reader（阅读器）可用以阅读 PDF 文档；那是 Adobe 公司的软件产品，可自 www.adobe.com 免费下载。

○ PDF 文档如何制作？

- 可使用 PDFWriter (12.5 节) 或 PDFDistiller (12.6 节) 将其他格式的文档转为 PDF。

○ PDFWriter 是什么？

- PDFWriter 是一个打印机驱动程序；以软件模拟打印机，将输入资料（某种文档）转换为 PDF 格式后输出为一个文件。

○ PDF Distiller 是什么？

- Distiller（蒸馏器）也是一个打印机驱动程序，用以拦截其他软件的打印机输出，转而输出为 PDF 文档。用户一旦安装此软件，Word 便多出一个 "Acrobat Distiller" 打印机可选（完全就像 HP 或 Epson 或其他实体打印机一样）。只要选择了它，所有打印输出便流往 PDF Distiller 并转而产生一份 PDF 文档（详见 12.6 节）。

○ Adobe Acrobat 是什么?

➤ Acrobat (特技演员②) 是 Adobe 公司出品的软件, 用来编辑 PDF 文档。它可以以页为单位来增删文档内容、设置页码、制作书签和缩图。虽然 Acrobat 也允许修改页内的文档内容, 但它毕竟不是文字处理软件, 也不是排版软件, 这方面功能较弱。详见 12.7 节。

○ 透过 Adobe Distiller 产出 PDF 文档, 其形貌是否与原件相同?

➤ 按理想应该保持完全不变。但现实情况很难说, 毕竟软件是人写的, 人难免犯错, 臭虫难免会有。就我个人经验, 就发生过意外, 详见 12.6.7 节。

○ PDF Reader 或 Adobe Acrobat 的导览窗格内的书签如何设置?

➤ 所谓导览窗格, 其中列出书签, 可带来超链接效果, 通常被用来作为文档的目录使用, 类似 Word 的文档结构图。我们可以手动制作书签, 亦可使用 PDFMaker 自动化抽取某种 Word 样式 (的文字内容) 做成书签。详见 12.8 节。

○ PDF 是便携式文档, 馈给网片输出机直接制版最好了。可以吗?

➤ 只要输出机支持 PDF 格式, 就可以。

本书所有版式与样式细节

也许您对本书所使用的字体、版式、样式、图片感兴趣。以下列出所有细目。

表 1-1 本书各种组成描述

细目名称	细目	说明
字体	华康简宋、宋体、 方正隶书简体 、 方正粗圆简体 、 华康简黑、 Arial 、 Arial Black 、 Book Antiqua 、 Courier New 、 Lucida Console 、 Lucida Sans 、 Lucida Handwriting 、 Times New Roman 以下字体也用到, 但只在表 2-3 和表 2-4 作为展示: Cataneo BT 、 Comic Sans MS 、 Footlight MT Light 、 Verdana 、华康简仿宋、华文行楷、华文琥珀、华文新魏、华文中宋、华康简综艺、方正粗倩简体、 方正大黑简体	本书用到的所有字体, 乃以宏取得。宏代码见 VBA10-21

细目名称		细目	说明
版面设定	纸张大小	185×230 mm (大 18 开)	
	边界	上 2.3cm, 下 1.4cm, 左 3.25cm, 右 3.25cm, 装订边 0cm, 距边界: 页眉 0.95cm, 页脚 0.6cm	
屏幕截图 hardcopy		除极少数外, 都是 16 色点阵图 (bitmap)。 需调整【桌面→属性→外观】之窗口标题栏为纯色, 并且不使用渐变色, 才能在 16 色模式下获得好效果。	
矢量图		所有卡通画都是从 Microsoft Office 所附之 "Microsoft 剪辑库 5.0" 取得	两张例外, 手绘扫描
目录		四个目录: (1) 章节 (五层) ; (2) 图、(3) 表、(4) VBA	制作方式如第 7 章
索引		三层、双栏	制作方式如第 8 章
节		每个单元一个节。 其中某些“与前一节同”, 另一些不同。	
页眉设置		左右页不同。页码均在外侧。 右页内侧为大标 (标题 2) 名, 左页内侧为章名。	
页脚设置		左右页不同。英文书名在右侧。	
项目符号		使用 ■、○、●、➤ 四种符号	
宏		四个: (1) 两岸用语转换 (2) 为程序代码加行号 (3) 搜寻所有字体 (并可找出特定字体出没处) (4) 列出所有样式的文字说明 (用于表 1-2)	
本书所用 题注标签		图、表、VBA	三者皆自定

※以下两种样式类型，**↓** 表示段落样式，**a** 表示字符样式。由于写作本书之前我已有许多惯用的自定样式（名称都带有前缀字**!**），也已设置好快捷键，因此多半让内建样式继承既有之自定样式，如此一来便可免除一大堆重设工作。

表1-2 本书页面设置与样式细目

类型	样式名称	细目	说明
↓	正文	字体：(中文) 华康简宋，(默认) Times New Roman, 9.5 磅，(中文) 中文(中国)，(其他) 英语(美国)，两端对齐，行距 最小值 16 磅，段落间距 段前 4 磅 段后 8 磅，正文文字，允许标点溢出边界，自动调制中文与数字(西文)的间距，字体对齐方式：基线对齐	套用于全书正文身上
↓	!自定正文 Alt+c	正文 + 缩进：首行 2 字符	有快捷键
↓	!章名 Alt+m	!大标 + 字体：24 pt, 字符缩放 120%，右对齐，段落间距 段后 30 pt	章名 有快捷键
↓	标题 1	!章名 + 缩进：悬挂 0.5 cm, 与下段同页， 1 级 多级符号 表示会自动编号	用于章名 第 1 层标题
↓	!大标 Alt+l	正文 + 字体：(中文) 方正粗圆简体，(默认) Footlight MT Light, 四号，加宽 0.4 磅，字符缩放 110%，段落间距 段前 6 磅 段后 3 磅	大标题。 有快捷键
↓	标题 2	!大标 + 缩进：悬挂 1 cm, 与下段同页， 2 级 多级符号	第 2 层标题
↓	!中标 Alt+2	正文 + 字体：(中文) 方正粗圆简体，(默认) Footlight MT Light, 12 pt, 加宽 0.4 磅，字符缩放 110%，段落间距 段前 10 磅 段后 2 磅	中标题。 有快捷键
↓	标题 3	!中标 + 缩进：悬挂 1.5 cm, 与下段同页， 3 级 多级符号	第 3 层标题
↓	!小标 Alt+3	正文 + 字体：(中文) 华康简黑，(默认) Footlight MT Light, 10 磅，加宽 0.4 磅，字符缩放 110%，段落间距 段前 3 磅	小标题 有快捷键
↓	标题 4	!小标 + 段落间距 段前 6 pt 段后距离 4 pt， 与下段同页，4 级	第 4 层标题
↓	标题 5	!自定正文 + 缩进：首行 0 cm，段落间距 段前 1 pt 段后 1 pt，与下段同页，5 级	Q/A 提问。 第 5 层标题

类型	样式名称	细目	说明
↔	!目录每行	正文 + 行距 最小值 10 磅, 段落间距 段前 2 磅 段后 2 磅	
↔	目录 1	!目录每行 + 字体: (中文) 华康简黑, 10.5 pt, 段落 间距 段前 6 磅, 图案: 10%, 自动更新	第 1 层目录 搭配 标题 1
↔	目录 2	!目录每行 + 字体: (中文) 华康简宋, 不作校对, 缩 进: 左 0.5 厘米, 制表位: 11.98 厘米 右对 齐 …, 自动更新	第 2 层目录 搭配 标题 2
↔	目录 3	!目录每行 + 字体: (中文) 华康简宋, 不作校对, 缩 进: 左 1 厘米, 制表位: 11.98 厘米 右对齐 …, 自动更新	第 3 层目录 搭配 标题 3
↔	目录 4	!目录每行 + 字体: 9 pt, 不作校对, 缩进: 左 1.5 厘米, 制表位: 11.98 厘米 右对齐 …, 自动更新	第 4 层目录 搭配 标题 4
↔	目录 5	正文 + 字体: 9 pt, 不作校对, 缩进: 左 1.5 厘 米, 行距 最小值 10 磅, 段落间距 段前 2 磅 段后 2 磅, 制表位: 11.98 厘米 右对齐 …, 自动更新	第 5 层目录 搭配 标题 5
↔	!索引文字	正文 + 字体: 8.5 pt, 行距 最小值 11 pt	
↔	索引 1	!索引文字 + 缩进: 悬挂 1 厘米, 段落间距 段前 2 pt 段后 0 pt, 自动更新	搭配 第 1 层索引
↔	索引 2	!索引文字 + 缩进: 左 0.3 厘米, 悬挂 0.7 厘米 左 对齐, 段落间距 段前 0 pt 段后 0 pt, 自动更新	搭配 第 2 层索引
↔	索引 3	!索引文字 + 缩进: 左 0.6 厘米, 悬挂 0.4 厘米, 段落间距 段前 0 pt 段后 0 pt, 自动更新	搭配 第 3 层索引
↔	!表格前置 样式	正文 + 行距 最小值 12 pt, 段落间距 段前 3 pt 段后 3 pt	用以设置表 格的 row 高
ⓐ	默认段落 字体	下段样式的字体 + 和 p.253 的问题有关	
ⓐ	脚注引用	默认段落字体 + 字体: (英文) Arial Black, 11 pt, 上标, 蓝色 粗壮字+蓝色, 以求醒目	位于正文中 的编号
↔	脚注文本	正文 + 字体: 9 pt, 行距 最小值 12 pt, 段落间距 段前 0 pt 段后 0 pt	位于页末的 文字

类型	样式名称	细目	说明
a	尾注引用	脚注引用	位于正文中的编号
↓	尾注文本	脚注文本	位于章末的文字
a	超级链接	默认段落字体 + 字体: (英文)Times New Roman, 9.5 pt, 没有下划线, 蓝色	带超链接者
a	已访问的 超级链接	超级链接, 紫色	带超链接者
↓	题注	正文 + 居中, 段落间距 段前 0 pt 段后 4 pt	
a	!程序代码 alt-L	正文 + 字体: (英文)Courier New, 9 pt, 缩进: 左 0.3 cm, 行距 最小值 11 pt, 段落间距 段前 0 pt 段后 0 pt	有快捷键
a	!变量 alt-v	默认段落字体 + 字体: (英文)Courier New, 9 pt,	例: var1
a	!类名 alt-x	!变量 + 字体: (英文)Lucida Console (本书用之于 10.3 节 Word 对象名称)	例: Styles
a	!术语 alt-z	!变量 + 字体: (英文)Arial, 9 pt (本书用之于 10.3 节 properties 名称)	例: Count
a	!书名 alt-p	!殊词 + 字体: (英文)Book Antiqua, 9.5 pt, 斜体	例: Art of Word
↓	!条列 alt-b	!自定正文 + 缩进: 左 0 cm 悬挂 0.5 cm, 段落间距 段前 1 pt 段后 1 pt, 首行 0 cm, 无编号 日后再指定编号	
a	!按钮 alt-shift-z	默认段落字体 + 字体: 9.5 pt, 图案: 25%, 边框:(带阴影单实线, 灰色-50%, 0.5 pt 行宽)	例: 请点击 确定
a	!组件 alt-shift-q	默认段落字体 + 字体: 9.5 pt, 图案: 20% (自动设置前景, 白色 背景), 边框:(无框线)	例: 请在属性页的列表(L):任选一项
a	!对话框 alt-shift-s	默认段落字体 + 字体: 9.5 pt, 图案: 10%, 边框:(双实线, 自动设置)	例: 请点击 打开对话框

类型	样式名称	细目	说明
a	!域 alt-=	默认段落字体 + 字体: (英文)Lucida Sans, 9 pt	例: StyleRef 是域
a	!高亮 alt-4	默认段落字体 + 字体: (中文) 华康简黑, 9.5 pt, 字体颜色: 灰色-50%	例: Word 是个排版软件
a	!样式名称 alt-shift-0	默认段落字体 + 字体: (中文) 方正隶书简体, (英文)Lucida Sans, 10 pt, 紧缩 0.3 pt, 图案: 12.5%	例: !自定义是个样式
a	!页签 alt-shift-w	默认段落字体 + 字体: 9.5 pt, 边框:(带阴影点式, 灰色-50%, 0.75 pt 行宽)	例: 请点击多级符号
a	!word 指令 alt-7	默认段落字体 + 字体: (中文) 华康简黑, 蓝色	例: 交叉引用是很棒的写作辅佐

2. 总论	59
2.1. 排版，被个人电脑完全取代的行业	59
2.2. 个人出版思潮.....	67
2.3. 排版基本概念.....	68
2.3.1. 最佳作业模式.....	70
2.3.2. 文档结构视图.....	71
2.3.3. 页面视图.....	72
2.3.4. 大纲视图.....	74
2.3.5. Web 视图.....	76
2.3.6. 主控文档视图.....	77
2.3.7. 细说页面元素.....	79
2.3.8. 细说排版要件.....	83
2.3.9. 心得偶拾.....	84
2.4. 版面美学之我见.....	88
2.4.1. 字号、字距、行距.....	88
2.4.2. 字体、色彩不要过于丰富.....	89
2.4.3. 斜体中文字的杀伤力.....	90
2.4.4. 中英文字体的调和搭配.....	91
2.4.5. 图片的大小.....	92
2.4.6. 不要每一页都出血.....	92

2

总论

2.1. 排版，被个人电脑完全取代的行业

我的第三文书中心经验（1984-1985）

1984~1985 我在金门服役。当时在后勤支援指挥部担任幕僚，兼管金防部第三文书中心。这是一个任务繁重的单位，每天有大批公文，各式各样的签呈、命令、日报表、季报表、月报表……从防区上上下下各单位涌来这里。重要文档交付打字，一般文档交付缮写。

我们有 6 名打字兵，每人操作一台中文打字机。还有 4~5 名缮写兵，每人操作一枝钢针笔和一块钢板。不论打字或缮写，都作用在一种蓝色蜡纸上：打字机铅字的重力锤击，或钢针笔的用力刮勒，可以除掉蜡纸上的蜡。这样的蜡纸上了油墨之后，便能让油墨从去了蜡的字迹中渗透过去，沾染在白报纸上，成为一份份油印品。

那时的中文打字机，是占据整个标准铁壳办公桌面的庞然大物。桌面上有一大盘一大盘铅字，桌上竖立的纸夹夹着铅字分布表，供新手学习或老手参考，也夹着等待被打字的公文稿。操作员先以左手将“铅字镊”（一整套重机具，包括轴承、连杆、滚轮等等）沿着水平面移到适当的字盘上方，再以右手操作“铅字镊”，像敲键盘似地（动作大 10 倍，要动用掌长肌、桡侧屈腕肌、尺侧屈腕肌、伸指肌、外展拇指肌、尺侧伸腕肌、肱二头肌、三角肌，和肩胛力量¹），兔起鹘落，以铅字一上一下的重击力敲掉蜡纸上的蜡。铅字镊的移动、铅字的拣选起落，以及敲击点的逐次移动，由一整套机械装置连动着。如果委印内容是陌生的格式（这种情况不常见，公文长相经常千篇一律），还得先用钢针笔在蜡纸上划出新表格。

¹ 总之就是整支手臂啦。

熟练掌握这样一套中文打字设备，以及铅字的排列，需要相当不短的训练时间。即使是熟手，平均也得大约 3 秒才能敲出一个中文字。所以打字兵通常从三年义务役员额中挑选（当时的一般义务役期是二年，但有特殊的三年役期。中奖与否完全看手气）。

至于缮写兵，那就更辛苦了。每人每天不停地写、写、写。字迹不但必须保持端正秀丽，而且还要入木三分，才不会因为括痕过轻而影响油墨的渗透。眼睛也得一直盯着深蓝色微微反光的蜡纸。一天工作下来，每一名缮写兵的内缩拇指（大拇指和食指用力并拢后手背拱起的一小块肌肉）和眼睛的负担可想而知。

我在第三文书中心的所见所闻，反映当时粗糙的排版、制版、油印²的境况：机器成本昂贵、人力成本昂贵、学习成本昂贵、品质一般般、弹性与应变力极低。

我的 90s 年代出书经验（1991-1995）

1991 我完成第一本著作《虚拟记忆体》，而后 5 年又陆续完成近 10 本作品。每一次书稿完成，我便要求进驻出版公司，花两三天的时间和排版人员长相左右。

那时的作业方式是，由排版人员依循出版公司的一贯风格，以电脑进行一个段落一个段落的文字处理³，然后以激光打印机输出到铜版纸（效果才会好），再由美工人员以剪刀剪出一段段文字，胶粘在特殊方格纸的适当位置，上覆透明蜡纸，再交给我（作者）校对。虽然校对前的所有动作都不是我的责任范围，有时候还是必须向工作人员献个殷勤，以示感谢与体贴。如果校对有所得，理直气壮在透明蜡纸上以红笔圈点即可。怕就怕在对版式不甚满意，或临时想增减内容。这时候就得有看晚娘面孔的心理准备。偶而献个咖啡茶点对于改善紧张关系不无帮助。



经过这样往往返返，成品是大约 10cm 高的一叠纸（单面粘贴）。接下来进入照相制版。照相品质（包括人、光线、器具）会影响制版品质，也就连带影响印刷品质，不过那已不是我管得到的范围了。

² 这里是指一般人可能接触的（非精密）油印。适用于学校试卷、补习班讲义、部会公文、团体名单清册等等。

³ 那时候的排版软件功能不足，一小段一小段的文字处理尚可承受，真枪实弹的排版就难以胜任了。

俱往矣！

不论是前述哪一个阶段，所有工作量估计如今只需一台 PC、一套适当的排版软件、一个员额的人力，就可全部搞定。首先，十指运行的打字速度比起刻钢板，速度快 n 倍，比起中文铅打，速度也快 n 倍。其次，任何文档若有任何格式上的变化，现今排版软件都可以轻松应付。现今排版软件可以不耗用一张纸，在电脑屏幕上直接呈现最终完稿结果。电子文档，甚至可以连接输出机，直接制版。

当桌上电脑和桌上排版软件的价格愈来愈便宜，功能愈来愈强时，排版这一行业日益被扫到历史的一角。如今除了大型文档的排版还倚赖行业人士，任何人都可以在浅浅学习诸如 Word 之类的易上手软件后，不假他人就在自己的办公室或教室完成一份份精美的小型报告。甚至小学生都可以完成 10~20 页图文并茂的作业。



但是 99% 的 Word 用户只把 Word 当做“文字处理器”（word-processor）来使用，没有认识到它处理大型文档的正规排版能力，更没有认识到 Word 作为一个文档创作平台对创作者带来多大的方便。这些高级操作（相对于低级的文字处理）并不困难，概念也颇为简单。一般人的不理解只因为缺乏教育资源，缺乏足够的好书，完整讲述 Word 作为排版软件与创作平台的能力。一旦这部分的学习资源被开发出来，人人可以花 2~3 天的时间释放手上那个文字处理器的能量，使它升级为专业排版软件。

自动化、电脑化、技术密集化、个人化，是必然要走的排版趋势。任何个人没有办法抗拒历史潮流。瓦特发明蒸汽机，引爆工业革命，虽然造成许多工人失业，但时代巨轮的航向不可改变，咱们也不必妇人之仁地想要为此常驻旧世界。工业革命对全人类的价值，毕竟远超过当年工人失业连带而来的动荡。计算机的发明引爆了信息革命，现在谁又能够想象没有电脑的日子该怎么过？同样道理，排版这一行业，一定会因为个人电脑的极度普及，以及平价易操作之排版软件的极度盛行而衰落⁴（这种情况事实上已经发生了）。当每一个人能够按自我意志完成自己的大型文档的最终形貌时，节省下来的协调与往返、乃至生命的折损，肯定远大于学习排版软件的成本（当然首先这人有大型文档的排版需求）。

⁴ 排版行业“完全”消失是不至于的。社会分工日益精密，当然存在对大型文档排版有所需求，但没时间或没意愿实际操作排版的人。

也许本书能够为“将排版行业推往生命尽头更近一点儿”以及“人人可随心所欲完成自己的文档”尽一丁点儿力。果真如此，我将感到骄傲。

我的 21 世纪出版作业模式（2003-date）

我是一个资讯教育工作者，每年完成大量的大型文档（书籍和长文章）。目前，当我进行创作或翻译工作时，我的工作方式是，先以 Word 设好我喜欢的页面设置和各种样式⁵，做成模板，然后在这样的“空壳”（详见下页）中创作或翻译每一本书籍（或文章）。有了文档结构图所提供的大纲提示和快速定位协助（图 2-1），以及图表题注的自动编列与更新（5.10 节），以及随着文档的进行与变动而自动修正引用点的交叉引用（5.12 节），创作一事，竟从原本“白了少年头”的繁琐，变成了只需关心创作内涵的一种愉快过程。各种繁琐问题，如文前⁶的 i, ii, iii 编页与正文的 1, 2, 3 编页（3.4.2 节）、页眉放置章节名称与页码（第 4 章）、正文形貌（第 5 章）、图表形貌（5.7 节、5.8 节、第 6 章）、各章奇数页起头（3.4.1 节）、目录自动化生成（第 7 章）、索引自动化生成（第 8 章）……都不必再费心神。创作过程中我所看到的，就是读者最终看到的形貌。

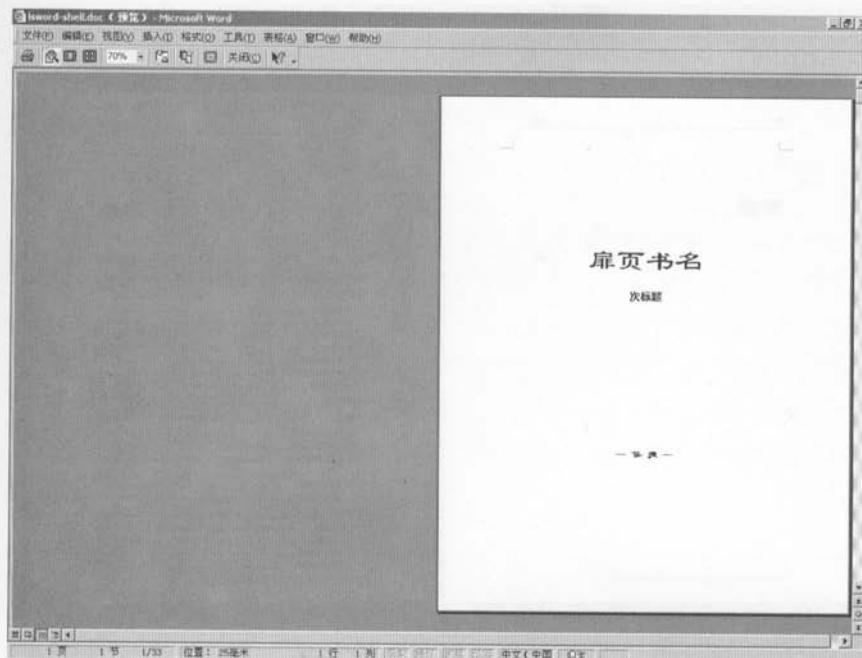
一旦创作完毕，书籍或文章已经完全按我的意思排版妥当。此时可操作 Acrobat 将 Word 文档转制为 PDF 文档（并自动制作书签），然后将 Word 文档和 PDF 文档⁷传送给出版公司，直接制版/输出网片，交付印刷厂。

下面便是我所使用的“壳”。此处呈现的是 Word 打印预览画面。实际写作时，只要将内容填入即可。

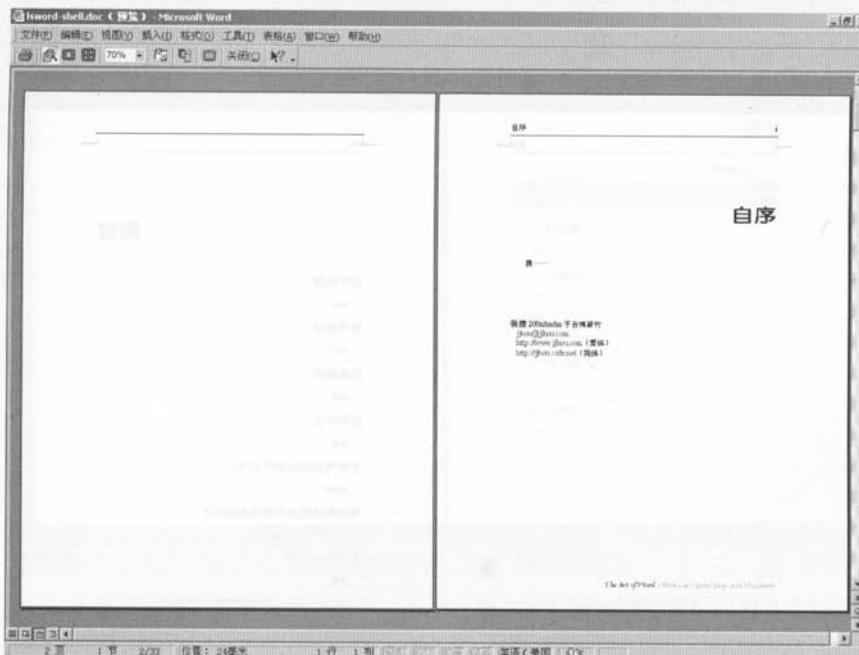
⁵ 这些样式既然是自己喜欢的，也就不会频繁改变。事实上这么多年来我没有改变过我的书籍的版式风格，而这一点又深受读者的喜爱。

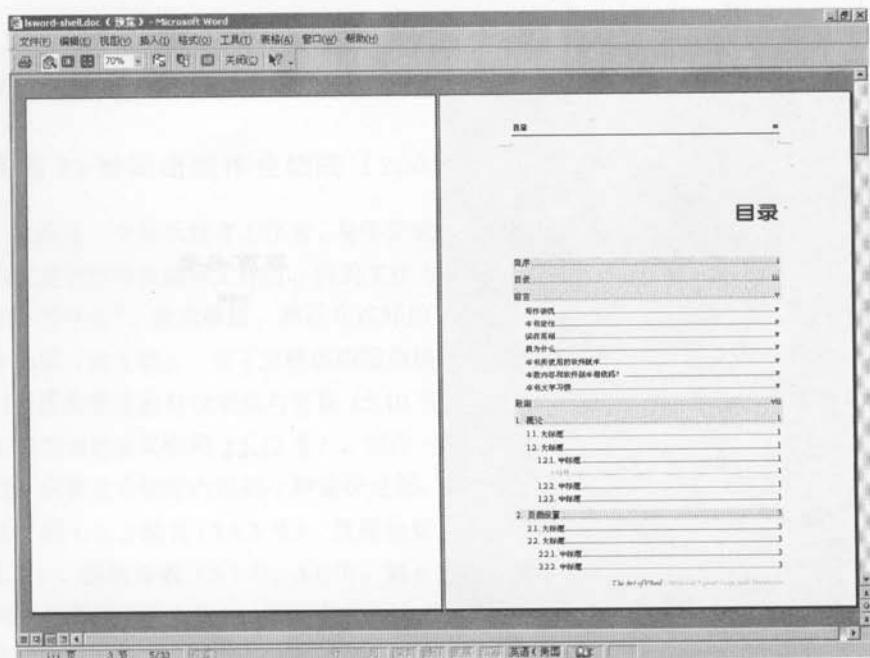
⁶ 文前是指第一章之前（目录、前言之类），文后是指最后一章之后（附录、索引之类），其余的就是正文。这在第一章“问答 101”中介绍过。

⁷ 有些制版厂接受 PDF 文档（这是最完美的结局），那就不需另附 Word 文档。有些制版厂只接受 Word 文档，而由于 Word 文档在不同机器上可能会有一点点字距上的偏移（见 p.331），造成版面可能会有一点点跑位，此时最好附上 PDF 文档供校对参考。

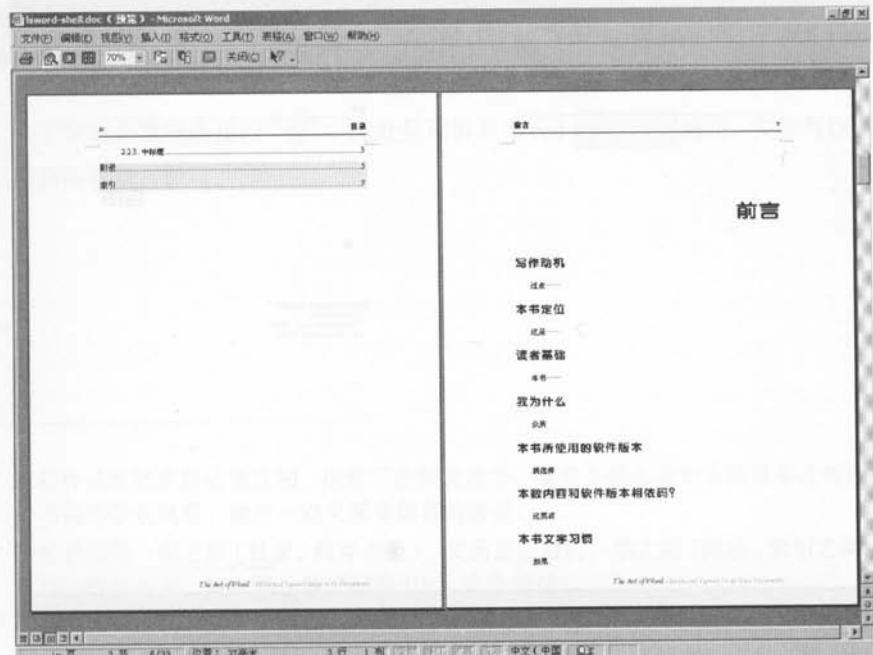


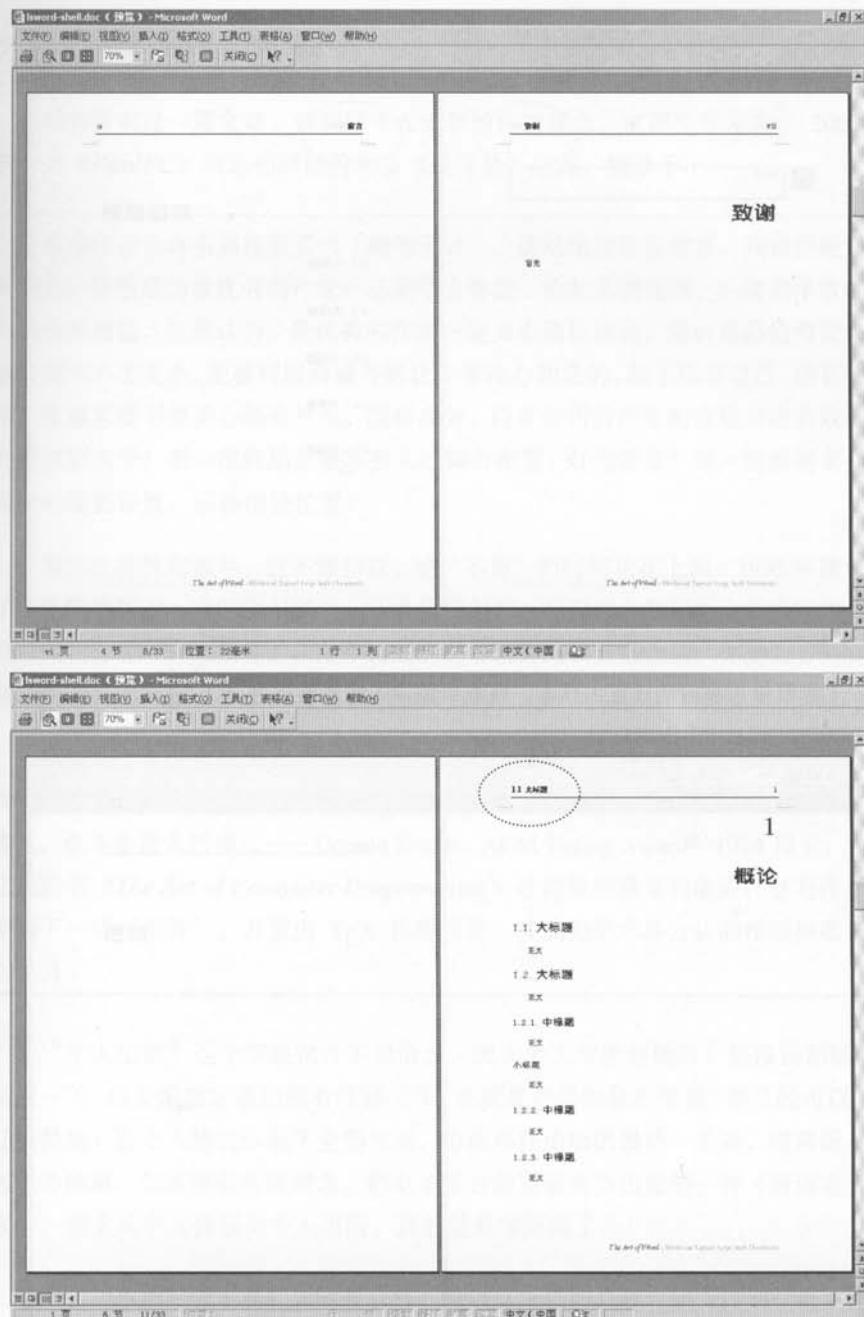
↑ 空壳第一页



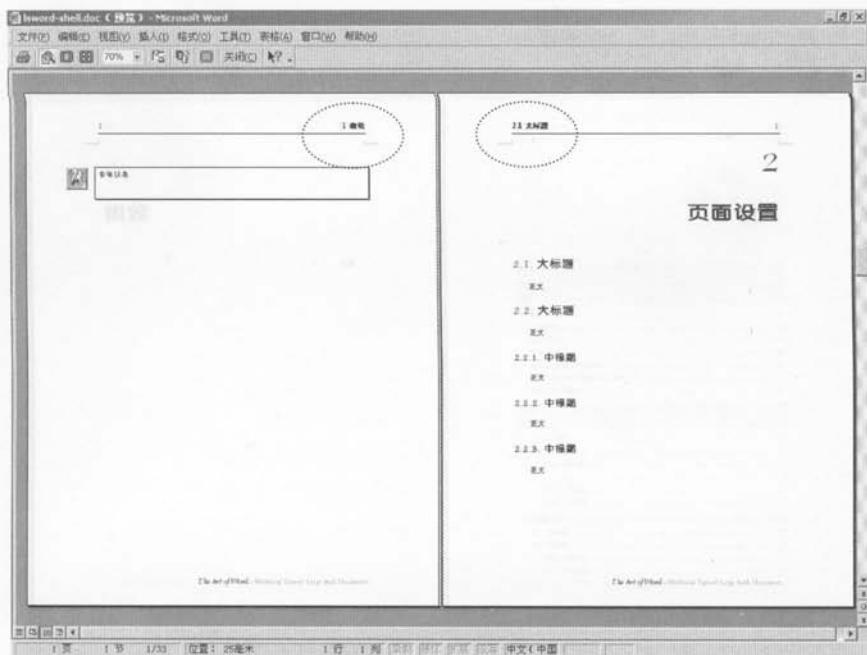


↑注意，前图的“自序”目前只一页，但本图的目录仍从奇数页（右页）开始。中间一张空白。目录由章名和大中小标题自动化构成，随时可以轻松更新（详见 7.3 节）。





↑此前的奇数页页眉内侧是章名，本图（第一章）起相同位置放的则是大标题。此前采用罗马数字编页，第一章起采用阿拉伯数字编页。



↑ 奇数页页眉内侧放置大标题，偶数页页眉内侧放置章名。



↑ 索引目前空白。只要用户设好索引项（8.1.1 节），Word 便可自动完成索引。

2.2. 个人出版思潮

我曾经有过一篇文章，谈到对个人出版的排版理念。这篇文章发表于 2002/03 的台北《Run!PC》杂志和同期的北京《程序员》杂志，摘录于下：



有些作者会将书稿排版妥当（酬劳另计），这对出版社很省事。我曾经提过一个论点：要想成为最优秀的作家，必须学会排版。听起来很荒谬，好像要学者弯腰处理柴米油盐。但我认为，最优秀的作家一定关心最后成品，最后成品的可读性和版面脱离不了关系。能够将版面操弄到让作者称心如意的，除了作者自己，还有谁？第一流画家要不要关心画布材质、颜料成分、两者作用后产生的渲染或收敛效果？乃至装裱水平？第一流舞蹈家要不要关心舞台布置、灯光效果？第一流演奏家要不要关心现场音效、乐器摆设位置？

写作生涯的前数年，我不懂排版，也“不屑”把时间花在上面。1995 年我经历了一次惨痛教训，决定学习排版，结果受用无穷。回顾过去与排版人员之间的沟通和拉锯，仿佛一种生命折磨。至今我的所有书籍都由我自己排版，并且无心插柳地成就个人版面风格（很多人喜欢我的书籍版式）。

我相信至今还有很多人对于“自己的书自己排版”这种论调嗤之以鼻，甚至讥为傻瓜：“写作才是正务，内容决定一切，排版算什么？”唔，侯捷绝不是第一个傻瓜，也不是最大的傻瓜——Donald Knuth，ACM Turing Award⁸ 1974 得主，为了让他的书《The Art of Computer Programming》达到他所满意的版面，在写作过程中搁下一切“正务”，开发出 T_EX 排版软件，并成为学术界公认的排版标准格式与工具。

“个人出版”这个字眼也许不很恰当，因为个人所能够做的，还没到制版+印刷这一关。但是出版之前的所有完稿工作，也就是产品的最后形貌，都已经可以（并且轻易地）在个人排版作业下全部完成。由此再往出版的最后一关走，道路既不遥远也不崎岖。如果突破传统观念，把电子平台的发表视为出版的一种（所谓电子出版），那么从个人排版到个人出版，真的就是零距离了。

个人出版的时代来临，意味着任何人都可以自己的意志打造自己的产品。任何时候皆可进行，任何地点皆可进行。您的作品有没有被阅读的价值，那是一回事儿，

⁸ Turing Award，图灵奖，计算机界的诺贝尔奖（Nobel Prize）。

您能够按自己的意志完成自己的作品，才是“个人出版思潮”的重点。

2.3. 排版基本概念

相对于简单的文字处理，以及兴之所至的涂涂抹抹，排版是一个高度组织化与结构化的工作。换句话说，如果您没有按照排版软件的要求，事先（或过程中）设置必要的样式，您就不能享受许多自动化功能：自动产生图表题注（编号）、自动产生目录、自动产生索引、自动产生文档结构、自动切换页码形式、自动更换页眉内容……

以下，我要介绍一些基本概念，和一些大局视野。

何谓专业排版

我个人简单地认为，能够方便地让用户完成图 0-2 的所有文档组成和图 0-3 的所有页面元素，斯可谓专业排版。

就强项领域而言，专业排版软件还有所划分。例如可区分为擅长彩色印刷（如 PageMaker, Publisher）或适合单色灰阶印刷（如 Word），组页排版（较不方便，如 PageMaker 或 Publisher）和页面能力高强者（较方便，如 Word），以及是否允许用户撰写宏更进一步操控软件功能（如 T_EX、Word）。

何谓便携式电子文档

我个人简单地认为，所有可播放或显示、储存于电子设备、由电子媒介传递的文档，都是电子文档。至于便携式（portable）的意思则是，文档形貌在任何设备上都不变形，保持绝对一致。Word 文档无法达到可携性，而 PDF 文档可以。后者的全名是 Portable Document Format（便携式文档格式）。

何谓样式

样式是 Word 排版工程的灵魂（其他排版软件也是如此，称谓或许不同）。Word 的每一项自动化工程，都根据用户事先规划的样式完成。所谓样式，就是用以呈现特定页面元素（扉页、题献、正文、页眉、大标、中标题、小标题、章名、图说、脚注、目录、索引）的一组格式（字体、字距、行距、特殊效果、对齐方式、缩进位置……）。例如：

纸张大小

通常，如果您的文档将来要拿到输出中心输出网片，并送至印刷厂印刷，最好是将纸张大小设为将来成品（例如书籍）的大小。如果您的文档并非那么正式或大量，只以打印机输出（再配合复印），那么应该依据打印机所用的纸张来进行设置。

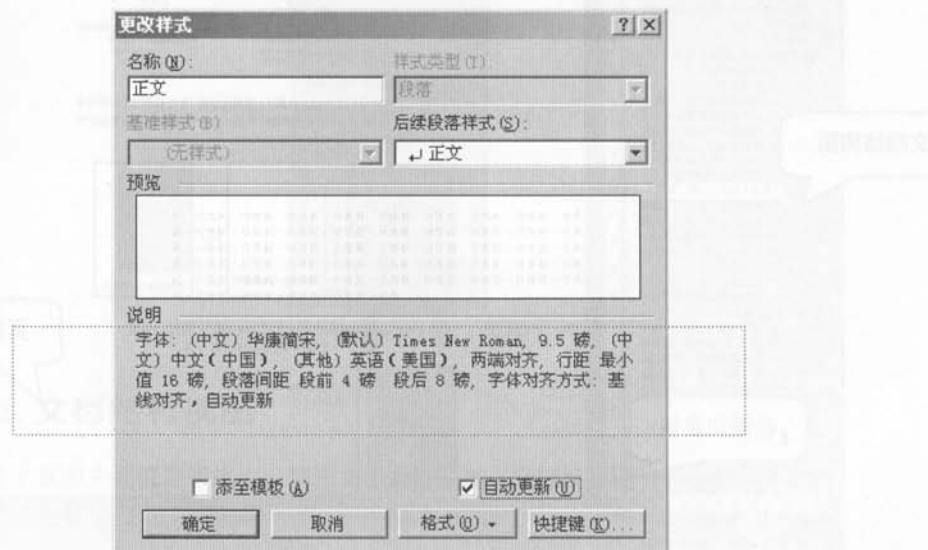
上述之原始内容套用**标题 2** 样式和**正文**样式后，呈现的形貌如下：



3.1. 纸张大小

通常，如果您的文档将来要拿到输出中心输出网片，并送至印刷厂印刷，最好是将纸张大小设为将来成品（例如书籍）的大小。如果您的文档并非那么正式或大量，只以打印机输出（再配合复印），那么应该依据打印机所用的纸张来进行设置。

上例的**正文**样式设计如下（详见 5.1 节）：



每个样式必须拥有一个专属名称。此前所说的 Word 自动化工程便是根据特定（亦允许用户指定）的样式名称进行。例如 Word 可抽取样式名称为**标题 1**、**标题 2**、**标题 3**……的文字作为目录（第 7 章），亦可抽取 XE 域（indeX Entry，索引项）所在页码，做成以**索引 1**、**索引 2**……样式为形貌依据的索引（第 8 章）。5.1 节详细说明如何设置样式。

何谓创作平台

我个人简单地认为，凡是能够在文档创作过程中，为创作者减少琐屑工作，使创作者的心力尽可能完全集中于“创作”这件事上，这样就是一个创作平台。所谓琐屑工作包括章节和图表的自动编号，目录和索引的自动化，页眉页脚的内容变化，正文之间的彼此交叉引用等等。

2.3.1. 最佳作业模式

下面是我个人认为最棒的 Word 排版作业模式，用于文档创作或排版都很理想。以下所称的文档结构视图、页面视图和大纲视图等等都是所谓的视图，是文档的某种表达形式，是同一份文档以不同角度解读所得的不同画面⁹。

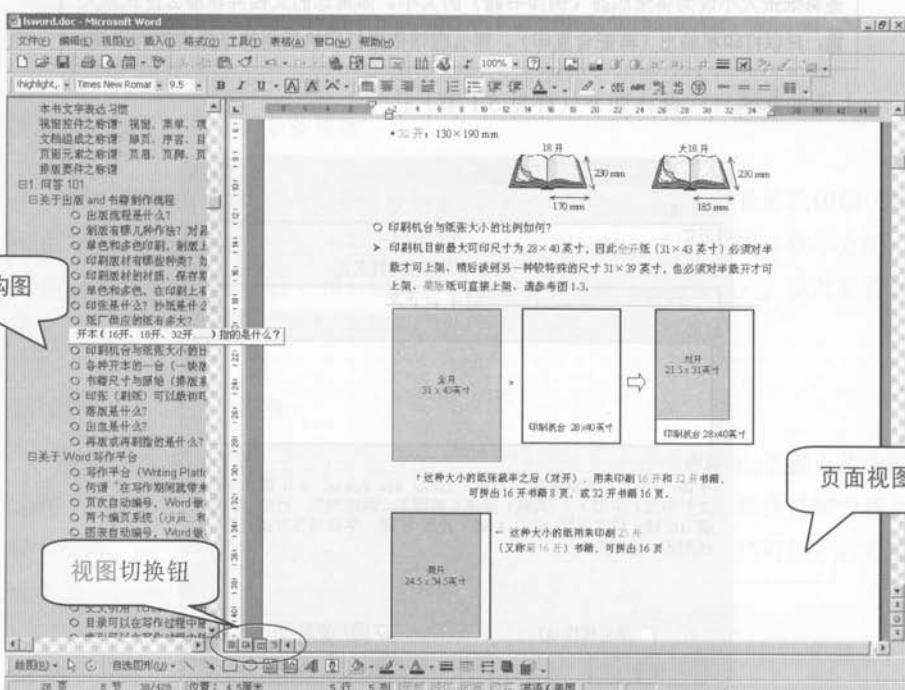


图2-1 文档结构图 + 页面视图: Word 最佳作业模式

⁹ “一份实体、多个观察角度”在软件技术上就是所谓的 Document-View (文档-视图) 概念。这在 MFC programming 中十分出名。Word 是不是以 MFC (Microsoft Foundation Classes) 写出来的？唔，我不知道。

图 2-1 左为文档结构图，右为页面视图视窗。点击左框中的任何标题便可使右窗之光标（键盘输入点）移至对应的正文位置。左框大小可调整，如果标题过长，会自动延伸（如上图左框之黄色条状延伸）。右视窗亦可设为大纲视图（图 2-3）或 Web 视图（图 2-5），它们之间的切换可使用【视图→页面(P)】、【视图→大纲(O)】或【视图→Web 版式(W)】进行之，更简便的办法是以图 2-1 左下角之小按钮进行切换。

图 2-1 左侧文档结构图之开启与关闭，可由【视图→文档结构图(D)】完成，亦可直接于框中按鼠标右键，并以图 2-2 快捷菜单的文档结构图(D)关闭之。结构图中的标题可折叠：标题前的符号 表示可展开，标题前的符号 表示可收合。结构图的大小可经由鼠标的拉动而改变。



图2-2 文档结构图的右键（快捷）菜单

2.3.2. 文档结构视图

这个视图主要就是带出上一节图 2-1 中的左框（文档结构图）。这种视图的开启、关闭和呈现见上一节说明。

这种视图最棒的特点就是，藉由结构图的阶层式标题，创作者或排版者（乃至读者）可对整份文档取得一目了然的效果。阶层式标题由 **标题 1**、**标题 2**、**标题 3**……等样式名称¹⁰组成，可收合亦可展开，操作极为方便。每一个标题（不

¹⁰ 这些都是内建标题。用户亦可自定标题名称取代之。详见 5.1 节。

论第几层) 都有超链接(hyper-link)效果：点击标题便可直接进入右视窗的正文对应位置，进行创作或编辑。这种便利性简直是为老虎加上一副翅膀！

文档结构图本身无法编辑，它只是文档的另一种呈现。任何编辑动作均需于稍后介绍的页面视图或大纲视图中进行。右视窗增减任何**标题1、标题2、标题3……**样式，均会自动而立即地反映于结构图中。



但偶而有时候结构图会“突槌”(有程序设计经验的人都能够理解这一点⑪)：它无法即时反应文档的变化。这时请不要怀疑你自己，请怀疑Word！将引导框收合展开一两次，多半就能让它从浑浑噩噩中醒来。



一般情况下我们可以拉动文档滚动条最上方的“小方块”，形成上下各有滚动条的两个子视窗，因而可以展现同一份文档的两个不同部位。这在某些时候是很有用的。但如果使用文档结构视图，就无法使用上述分裂视窗，因为“小方块”不见了。

2.3.3. 页面视图

关于页面视图之开启、呈现、关闭(切换至其他视图)，请见图2-1及其说明。如果当时也处于文档结构视图中(带出了结构图)，那么页面视图就出现于右视窗，否则就出现于全视窗。

页面视图最棒的特点是，忠实呈现所有页面元素(扉页、题献、序言、目录、前言、正文、页眉、页码、附录、参考文献、书目、索引)于制版和印刷时的实际表现。想当然尔这会耗用大量电脑运算。以往硬件设备不足，专家有时候会建议您选择性地使用其他视图来进行编辑。如今速度问题已经罕见¹¹，因此我建议完全在页面视图下进行所有工作。

文档结构图 + 页面视图 = 创作者的好帮手

创作虽是一种艺术，其过程却是个反复修改的大工程。即便是诗圣杜甫也要“为求一诗成，捻断三茎须”。诗求精简，所以字字推敲；长文档(书籍、论文)规模

¹¹ 以我个人经验为例，在Windows98+Word97+512MB RAM+Intel Centrino CPU(1200 MHz)+30GB HardDisk的环境下，使用页面视图操作高达60MB的电子文档，速度还是可以的。

庞大，创作过程中也需要不断调整架构，修改细节，往往返返，敲敲打打。

文档结构图可以让创作者对“大纲（标题）与正文”有最直接的链接。页面视图则可以让创作者对文档的成品模样有最直接的感受。这些都将反馈至创作过程，影响创作。换言之，文档结构图+页面视图（加上往后将介绍的 5.11 节大纲与样式的层次链接、5.10 节图表题注、5.12 节交叉引用等功能）使得 Word 不只是文档的排版工具而已，更成为一个良好的文档创作平台。书籍、论文、大型文书报告的作者，如果能够善用这样的助力，不单单自己可以轻松掌控成品面貌，更可以让创作过程更顺畅。我自己便是这样的实践者和受惠者。

这便引出了两个疑问：

Q1：创作者未曾采用如此“先进”的写作平台，会不会影响日后排版？

A1：不会。旧石器时代的情况是，供稿者以 plain text（无任何格式化的文字）完成全部文稿，那么排版的所有责任落在排版者身上（见下一问答）。新石器时代的情况是，供稿者懂一点点排版，也许会设置某些（非正规/非内建的¹²）样式，也许漫无章法地对某些文字做一些文书处理（例如对某种特定身份的文字使用特定字体和特定大小）。这种漫无章法的行为不但无益，反而可能对日后的排版作业造成伤害。这些设置日后被排版者的样式覆盖上去，多半统统不见了，所以先前的工作无益，但字符的某些设置，即使在正式排版作业中被样式覆盖，仍有可能保留下（视 Word 版本不同而异），这就使得排版者的批量作业受到妨碍，必须在许多地方一一微调。这是惟一可能因为“一开始的胡乱设置”而“妨碍排版效率”的地方。



在将近 12 年的写译历程中，近来有些书籍我采取合译模式：合译者先将整本书籍初译/粗译给我，再由我接手。有些合译者会在初译稿中加上某些文字格式处理，给我带来如上所说的麻烦。因此后来我总要求他们交 plain text。当然，合译者先对初译稿做出一点点视觉效果，对其工作的进行也有帮助，这一点我能理解。

Q2：排版者是否可以不理会本节所说的“正规作业”，仍能做出好成品？

¹² 所谓正规（内建）样式，是指“Word 内部已经对它们设定了很好的整组互动机能”的样式，例如**标题 1~标题 9**、**目录 1~目录 9**、**索引 1~索引 9**、**正文**……，详见 5.1.3 节。

A2: 一个不懂样式的排版者，不可能做大型文档（动辄两三百页）的排版工作！如果排版者会使用样式，但所学不精，没能取正规样式名称，而且也不知道这些所谓“正规（内建）样式”的功效，其实只需花三两天时间理解 Word 正规排版，很容易就可以把自己从游击队提升为正规军。先前以 dirty work 完成的产品也可以轻松修改过来。



我曾经是我自己口中那种学艺不精——非指品质，而是过程——的排版者。我不知道如何让 Word 为我自动化产生目录、产生索引，我不知道如何调整页眉上的章节名称、转换编页数字系统、产生图表编号和各种交叉引用。所以我纯手工完成目录和索引、硬撬硬掰页眉上的章节名称、把书籍拆成文前、正文两个文件分别给予 i, ii, iii … 罗马编页和 1, 2, 3 … 阿拉伯编页，再拆出一个文件做双栏索引。是的，所有事情当然都可以绕个弯解决（work around），但苦工做了不少。

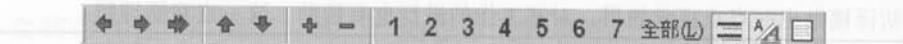
不过，写得出这本书的我，如今当然已在华山之巅 ◎

2.3.4. 大纲视图

关于大纲试图之开启、呈现、关闭（切换至其他视图），请见图 2-1 及其说明。如果当时也处于文档结构视图中（带出了结构图），那么大纲视图就出现于右视窗，否则就出现于全视窗。

图 2-3 左为文档结构图，右为大纲视图视窗。点击左框的标题便可使右视窗之光标（键盘输入点）移至对应的正文位置（但大纲视图下的正文往往只是一些简单条列）。右视窗亦可设为页面视图（图 2-1）或 Web 视图（图 2-5）。

大纲视图的特点是，可呈现长文档的大纲，并方便进行大纲编修（别忘了，结构图虽也呈现大纲/标题，但不允许编修）。然而大纲视图的这种方便性未必就比得上页面视图+快捷键（5.1.4 节），因为在大纲视图中藉由点击以下工具栏：



而决定标题层级，未见得比“按快捷键以决定使用哪一层标题”来得方便。

大纲视图会自动显现一个大纲工具栏如上。您在图 2-3 右视窗键入文字，便可点击大纲工具栏所提供的按钮，控制该文字成为哪一层标题，或令它成为正文。请注意上图的大纲工具栏右起第三个按钮被按下了，意思是“每一段只显示首行”。右起第二个按钮也被按下了，意思是“将格式显示出来”，也就是将样式上的设置（诸如字体、大小、字距、行距等等）显示出来。于是形成了图 2-3 所示画面。

大纲工具栏的细部说明如图 2-4 所示。

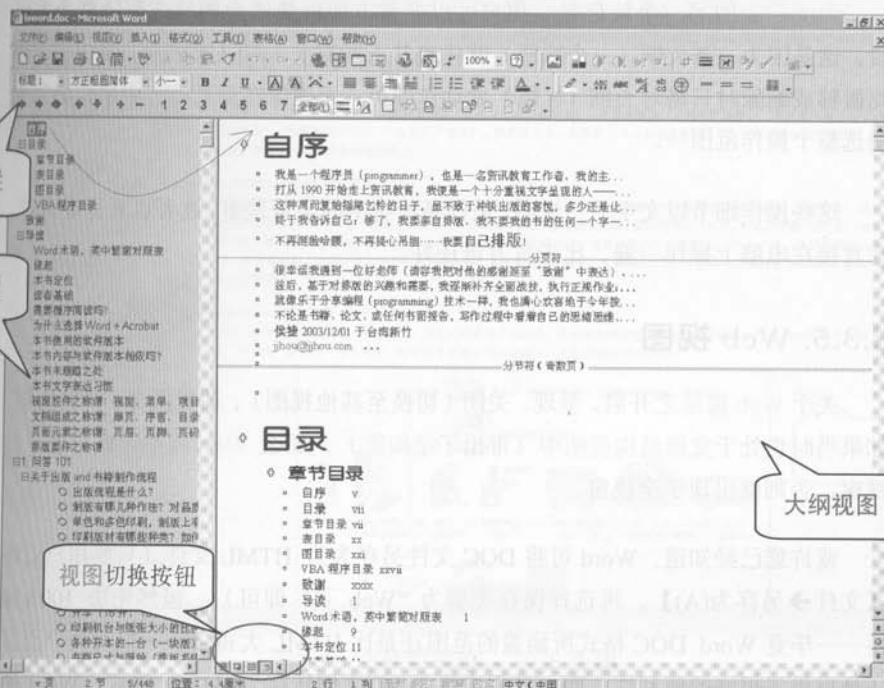


图2-3 文件引导框 + 大纲视图



图2-4 Word97 大纲工具栏，细部说明

大纲视图的另一个特色是，用户可以非常方便而快速地搬移或删除整个大区段，这是因为一条大纲（一个标题）就代表其下所有次大纲（次标题）及内容。因此搬移或删除时只需对大纲（标题）做动作即可，无须像在页面视图中那样必须先圈选整个操作范围¹³。

这些操作细节以文字一一说明，是件事倍功半的辛苦差事，读者也看得很辛苦。您直接在电脑上操作一遍，比千言万语还好。

2.3.5. Web 视图

关于 Web 视图之开启、呈现、关闭（切换至其他视图），请见图 2-1 及其说明。如果当时也处于文档结构视图中（带出了结构图），那么 Web 视图式就出现于右视窗，否则就出现于全视窗。

或许您已经知道，Word 可将 DOC 文件另存为 HTML 文件（只要用户点击【文件→另存为(A)】，再选择保存类型为“Web 页”即可）。虽然无法 100% 转换——毕竟 Word DOC 格式所涵盖的范围还是比 HTML 大得多——但对用户而言还是提供了很好的便利性。而一旦用户将 DOC 文档转存为 HTML 格式，Word 便自动转换至 Web 视图。此视图主要用以模拟浏览器画面，使得开发者身历其境。所以 Web 视图不提供标尺、不分页（也就没有页眉、页脚、页码）、文档内容随着视窗大小的调整而自动转折；其他特征和页面视图差不多。

由于不分页，也就节省了许多运算，Word 的反应速度会快得多。但一如前述，以目前的电脑设备和一般大型文档的体积而言，图 2-1 的页面视图并不会比这里所说的 Web 视图慢到哪里去。

¹³ 如果操作范围很大，超过一页，圈选动作有可能带来一点小麻烦——尤其当您的电脑+屏幕的速度非常快时◎

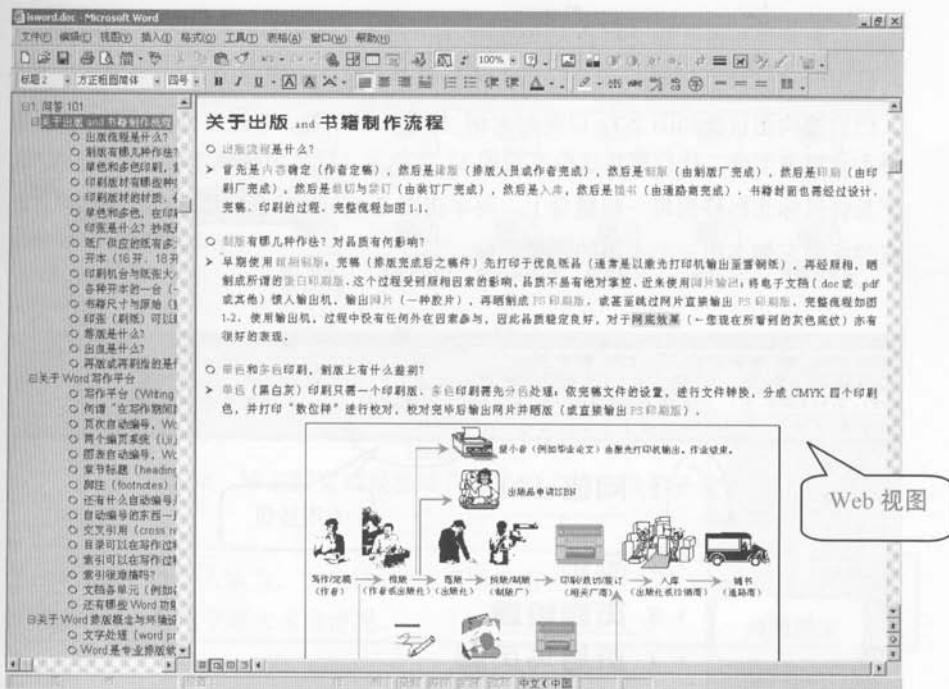


图2-5 文件引导框 + Web 视图

2.3.6. 主控文档视图

关于主控文档视图之开启、呈现、关闭（切换至其他视图），请见图 2-1 及其说明。

主控文档视图在 Word 2000 中有着远比 Word 97 方便的操作性。所以这个主题我特别以 Word 2000 示范；其 *Online-Help* 对于主控文档视图有良好而完善的说明，这里只言简意赅地给您一个大局观。

图 2-6 是 Word 2000 的大纲工具栏，其中 **主控文档视图** 按钮（含）左侧与 Word 97 的布局相同，右侧则是比 Word 97 大纲工具栏多出来的 7 个按钮。

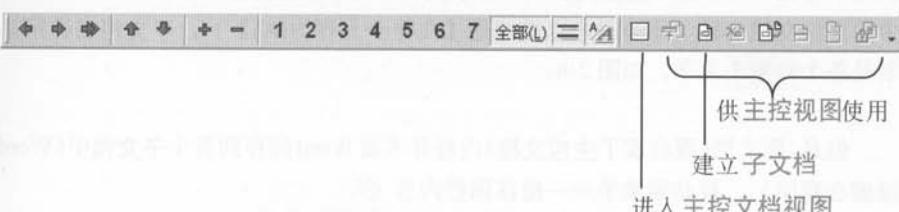


图2-6 Word 2000 的大纲工具栏。右侧 7 个按钮给主控视图使用

首先我将一份既有文档载入 Word2000，点击【视图→大纲(O)】再点击大纲工具栏中的**主控文档视图**，或直接点击【视图→主控文档视图(D)】。接下来将这份文档的结构图折叠如图 2-7，以便对大纲/标题一目了然。右视窗也将因此对应折叠得十分清爽干净。我打算将这份文档的 11 个单元（样式为**标题 1**者）做成子文档，于是将鼠标光标移到每一标题身上，再单击图 2-6 的**建立子文档**。此时图 2-7 右视窗的标题左侧多出一个小小的**文档图标**。

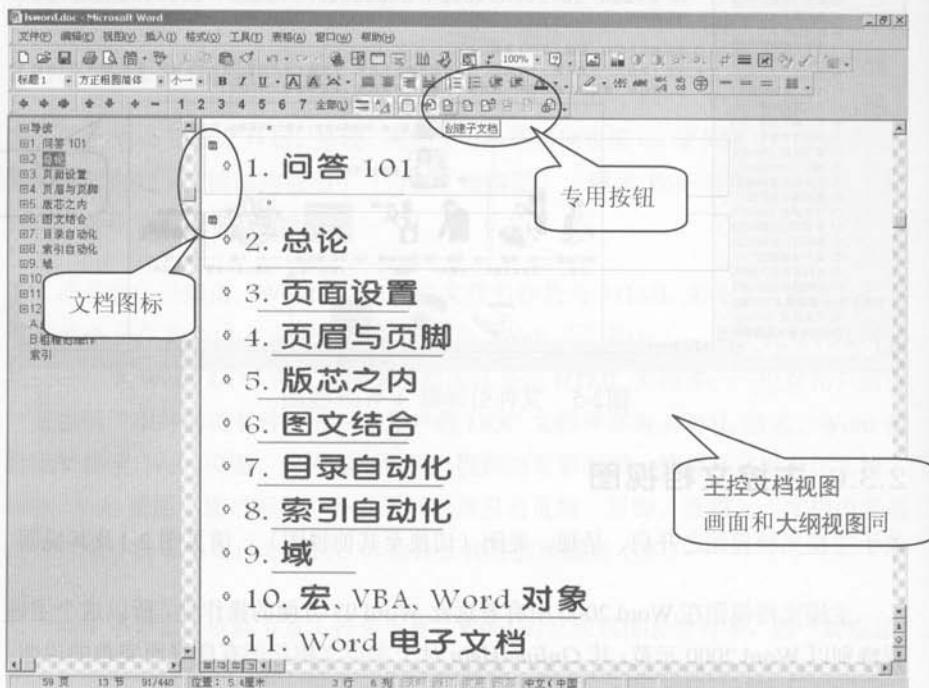


图2-7 文档结构图 + 主控文档视图 (in Word 2000)

逐一完成所有**建立子文档**动作后，点击【文件→保存(S)】或【文件→另存为(A)】，于是 Word 储存主控文档和所有子文档。其中主控文档的名称沿用目前操作中的这份文档名称（除非刚才点击的是【文件→另存为(A)】），子文档名称则分别是各个**标题 1**文字，如图 2-8。

但是，原文档(现在成了主控文档)内容并未被 Word 储存到各个子文档中(Word 没那么聪明)，我必须亲手一一搬移那些内容 ⑧。

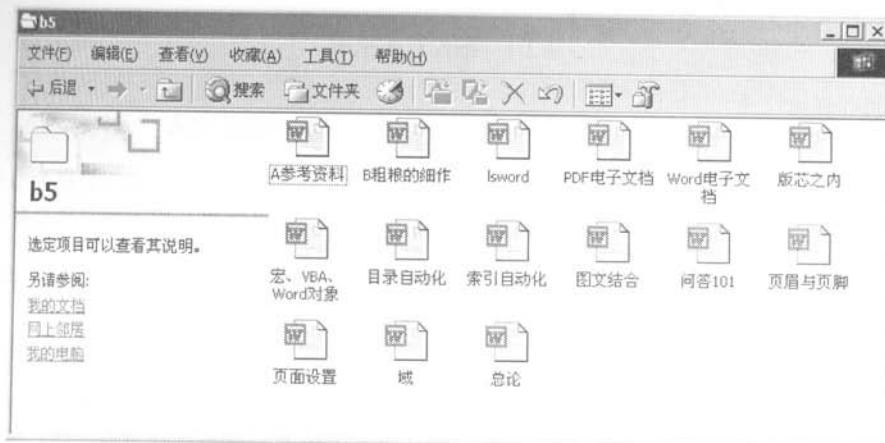


图2-8 被主控文档视图储存起来的主控文档和子文档



再一次强调，我个人认为，文档结构图+页面视图（图 2-1）是 Word 最佳写作平台和排版平台，适用于绝大多数情况。

2.3.7. 细说页面元素

构成大型文档页面上的所有东西，本书称之为**页面元素**。这些元素也就是排版过程中必须制造出来的东西。

下面表 2-1 是各种页面元素的称谓及概略说明，所有称谓都以 Word *Online Help* 为准，画面请参考图 0-3 和图 1-6。以下“说明”栏中的效果都可以在 Word 文档中做出来。

表2-1 细说页面元素

页面元素	本书 页次	说明
		注：以下各效果都可以在 Word 文档中制造出来
纸型	95	<p>如果文档只为少量传布，例如硕博毕业论文，或小规模的学术论文发表，以激光（或喷墨）打印机输出，再送到复印店复印、装订、加封面，大约也就可以了。这种情况下，台湾通常选择 A4 纸张（$210 \times 297\text{mm}$），欧美通常选择 letter-size 纸张（$216 \times 279\text{mm}$）。市面上各型打印机都接受这两种常见纸张。Word 之中的纸张设置非常简单：点击【文件→页面设置(U)】后再点击纸型。</p> <p>如果文档打算大量传布（例如书籍），就需经过制版手续，再以印刷机输出，并由装订厂装订。这种情况下需要精确知道书籍成品的尺寸。计算机书籍近年来的国际惯用尺寸是 $185 \times 230\text{mm}$，台湾也很习惯这个大小（俗称大 18 开）；大陆传统使用 $182 \times 257\text{mm}$，近年来也颇能接受上述国际尺寸。</p> <p>纸张大小最好先问清楚并设置好。万一设错了也来得及在任何时候修改，并不影响整个版面风格。</p>
版心	96	文档四周一定需要留白，满满的文字多难看！正文（含图片）出没区便是所谓的版心。版心是否包含页眉/页脚则是言人人殊。本书中我视版心为“不包括页眉/页脚”之正文区域。
页眉（繁体版）	98	在台湾，有人把页首和页尾合称为页眉。但繁体版 Word 并不使用页眉这一名称。
天头地脚	图 1-5	页眉以上区域至页缘，排版/印刷业称之为天头。页脚以下区域至页缘，排版/印刷业称之为地脚。
页眉 header	98	版心上方的一小块区域。通常用来放置页码和章节名称。常见的安排方式是左页页眉内侧放置章名，右页页眉内侧放置大标题（大节）名。左右页页眉之外侧均放置页码。也有人喜欢将页码安排在页脚。
页脚 footer	98	版心下方的一小块区域。有时其外侧（右页右侧和左页左侧）被用来放置页码。有些人（例如我）喜欢利用这个区域放置书名（字体宜小），让每一页所带信息更完整；想想看，如果有一天您发现您从图书馆复印来的书页非常重要，却不知道它们来自哪一本书，启不令人顿足！
页码 page number	109	方便统计页数和查找的编号；大型文档一定要有页码。书籍统统使用 1,2,3,4…流水编号，这是国际习惯。“x-x 页”（第 x 章第 x 页）的编页方式通常只在书籍以外的场合使用，但是台湾电脑出版界却特立独行，非常令我感到迷惑！

页面元素	本书 说明 页次 注：以下各效果都可以在 Word 文档中制造出来
	书籍通常包含两种编页形式：正文（第一章起）页码使用阿拉伯数字 1,2,3...，文前（版权页、扉页、序言、目录、前言、致谢）页码使用罗马数字 i,ii,iii...。版权页、扉页通常不显示页码，甚至不编页码。各章首页通常也不显示页码，但我并不欣赏这一点——同样是为了信息的完整性。
节 14 section	把一整份文档区分为二（或三或四或五）的一种逻辑切割，可以影响“原本会被连续运用、逐页运用”的页面元素，例如页眉设置、页脚设置、页码编号、脚注编号。
此非一般 所谓“章节” 的那个“节”	可以想见，若要让一本书（置于同一份文档内）同时存在罗马数字和阿拉伯数字两种编页系统，只要将文前和正文切割为两个不同的节，就可以办到。如果要让各章脚注的编号都从 1 开始，只要将各章切割为不同的节，也就可以办到。
文前 1-11	通常是指版权页+扉页+序言+目录+前言+致谢。版权页和扉页要不要计算页码？不一定，都可以。
正文	第一章起的所有内容（但也有人称附录、索引为文后）
文后	有人称附录+索引为文后。通常延续正文的流水页码。
扉页 titlepage	封面的下一页，有书名、作者名的那一页。版式设计上并无殊异于一般的“章”，若需特殊字体只要个别调整即可。
题献	“献给我亲爱的某某”之类的话所在的那一页。版式设计上并无殊异于一般的“章”，若需特殊字体只要个别调整即可。
版权页	上有内容提要、版权声明、出版公司名称、国际标准书号、字数统计、印次记录、印量记录等等的那一页。可能被放在扉页之后，也可能被放在书籍最后。版式设计上并无殊异于一般的“章”，若需特殊字体只要个别调整即可（因内容甚少，似乎无需杀鸡用牛刀地为它设计专属样式）。
序言、 前言等等	版式设计上并无殊异于一般的“章”，若需特殊字体只要个别调整即可。
目录 table of content	第 7 章 目录的制作从来不是问题：文档完成后，将所有章节名称抄出来，再抄出对应的页码，便可整理为目录。但这种工作可归类为 dirty work！运用特定的样式名称，Word 可为我们自动生成目录。而且可以随时生成、一再生成，以顺应创作过程中一再的变化。

14 此节非彼“节”（章节的“节”）

页面元素	本书说明
页次	注：以下各效果都可以在 Word 文档中制造出来
标题 heading	5.1.3 节 也就是章名和大中小节名称。只要您针对这些名称使用 Word 内建的 标题 1~标题 9 样式，Word 就可以自动化抽取它们的文字和所在页码，做成目录。也可以指定 Word 抽取 标题 1~标题 9 以外的样式做成目录。
章,chapter	由章名、大中小节、段落、图、表……构成。见各元素之介绍。
正文/ 段落 paragraph	文档的主体内容。您可以根据不同的正文属性（例如一般性文字段落、程序代码、特别说明等等），为它们设置不同的呈现风貌（亦即为它们设置不同的样式）。
脚注 footnote	5.3 节 正文中某些不适合直接置放（以免影响阅读流畅）的额外说明。因位置的不同而有页脚和章末两种，前者称为脚注，后者称为尾注（Endnote）。科技文档通常采用脚注（页脚），文艺类书通常采用尾注（章末）。它们的编号可以分章归零，也可以一路流水到底（和前述的节的设置有关）。 15 —这就是一个脚注。  这就是一个尾注。
图文框	5.9 节 可摆放在任意位置（甚至版心之外）的文字框。 ← 本栏左边就有一个图文框。 
图片 figure	5.8 节 图片内容可以是各种格式（bmp, jpg, gif,...）。有多种手法可以插入文档内，对文档大小的影响各异。可透过【插入→题注 (C)】由 Word 自动化设置、排列、更新图片编号。图片位置可有多种选择。本栏左侧就有一张图片。
表格 table	5.7 节 您可以先画表格再填内容，也可以先写好文字内容再转换为表格。后者一点都不麻烦。可透过【插入→题注 (C)】由 Word 自动化设置、排列、更新表格编号。
题注 caption	5.10 节 版心内的页面元素如图片、表格，其编号可由 Word 自动维护，包括赋予初值、更新重排（以反映页面元素的添加或删除），称为题注。例如“图片 3-2”或“表格 2-1”等等。自动化题注对创作过程有非常大的帮助。另一个对创作过程带来极大帮助的是交叉引用，见表 2-2。
标签 label	5.10 节 “图片 3-2”或“表格 2-1”中的“图片”或“表格”，称为标签。Word 允许用户自定标签，例如“小秘诀 1-1”。

15 您现在所见的就是一个（页脚）脚注。其编号为 '15'，以**脚注引用**呈现，其内容（本处文字）以**脚注文字**呈现。

参考文献 reference, bibliography	可视为一章，版式上并无特别之处。若需特殊字体或格式，个别调整（或做出一个样式来用）即可。
附录 appendix	可视为一章，版式上并无特别之处。
索引 index	第 8 章 索引的制作似乎是 Word 的罩门。“无法制作索引”也成为“Word 不是专业排版软件”的证据。然而误解皆因不了解，Word 完全可以自动化生成索引，而且相当方便。当然，没有任何人或任何软件有心灵感应，能够知道您想做出什么样的索引。您必须下命令给 Word，它才能为您服务。告诉 Word 将 “xxx” 抽取为索引是一件非常简单的事。如果您想为 10,000 个语汇做索引，您就得下达 10,000 个命令（还有其他办法吗？）。幸运的是这 10,000 个命令可以整批下达，好极了！

2.3.8. 细说排版要件

上一节的 Word 页面元素，几乎都有对应且同名的 Word 排版要件。另有不可见的排版要件，列举于下。所有称谓都以 Word *Online Help* 为准。

表2-2 细说排版要件

排版要件	本书说明	注：以下各效果都可以在 Word 文档中制造出来
页面设置 page setup	95 纸型、页边距（关系到版心、页眉、页脚的位置）、奇偶页配置，合称为页面设置。作用范围可设在一个节之内。	
分隔符号 breaks	图 3-8 有分段 (lines)、分页 (pages)、分栏 (columns)、分节 (sections) 四种符号。分节符号有多种选择，见图 3-8。	
格式 format	115 这本是一个平常的称谓。Word 用它来表示字体、段落、制表位、边框、语言、图文框、编号等七种设置。	
样式 style	5.1 节 Word 允许用户为一整群文字设置格式（见上）。样式是 Word 排版的基础，也是其灵魂。少了样式不成其为排版。视作用对象之不同，有两种不同的样式：用以设置整个段落者，称为段落样式，用以设置段落内某一群文字者，称为字符样式。	
快捷键 hot key	5.1.4 久久才使用一次的样式，其施行步骤即使慢条斯理亦无妨。常需用到的样式则最好搭配快捷键，才能发挥疾如风火的排版速度。毕竟鼠标的单键神功，永远比不上键盘上的十指齐发。	

排版要件	本书说明
页次	注：以下各效果都可以在 Word 文档中制造出来
模板 template	5.2 又称为样式库，顾名思义就是“样式的仓库”，是把设置好的一群样式以后缀名.dot 储存起来。一旦如此，当 Word 用户新建文件时，Word 会列出所有模板（样式库）供选择。不过，只有【文件→新建(N)】才会提供这样的选择机会，如果点击的是工具栏上的 图标（代表“新建文件”），不会有此一问，将采用 Word 内建的 normal.dot（其内容可能被修改）。
域 field	第 9 章一种像编程语言（programming language）中的变量（variables, fields）之类的东西。事实上英文版 Word 所用字眼就是“field”。Word 的许多自动化功能少不了它。
索引项 index entry	8.1.1 一种域（见上），名为 XE ，意指 inde X Entry。【插入→索引和目录(D)】再加上一些后续动作，便可将整份文档中所有 XE 域收集妥当，安排出一份漂亮的索引。再也不需要 dirty works 了！
交叉引用 cross reference	5.12 文档创作过程中，常为了内容的反复变动，导致如“请参考 2.3 节”或“请参考第 23 页”或“请参考图 5-3”之类的引用无法固定下来。这个困扰可交给交叉引用解决，对创作过程有极大帮助。引用的对象可以是文档中的（几乎）任何东西。另一个对创作过程带来极大帮助的是题注，见表 2-1。

2.3.9. 心得偶拾

下面是我多年来的计算机书籍排版心得。许多看法见仁见智，纯粹提供给您参考。

一致性有多重要？

一致性当然很重要，使得整份文档的标题有一致的长相、段落有一致的长相、表格有一致的长相、页眉页脚有一致的长相、版面布局有一致的长相。

但是一致性并非是“必须死守不变”的东西。在肉眼不易察觉，或可察觉但无碍观瞻的情况下，为了版面安排的方便，可以便宜行事。

我的看法是，区域一致性重于全体一致性。某一页的段距和行距比其他页稍微挤一点点，读者根本察觉不出来，就算察觉出来了也不会在意。只要您是有原因地做这类调整，例如为了挤进一张图或一个表格，以求该页布局更加美好，just do it!



多年以来，我翻译英文书籍的原则是：一页对译。一方面为了让内容与原书保持几近相同的位置（页次），方便不同语种版本的读者在网络上讨论或比对（这在计算机领域是常有的事），一方面为了保留原书索引（我的翻译理念是保留相当程度的专业术语，因此英文索引原物照用，没有问题）。

要能够做到一页对译，排版必须完全掌握在自己手里，才不会在事后的沟通往返上折损生命。如果某一页的中译结果比英文原页篇幅少一点点，没关系，留白可以作为将来笔记眉批之用，读者也很欢迎。万一中译结果比英文原页篇幅多一点点，那就比较麻烦。遇此情况，我最常用的手法是将该页每一段落的行距缩小 0.5 pt。不够的话，段距上下各少 1pt。还不够的话，手段再霹雳一些，将缩小点数再加大一些，直到英文版一页内容真能够塞进中文版一页为止。

0.5pt? 1pt? 2pts? 这……这么少有用吗？唔，一页是由多行多段构成的，集腋成裘，或许就可以让原本挤出页外的东西再塞回来 ☺

有时候我会为中译本加上一些译注，这就可能较大幅地增加中译本篇幅，不再是上述 1~2 个 pts 微调能够解决。此时再怎么东挪西移也不够，多出来的只好被挤到下一页去。下一页东榨一点西榨一点，下下一页也如法泡制，顶多两三页也就完全追补回来。这可能会造成原书索引的一小部分失真，只好睁一只眼闭一只眼了。

现在我来对于局部一致性做个实验。您现在所看到的这整个框框，其行距其实比其他页少 0.5pt，段落前后距各比其他页少 1pt。看得出来吗？您不会是鹰眼吧☺。这样一来我便空出了 $0.5 \times 20(\text{行}) + 1 \times 5(\text{段}) \times 2(\text{上下}) = 20\text{pts}$ 空间，本页便因此额外多挤进了一行文字。



文档 (document) 与文件 (files)

一整份文档放进一个 .doc 文件好呢？还是分章放进个别的 .doc 文件比较好？

如果单一文件过于庞大，当然会拖累许多操作的反应速度。但是随着硬件的长足进步和高档设备的普及，这种影响愈来愈不成为重点。至于巨大单一文档的储存更不成为问题，因为储存媒介的容量成长较诸运算速度的成长，幅度更是惊人。



因此，虽然 Word 提供了主控文档视图（图 2-7），可使用一个中控

文档（文件）控制其他数个子文档（文件），我还是认为，50MB 以下的文档“一个文件搞定”最方便，但是记得把您的 RAM 加多一些。



随着对 Word 的理解和自我功力的进步，我的书籍由过去的一章一档，变成一书三档，变成现在的一书一档。一书三档是因为，当时我不知道如何在同一份文档（文件）中切换罗马页码和阿拉伯页码，而索引的双栏格式又常常造成页码错乱。因此过去我总是把一本书分为文前、正文、索引三份文档（三个文件）。

页码最好采用流水号

恕我孤陋寡闻，只在台湾看过“x-x 页”（“第 x 章第 x 页”）这样的书籍编页形式。拿到这样一本书，我无法立即知道总页数，而我也不可能拿起笔来为它们加总！吃饱了没事干呗？

使用流水号码，一是为了符合国际惯例，二是为了馆藏方便（总页数是书籍的基本资料），三是为了给读者方便，四是为了它本身没有任何不方便——阅读上没有，制作上也没有。经过正确的设置后（4.3 节），页码由 Word 维护，又何患牵一发动全局呢！动了全局又如何？不到半秒钟就搞定了！



尽量让页面信息丰富

很多书籍不喜欢在最前面数页（扉页、题献等等）显示页码，也不喜欢在每一章的第 1 页显示页码（英文书尤其如此）。对于前者，基于美观的角度，可以理解。对于后者，我总觉得无此必要，徒然降低页面信息量而已。如果读者拿这一页去复印，或如果这一页脱胶掉落（杞人忧天？），便会因为缺少页码而带来不便。每章首页少个页码，美观上没有独到之处，大多数人不过东施效颦罢了。

善用页脚的人不太多。我喜欢在页脚加上书籍名称。理由同前：为了提高页面信息量。在空间不大的页脚上添加书名，会不会太拥挤？唔，调整字体大小是个办法，而且美观问题总是见仁见智。

快捷键的取舍



右手持鼠标者，通常只能以左手按快捷键（偶而才会放下鼠标，以右手辅助）。好按的键就那么几个，例如以 **ALT** 或 **CTRL** 搭配左手附近的按键如 **Q, W, E, R, T**。

A, S, D, F, Z, X, C, V 等等（当然啦，按键习惯因人而异）。

显然，如果要把“按下双键组合快捷键”的责任完全交由左手承担，使右手完全不必离开鼠标，您最好谨慎规划，把为数不多的几个好按的快捷键用于最频繁被使用的样式。

注意2-1 中英字体夹杂混用



我喜欢对我的计算机类书做各种术语上的字体变化。这个贴心的处理对读者（包括我——我也是我的书籍的经常读者）的阅读带来很大的帮助。在我的书籍中，一般英文字使用的字体像这样：normal，程序代码像这样：main()，类名称像这样：CWnd，特殊术语像这样：default construct 或 partial specialization，设计模式（design pattern）像这样：**Singleton**，重构（refactoring）准则像这样：Extract Method，重构坏味道（bad smell）像这样：Duplicated Code，书籍名称像这样：《Design Patterns》，操作符像这样：assignment operator，强调语气则采用**黑体**。面对这些散布全书各处的各种字体种类，最好是为它们各自设计字符样式（5.1.2 节），并制定快捷键（5.1.4 节），如此才方便随时看到就随时设置。

没有一位排版者能够对这么多种用途各异的字词做字体设置，除非作者本人就是排版者！

为了方便排版，我也对正文、各层标题、章名、程序代码都设了快捷键。对我而言快捷键的需求量十分庞大。

快捷键用惯了就不要改动，否则就像手上有 10 个密码，总是想不起来哪个账户用哪个密码一样。快捷键一定要事前先规划好。

样式命名考虑

当您为样式加上快捷键，除非日日使用，熟练得不得了，否则三个月后排版下一份文档时，可能又忘了哪个样式使用哪些快捷键。

为此，我后来学了乖，为样式命名时便把快捷键一并记上去作为名称的一部分，像是**!正文 alt-c**、**!大标题 alt-1**、**!中标题 alt-2**、**!程序代码 alt-L** 等等。

用户自定样式和 Word 内建样式（如**标题 1**、**正文**、**索引 1**、**目录 1**、**脚注文本**……），在**样式**对话框的**样式**列表中将按次序（英文字母顺序和中文音

序次序)混合排列。为了寻找方便,我建议不妨在所有自定样式前加上特定符号(例如 '!')。由于样式还分字符样式和段落样式两种,所以也不妨为所有字符样式名称加上前缀字(例如“段落内”或“字符”或某符号),便可与所有的自定段落样式区分开来。

不要有洁癖

洁很好,癖不好。您看“癖”不是病字边吗!

所有的样式设置(或其他Word设置),毕竟都是为了完成工作。这些设置的名称或内容若有某种程度的一致风格(就像上面所说那样),当然很好。闲暇之余把这些家伙整理整理也是好事,但不能走火入魔。智者千虑尚有一失,何况许多样式的设置乃是“且战且走”,没有人能够事前规划所有样式,并放之永恒皆准!

做与不做的分寸拿捏,属于艺术范畴!

2.4. 版面美学之我见

许多读者称赞我的书籍“版面清爽”,也有人说我的书籍版面透着科技书的“素雅”,字体和字体大小的安排有“科技的紧凑感”。感谢读者谬赞,为此我鼓起勇气大言不惭地发表我的“版面美学”。

2.4.1. 字号、字距、行距

自从亲自排版后,没有一本我的书的正文字体超过9.5磅¹⁶大小。小字代表是《深入浅出MFC》和《C++Primer中文版》(繁体),它们都是1000页的大部头,我用了9磅字,才没让它们成为超级大砖块。最近三四年写译的书籍都比较薄,我的眼睛又开始迈向老花,所以给它松半磅,改用9.5磅字。程序代码则一律使用8.5磅Courier New字体。

市面上许多中文电脑书籍采用的字体是12磅。这就怪不得朋友说我的书“有科技的紧凑感”。我的字体虽然比一般小两磅,但从没有人反映说看起来很难受。我想这是因为字的大小和字距行距有着协调的比例。

¹⁶ 惟一例外是《Thinking in Java中文版》。破天荒地正文使用12磅字,程序代码使用11磅字。这是为了与原书一页对译。原书排版文件就是采用那种大小的字。很多读者来信表示不习惯。我初始也觉得丑,但由于老花眼渐渐有了一些,也就还挺能接受这种大字报。

或许基于某种工程师性格，为了让页面信息多一些，我喜欢页面风格紧凑一些。字小不是问题，只要字距和行距配合得宜。字大不见得阅读起来就舒服——如果行距不足以搭配的话。

虽然页面风格比较紧凑，但我又是很重视“留白”的一个人。阅读技术类书难免需要空白页面写点心得笔记眉批。再就视觉感受而言，文字塞得满满整页恐怕读起来也不会舒服。作为读者的我喜欢有适当留白的书，作为作者的我也因此为自己的书做了适度的留白。

2.4.2. 字体、色彩不要过于丰富

大型文档的排版，就读者角度考虑，应尽可能以阅读舒适为要。毕竟长篇累牍下来眼睛的负担不轻。根据研究，宋体（也就是您现在看到的这种字体）是中文各字体中阅读起来最轻松的一种，这正是为什么宋体被如此广泛使用于书籍、报纸、刊物正文的缘故。如果通篇以**隶书体**或黑体或仿宋体或**行楷**或楷书或魏碑或瘦金或颜体为正文，美则美矣，眼睛受不了！

排版者容易犯的兵家大忌就是：花俏。“花俏”在中文词汇里头并不是个褒义词，通常伴随着杂乱、用力过度、为赋新词强说愁。一个缺乏克制力或美学眼光或人文素养的人，会把一切可用资源（字体、色彩、各种花样）倒到每一张页面上。

多半时候，读者其实要的是一种素雅。别说科技类文档极适合这个基调，即使其他书种，干净素雅也永远是一个不败的选择。不要因为“我手上有 20 种字体”或“本书使用全彩”，便把所有字体、所有色彩倾倒满地。有一阵子台湾流行全彩电脑软硬件操作学习书，绝大多数给我的感觉便是“俗艳”。

就我个人，中文字体我一般只使用宋（作为正文）、**粗圆体**（作为标题）、**黑体**（作为亮点提示）¹⁷。英文字体种类可以多一些（请参考 p.87），因为英文字体种类极多，较易找出彼此协调者。字体的搭配需要做实验，不妨选择文档内页面元素最丰富的部分，试排三两页，看看您所选择的这些字体是否协调。

协调与平衡，永远是美学的第一课。

¹⁷ 本书倒是多用了**隶书**作为样式名称。



我的一位好朋友译了一本 C++ programming 领域的扛鼎之作。原文书的分量、作者的地位、市场的期待和反应，都是不必说了。出版之后我获得一本赠书。

在这本地位超拔的书籍（的中译本）上头，翻开扉页，竟有花花草草的点缀。印刷品质和复印机输出相差无两。第三页“译者序”三个字是**粗圆体+斜+粗底纹**，序文内则有品质如夹报小广告的边框手法。页眉内侧有一方块+粗底纹+阴影，上以“海报体”标示出版社大名；并且由于它出现在页眉，所以每一页都有！目测正文字体大小为 12 号。页码采“x-x”分章编法（而非流水号码）。至今我不知其总页数多少，只量得高度为 6 公分整。

朋友对这译本的样貌，不满意的程度似乎到了“不好意思谈及”。刚好当时我也写了几篇文章，鼓励每一位专业写译人员都应该亲自排版。

数年之后朋友又译了另一部经典。我又获得了一本赠书。这次的样貌极为不俗，是他自己排的版。我写信问他，对于我向来鼓励的“专业写译人员应该自己排版”有什么看法，他回复道：I can't agree with you more. ☺

2.4.3. 斜体中文字的杀伤力

英文字原就有斜体的传统。蟹行文字写得快一点，不都带点倾斜吗⑤。所以斜体英文字（例如表 2-3）在我们的视觉中并不突兀。

中文字号称方块字，从来没有斜体的传统。斜体中文字在传统观念和视觉印象中非常突兀（例如表 2-4）。

千万别使用斜体中文字！阿弥陀佛！

表2-3 斜体英文范例

字体呈现	字体名称	特殊处理
<i>How these words looks like?</i>	Times New Roman	加斜
<i>How these words looks like?</i>	Book Antiqua	加斜
<i>How these words looks like?</i>	Lucida Sans	加斜
<i>How these words looks like?</i>	Arial	加斜
<i>How these words looks like?</i>	Comic Sans MS	加斜
<i>How these words looks like?</i>	Cataneo BT	无

表2-4 斜体中文范例

字体呈现	字体名称	特殊处理
这些字看起来如何？	华康简宋	加斜
这些字看起来如何？	华文琥珀	加斜
这些字看起来如何？	华康简黑	加斜
这些字看起来如何？	华康简楷	加斜
这些字看起来如何？	华康简仿宋	加斜
这些字看起来如何？	华文行楷	加斜
这些字看起来如何？	华文新魏	加斜
这些字看起来如何？	华文中宋	加斜
这些字看起来如何？	方正粗圆简体	加斜
这些字看起来如何？	华康简综艺	加斜
这些字看起来如何？	方正隶书简体	加斜
这些字看起来如何？	方正粗倩简体	加斜



如果要在文句之中对某些话加强语气或希望获得突出效果，可使用比较粗壮的字体。在一堆宋体中对某些字做**加粗处理（bold）**，屏幕上看起来效果不错，输出则不够明显。我喜欢在一堆宋体中以黑体形成强调。这么做的输出效果极好，但屏幕上即使放大至 150%（我最常用的工作模式）也不很明显，除非放大到 200%。Well，很少人有那么大的电脑屏幕允许做 200% 放大的全页排版工作吧②。但无论如何，还是要视文档的呈现方式（屏幕或输出）来决定采用什么做法让您的文字抢眼起来。

2.4.4. 中英文字体的调和搭配

计算机书籍的中英夹杂情况十分频繁（我所写译的书尤其频繁），所以中英文字体的搭配极为重要。搭配得宜与否，是见仁见智的问题。例如先前 p.87 的搭配我就觉得可以接受，但您不一定看法相同（不过我可是有雄厚“民意基础”的③）。

总之，要做实验！至少先说服自己，才能端出来。字体搭配得丑不是您的错，自己也认为丑却还端出来吓人，那就是您的不是了④

2.4.5. 图片的大小

图片当然是大型文档中抒缓视觉压力的头号帮手。不少人却把它拿来挤、拿来塞，不当地运用它。这种情况在书中不常见，但在“篇幅固定、作业期限紧迫”的杂志刊物上头，为配合版面而把图片塞进“剩余空间”的情况，却是俯拾皆是。

另一种情况是，在同一份文档采用分辨率差异极大的图片。有的极精美，有的极粗糙（连锯齿都跑出来了）。冰火两极之间不知该如何评量其水准。这个问题必然出自图片的来源，但总要设法解决。我真的看过发行量数万的杂志不时出现这种情况。

图片有主角的身价，可别只把它当成填满空间的花瓶。有的图被缩小到了上面的字迹已经模糊不清，这种处理方式完全不可取！有的图分辨率低到读者眼睛瞪得铜铃大还看不清上面的文字，这种图根本不该用！

对于那种“图中文字是有用信息，希望让读者读到”的图（电脑书的图多半如此），最好的做法是让图片内的文字大小差不多等于正文字体大小，或小一点点，而且全书一以贯之。对于不含文字（或文字并非重点）的图片，惟一的要求是清晰，并且有着与上下文情境相匹配的尺寸。

留白是一种艺术！宁让图片有一致的分辨率与版面风格，并因此多用一两页，留下一些空白！也不要整个页面塞得像沙丁鱼罐头，既牺牲信息，也牺牲美感。

2.4.6. 不要每一页都出血

出血，奇怪恐怖的名词。请见 p.28。

血流遍地不是很可怕吗！如果每一页（或许多页）都出血，从书的侧面看，您会看到一大堆灰灰黑黑的东西，看起来又脏又旧。这又是页面元素太“丰富”的俗艳结果。

将出血运用在目录和索引上，引导读者从书侧快速翻查到他想要的大致位置，那是很好的。但别每一页都出血嘛。目前台湾电脑书业大概有许多排版人员突然都学会了页眉的学问，迫不及待要 show 一下，把页眉做得花俏至极，“血流满地”。

^a 本处就是一个尾注，出自 p.82。其编号 'a' 以尾注引用呈现，其内容（此处的文字）则以尾注文本呈现。

1. 篇幅大小

篇幅大小的美学，是版面设计中一个永恒的话题。篇幅太小，文字密密麻麻，视觉疲劳，阅读困难；篇幅过大，又显得空洞，缺乏信息量，无法满足读者的需求。因此，如何在两者之间找到一个平衡点，就显得尤为重要了。

篇幅大小的美学，首先取决于文章的内容和性质。对于一些专业性强、信息量大的文章，如学术论文、报告等，篇幅可以适当大一些，以便于读者深入研究。而对于一些通俗易懂、信息量较小的文章，如新闻报道、广告宣传等，则可以适当小一些，以便于读者快速阅读。

3. 页面设置.....	95
3.1. 纸张大小.....	95
3.2. 版心的大小和位置.....	96
3.3. 页眉与页脚的大小和位置.....	98
3.4. 节 (sections) 的概念.....	99
3.4.1. 单元永远始于奇数页.....	99
3.4.2. 多重页码系统.....	101
3.4.3. 奇偶页的页眉/页脚不同调	102
3.4.4. 自动编列行号.....	102
3.5. 出血.....	103

3

页面设置

Word 所谓的页面设置，是纸张大小、边界、页眉与页脚（之位置与内容）的总称。其作用范围是在一个节中（当然您也可以选择套用于整份文档）。换言之，一份文档可以有许多个节，每个节的页面设置都可以不同。正因为如此，我们才能：

- 让各单元总是始于奇数页
- 让页码混用罗马数字和阿拉伯数字（但同一节内必须型式相同）
- 让奇偶页有不同的页眉和页脚内容
- 让脚注于每一章重新编号

3.1. 纸张大小

通常，如果您的文档将来要拿到输出中心输出网片，并送至印刷厂印刷，最好是将纸张大小设为将来成品（例如书籍）的大小。如果您的文档并非那么正式或大量，只打算以打印机输出（再配合复印），那么应该依据打印机所用的纸张来进行设置。

图 3-1 是【文件→页面设置(U)】所引发的**页面设置**对话框。不论进入其中哪一个**页签**，右下角的**应用于(A):**都有三个选项：

- 本节
- 插入点之后
- 整篇文档

作为一份“不很作怪”的文档，我们可以将**页面设置**中的所有设置都选择**应用于(A): "整篇文档"**。

目前流行的电脑书尺寸是 185×230mm，于是我做了图 3-1 的**纸型**设置。

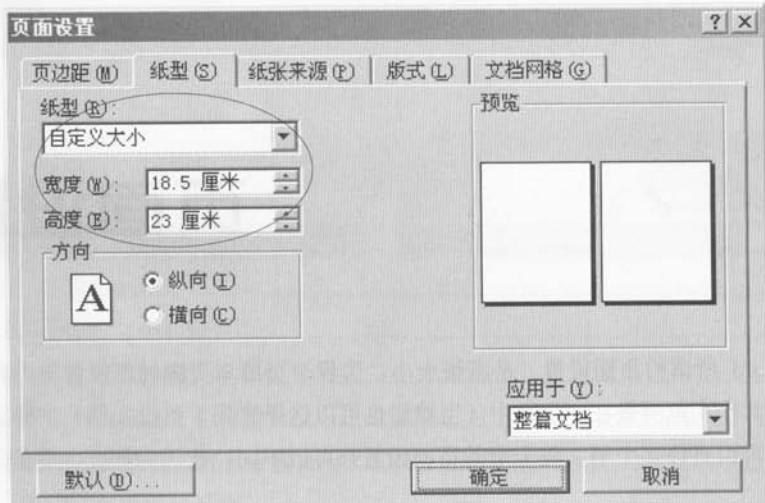


图3-1 纸型（纸张大小）的设置

3.2. 版心的大小和位置

所谓版心，是指正文所在的纸面范围。上有页眉和天头，下有页脚和地脚，左、右则有边距（留白）。下面是【文件→页面设置(U)】的【页边距】，展示本书的版心设置：

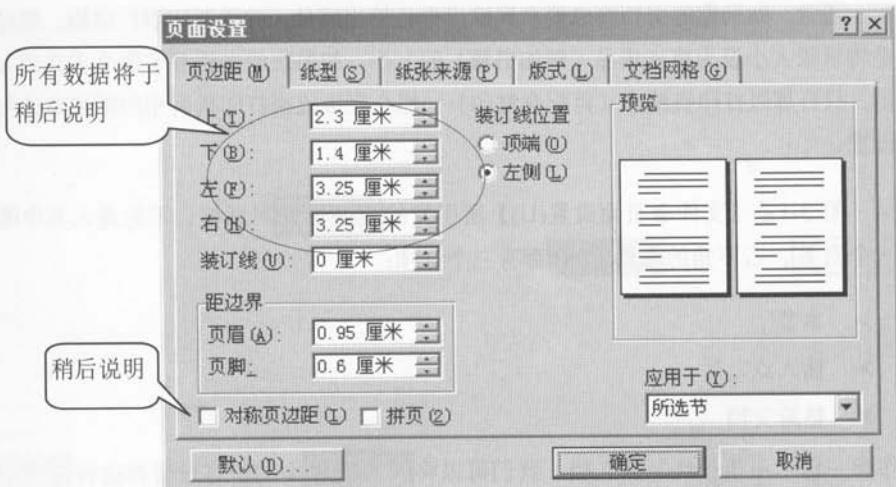


图3-2 页边距的设置（Word2002 有更多功能，见 p.41）

图中数字的意义如下：

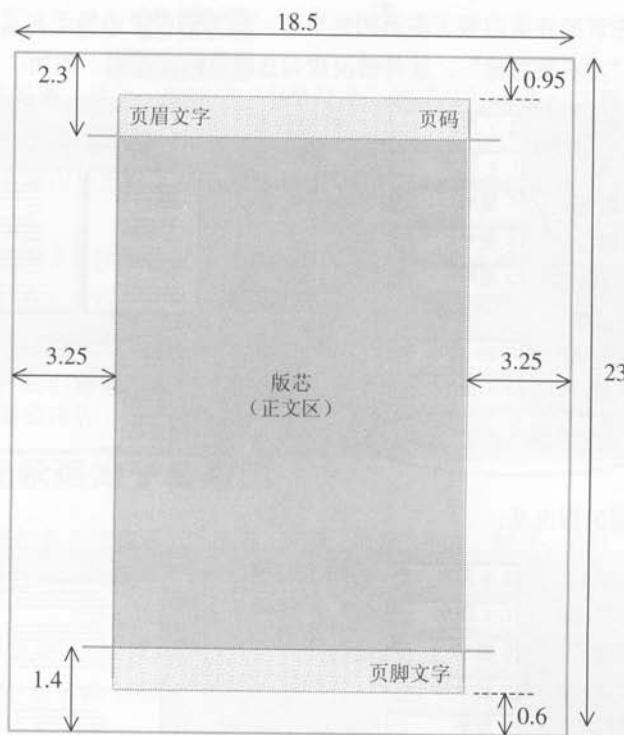
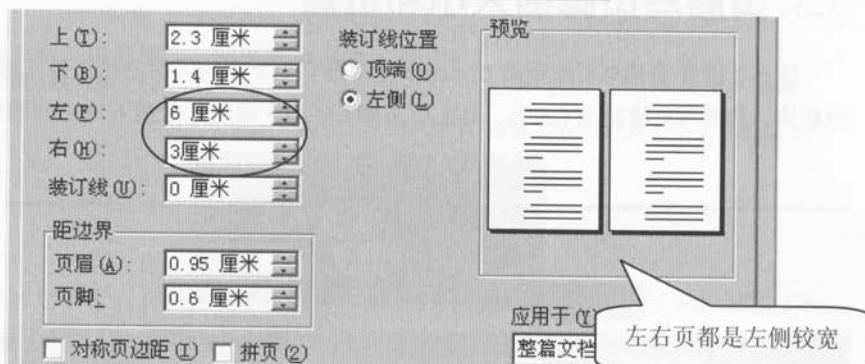


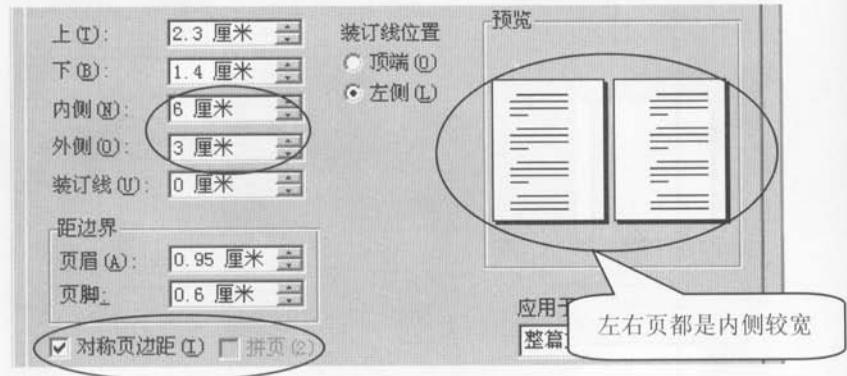
图3-3 页边距设置的数据意义

图 3-2 左下角有两个选项 对称页边距 (1) 拼页 (2) :

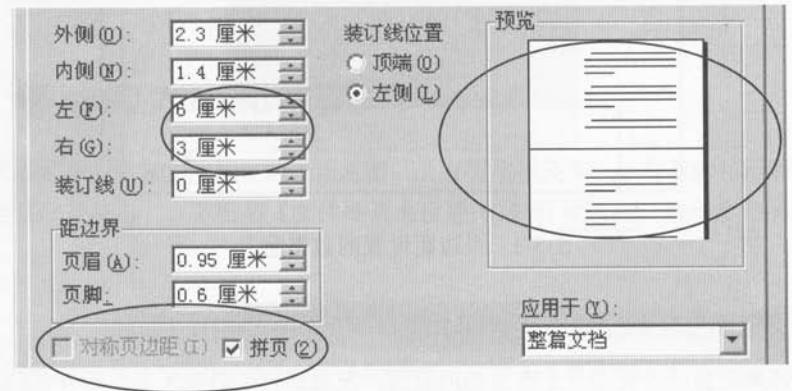
- 如果左页的外页边等于右页的内页边，左页的内页边等于右页的外页边，这就是“左右页相同”。这种情况不应勾选对称页边距，例如：



- 如果左页的外页边等于右页的外页边，左页的内页边等于右页的内页边，这就是“左右页对称”。这种情况应勾选对称页边距。例如：

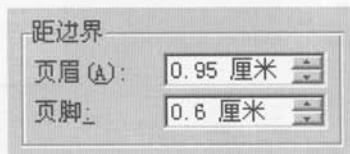


- 拼页的情况是：



3.3. 页眉与页脚的大小和位置

图 3-2 设置 页眉与页缘距离 0.95cm，指的是 页眉文字的上缘与纸张最上缘之间的距离。页脚与页缘距离 0.6cm，则是指 页脚文字的下缘与纸张最下缘之间的距离。



设置好这些数据后，页眉和页脚就确定范围了，见图 3-3。将来一旦双击页面的页眉或页脚区，便可看到图 3-4 和图 3-5 所示的画面（此刻请注意图中虚线即可）。页眉和页脚的内容设置将于第 4 章介绍）。

3.4. 节 (sections) 的概念

Word 所称的节，并非一般所谓“书籍章节”的“节”（见 p.17 脚注）。由于节不是一种可视的页面元素，而且一般人对于书籍/大型文档并不存在这种逻辑概念，所以很容易被用户忽略。然而如果少了节的参与，许多排版效果无法达到。

先前提到的图 3-2 页面设置，套用范围都可以是“以文档为单位”或“以节为单位”。但一如先前所说，将相同的页面设置套用于整份文档，可做出一个“不太作怪”的成品，而且适用于 99.9% 的场合（多少人会喜欢太过复杂而眼花缭乱的版式呢）。因此，我们对节的观察重点，放在它对页眉和页脚的影响上。

3.4.1. 单元永远始于奇数页

通常，书籍的各个组成单元（序言、目录、前言、各章、附录、书目、索引……）都由奇数页开始。为达此目的，我们应该为它们各设一个专属的节，步骤如下：

- 以鼠标双击页眉区，出现如图 3-4 所示的画面。左上角显示此单元之节编号（下图显示此前已设有两个节），右上角显示“目前此节的页眉设置与前一节相同”或否：

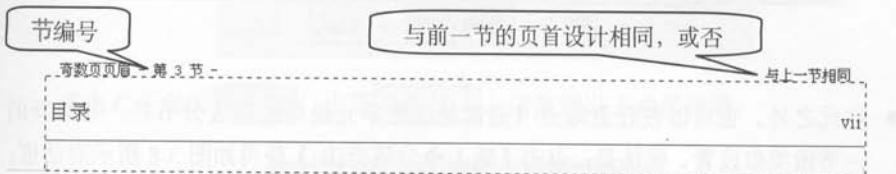


图3-4 页眉实例画面

- 以鼠标双击页脚区，得到类似画面，如图 3-5：

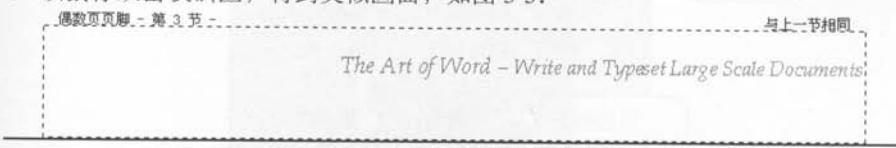


图3-5 页脚实例画面

- 此时您已进入页眉区或页脚区，屏幕上自动出现**页眉和页脚**工具栏如图 3-6：

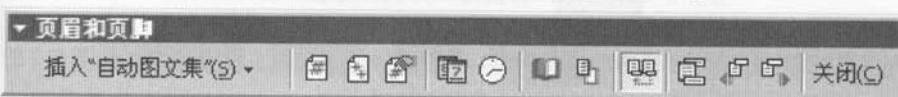


图3-6 页眉和页脚工具栏

点击右数第 5 个按钮（上图凹陷的那个）。那是个状态切换开关（按一下开，再按一下关，反复切换）。按下后页眉与页脚右上角的“与上一节相同”字样便告消失。这便切断了与前一节的相似关系。

- 现在，点击【文件→页面设置(U)】的版式，并设置如图 3-7：

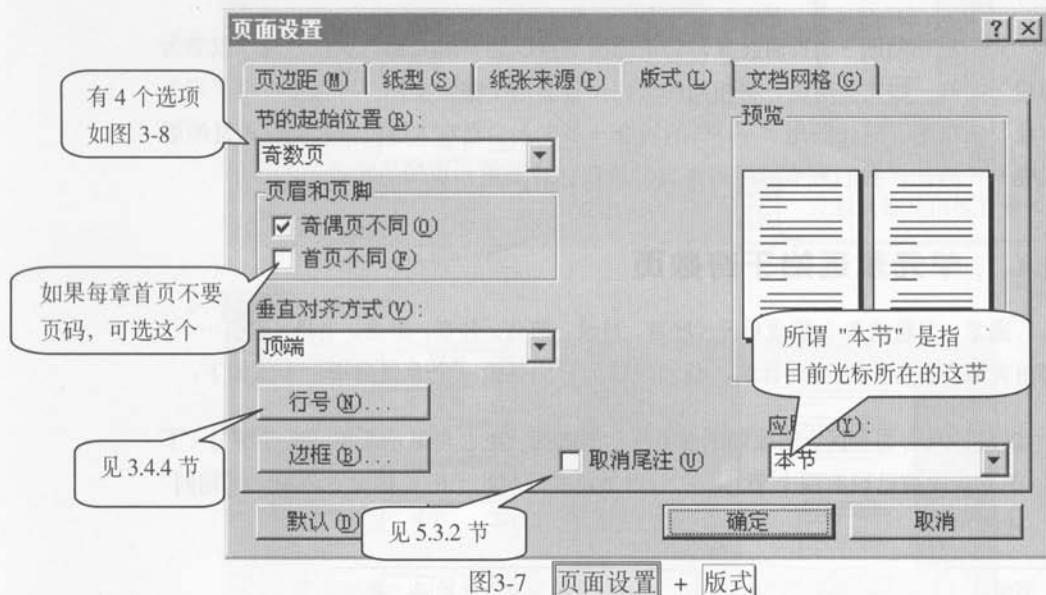


图3-7

- 除此之外，也可以在任意地点（通常是在各单元最后）插入分节符，并对新的一节做类似设置。做法是：点击【插入→分隔符(B)】获得如图 3-8 所示对话框：

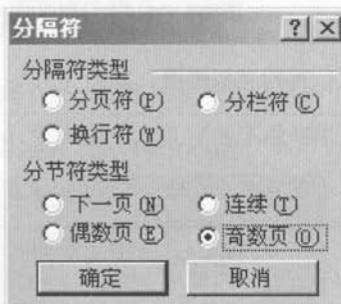


图3-8 [分隔符]: 有换行、分页、分栏、分节四种

其中分节符栏目内的下一页、连续、偶数页、奇数页，和图 3-7 的节的起始位置(R)栏目所提供的选项完全相同。

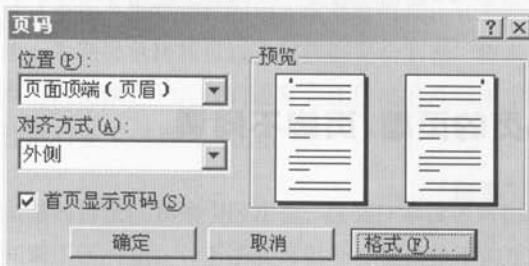


若对节做直观修改动作，常会出现糟糕的意外。例如想将一个分节符 奇数页 改为一个分节符 偶数页，直观想法是删除旧符号、添加新符号。但这么做会影响更前一节的属性。解决办法是：将光标移至该节，然后点击【文件→页面设置】 + 版式(L)，再在其中的节的起始位置(R)选择偶数页。

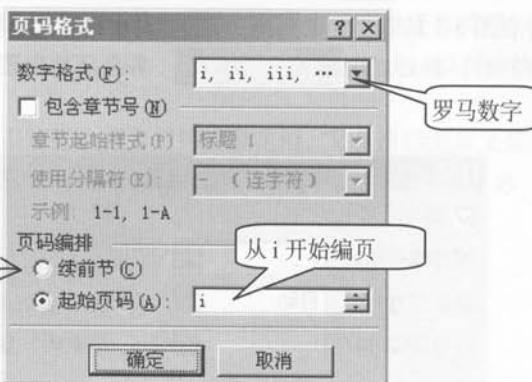
3.4.2. 多重页码系统

习惯上，书籍第一章起以阿拉伯数字 1,2,3... 编页，第 1 章前（统称文前）以罗马数字 i,ii,iii... 编页。过去这是一个非常困扰我的排版问题。

如今有了节的概念，此问题拨云见日。首先在起始编页单元（例如扉页）的页眉区内圈选页码（如果此时尚未有页码，见 4.3 节），然后点击【插入→页码 (U)】：



点击右下角的格式(F)，出现页码格式。请按照以下画面设置：



此后各节（例如目录、前言、致谢）都令其页码格式与上同，但令其页码编排为：

◎续前节。

接下来，进入第一章的页眉区，圈选页码（如果此时尚未有页码，见 4.3 节），并模仿上述步骤，将数字格式(F)和页码编排两个栏目修改如下：



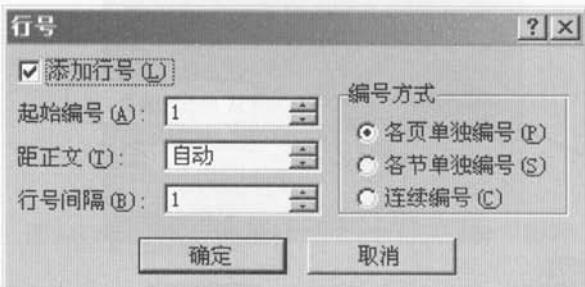
此后各章（各有专属的节），都令其**页码格式**与上同，但令其**页码编排**为：
 续前节。
 起始页码(A): 1

3.4.3. 奇偶页的页眉/页脚不同调

请见 4.2 节“页眉”。

3.4.4. 自动编列行号

图 3-7 的页面设置**版式**页中，我们看到**行号(N)...**。如果您当时圈选了某个文字范围，并在图 3-7 的右半侧的**应用于(A):**栏目中选择“选取文字”（意取“目前被圈选的文字”），然后点击**编入行号(N)...**，并在随后出现的对话框中勾选**编入行号(L):**



Word 便会将您所圈选的那段文字独立出一个节（因此必从新页开始），并在版心外侧加上由 Word 自动编列的行号（行号的形貌由**行号**样式来控制）：

```
1 Dim rge As Range          '存放从 Selection取出的 Range
2
3 For nLineNum = 1 To Selection.Paragraphs.Count    '对每个段落
4     ...
5     Set rge = Selection.Paragraphs(nLineNum).Range  '取出个段落
6     rge.Text = sLineNum + rge.Text                   '加上行号
7 Next nLineNum
```

由于我有“为程序代码加上行号”的强烈需求，因此这个功能曾经很吸引我的注意。不过它从新页（新节）开始，这便远离了我的希望。最终我还是写了个宏来解决这个问题，详见 10.4 节。

3.5. 出血

不论是封面设计（不在本书讨论范围内）或书籍内页，只要您要求色块紧贴上下左右任何一个页边，为确保色块和页边之间能够完全紧密，将来裁切时不至于出现白色边（纸张本身的内馅颜色），我们通常会让色块的范围漫出页边（反正将来要裁掉），这种动作的行业术语称为“出血”。

我相当喜欢让书籍的目录区和索引区出血，这么一来读者可以轻易从一本书的侧面就看到目录区和索引区所在，一下子就可以翻开查找他要的章节。这和“字典侧边针对 A,B,C,D...区域标出色块”实乃异曲同工，在很厚的书籍中有其必要（薄书就不必了吧②）。举个实例，我翻译的《*C++ Primer 中文版*》（繁体版）厚达 1237 页，其目录又不像一般紧临于序言之后（如果那样倒也还好找到），而是在我所添加的数十页“导读”之后。为此，我帮它加上页边色块，以利翻找。

物极必反！有不少书籍喜欢把页眉搞得很花俏，又是造形图案又是底纹，而且血流满地，整本书从侧边看黑糊糊的一片，有一种“肮脏”的感觉 ③

4. 页眉与页脚	105
4.1. 页眉/页脚的大小和位置	105
4.2. 页眉/页脚的内容设置	105
4.2.1. 奇数页页眉内侧的标题名.....	106
4.2.2. 偶数页页眉内侧的章名.....	107
4.2.3. 页脚右侧的书名.....	108
4.2.4. 页眉/页脚可放置交叉引用吗?	109
4.3. 插入页码.....	109
4.3.1. 同一份文档的不同页码格式.....	110
4.3.2. 页眉文字下的横线.....	110

4

页眉与页脚

4.1. 页眉/页脚的大小和位置

关于页眉和页脚的大小和位置，请见 3.3 节，在该处与页面设置一起讨论。

4.2. 页眉/页脚的内容设置

看看您手上这本书，奇偶页的页眉和页脚有何不同？某些单元（例如序言、致谢、导读、附录、索引）的奇偶页页眉和页脚内容完全相同（只页码不同），某些单元（例如各章）就不是这样。以本书各章为例，奇数页眉的内侧放置大标题（**标题 2**），偶数页眉的内侧放置章名（**标题 1**）：

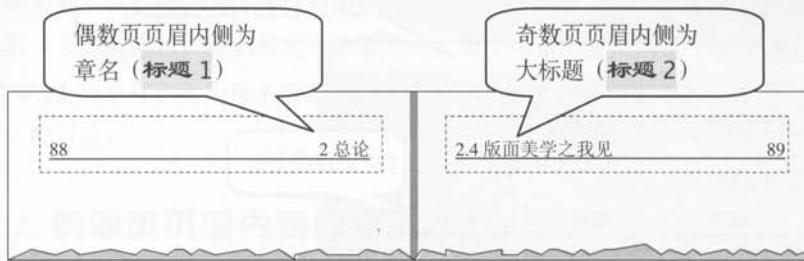
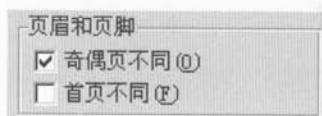


图4-1 奇偶页页眉不同的情况

事实上，奇偶页的页眉和页脚可以放置任何您想要的文字（甚至图片）。至于各单元的设置之所以能够不同，乃是因为我们为各单元设了各自的节（3.4 节）。

奇偶页页眉或页脚不同，主要是用于像书籍那样“可以摊开来，有左右页之分”的文档。另有一些文档，例如学校作业或报告，只是单面打印，通常也就不为奇偶页的页眉做差别设计。因此在做书时您有义务告诉 Word 该怎么做：请在图 3-7 中勾选“奇偶页不同”：



4.2.1. 奇数页页眉内侧的标题名

页眉区内的任何文字，会出现于同一节内的每一个页眉。然而我们通常希望页眉上的文字能够随页面内容而变¹，不要锁死。要达到这个目的，可以在页眉插入域（第9章）。

首先进入某一节的奇数页页眉（双击页眉区即可进入，双击正文区即可退出），令输入点停在您想要的任何页眉区位置（例如内侧），然后点击【插入→域(F)】，并选择域名(N)中的StyleRef（如图4-2）：

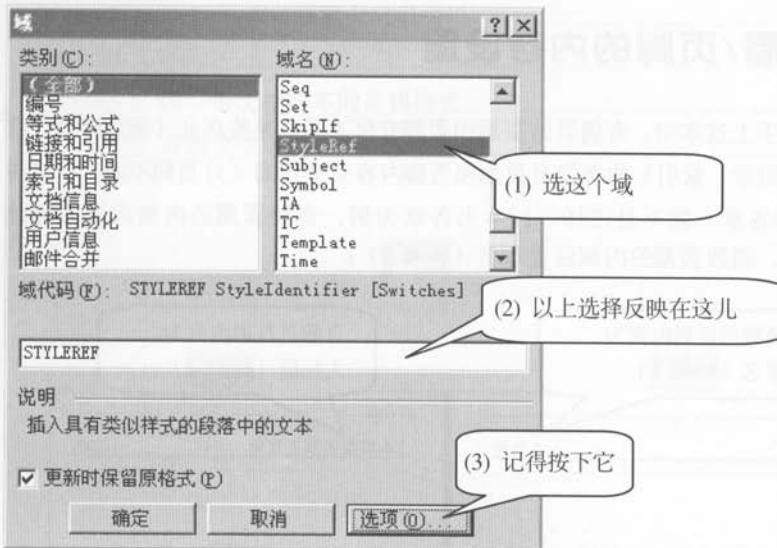


图4-2 为页眉插入一个域

接下来，点击右下角的【选项(O)...】，获得图4-3，并选择【样式】，再于其中选择

¹ 除非书名或作者姓名，才有可能在一本书中永远不变。但即使如此，workshop之类的书籍或论文，作者姓名也可能随页次而变。（workshop原意为专题研讨会，此处指三人合着、每人讨论一或两个主题的书。）

把作者姓名当做页眉或页脚，接近自恋行为，易招议论。但我真的看过有出版公司将公司名称放在页眉永垂不朽，嗯……勇气可嘉！（如果是制作简报，概念又不一样了：简报的设计和用途主要以“张”为单位，因此单张资讯完整性便显得更重要）

您希望置于页眉的样式。本书将各章第一级标题设为**标题 2**样式，因此我应该选择**标题 2**。得到的域是 **STYLEREF "标题 2"**。

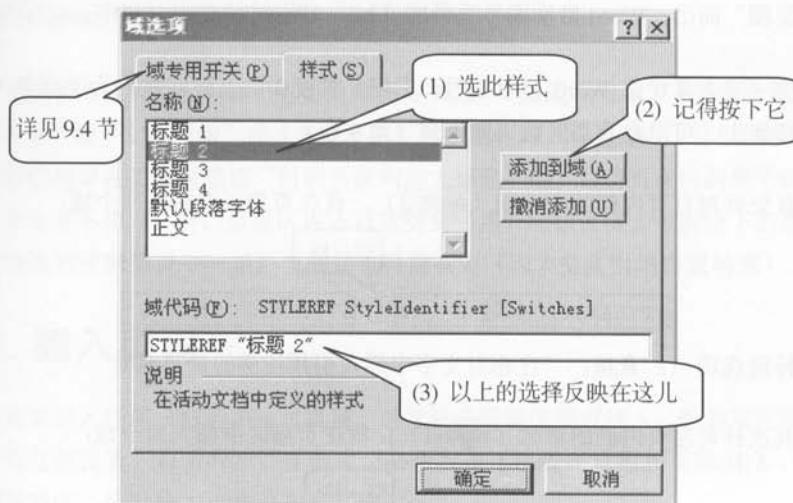


图4-3 选择域的选项

这么一来，每当重新组页，Word 便会将出现的任何**标题 2**文字填入页眉区内的那个域中，显示于页眉。

如果某页没有**标题 2**，我们会希望沿用前一页的**标题 2**；但当每章一开始尚未出现第一个**标题 2**时，我们却又不希望如此，而是希望页眉上的这个域保持空白。这对 Word 真是有点太为难了啦⑧。我在 9.4 节谈到 StyleRef 选项时，对这个问题做了一些讨论。

4.2.2. 偶数页页眉内侧的章名

请将上一节操作完全相同地实施于偶数页页眉上，惟一的不同是将**标题 1**样式（此为本书章名样式）置于页眉内侧。得到的域是 **STYLEREF "标题 1"**。每当重新分页，Word 便将出现的任何**标题 1**文字放到页眉区内的域中。如果该页并没有**标题 1**，我们应该会希望沿用前一页的**标题 1**，好极了，这正是 Word 的默认行为。

不幸的是，当 Word 抽取标题作为页眉时（不论是本节的**标题 1**或上一节的**标题 2**），它就“只抽取标题文字”而已。但难道标题之中除了文字还有什么其他东西吗？有的，还可能有 Word 的标题自动编号。如果您的标题由您自己手工编

号，例如写为“2.2 个人出版思潮”，那么“2.2”也是文字的一部分，会被抽取置于页眉。但如果采用 5.11 节的多层次样式，您所需要的手工撰写部分将只是“个人出版思潮”而已，Word 自动编号而得的“2.2”却没被抽取到页眉中。

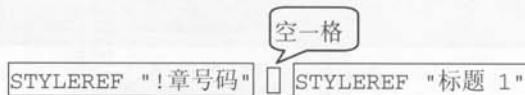
您一定会喜欢让 Word 自动编号，是吧！那就面临上述问题！为了改善 Word 的这项缺失，可以在页眉区插入两个域（第 9 章）。

首先针对右页内侧的大标题（**标题 2**），我在页眉区中插入两个域：



其中特殊选项 \r 意指：“在相对文字中插入引用段落的段落编号”。

其次针对左页内侧的章名（**标题 1**），我在页眉区中插入两个域：



其中**!章号码**是我自定的样式（见图 0-3）。得到的结果是（例如）位于偶数页页眉内侧的：



标题 1 和 **标题 2** 同样都承受了 Word 的自动编号，为什么我对两者却有不同的做法呢？这是因为对于章名，我并非乖乖地单纯接受 Word 自动编号，而是以另一个样式**!章号码**来辅助。原委请见 5.11 节。

4.2.3. 页脚右侧的书名

与先前所谈的页眉情况不同，页脚区内我放的是书名，而书名是不会变动的。因此请将上一节的操作同样实施于页脚，但不必再插入域，而是直接将文字写死；它会在每一个页脚区出现。页脚文字不妨直接进行文字处理，无需另设什么样式了。

4.2.4. 页眉/页脚可放置交叉引用吗?

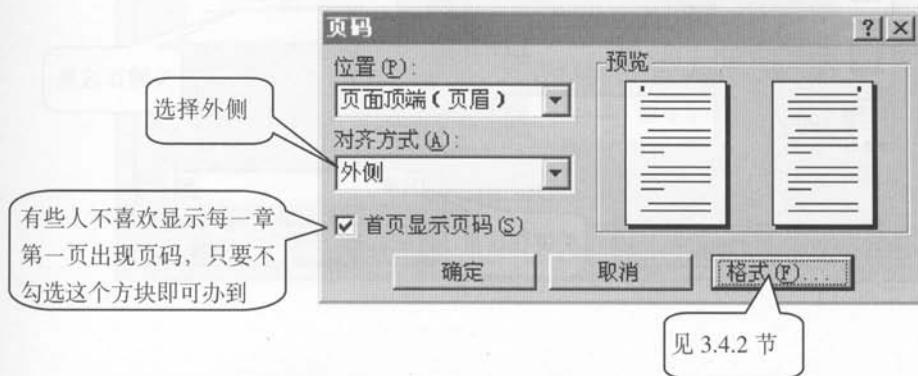
交叉引用（5.12 节）也是由域（第 9 章）构成，主要用来追踪某个页面元素的题注（Caption）变化或页码变化。

在文档中放置交叉引用，主要是希望将来万一“被追踪对象”（某个页面元素）的“编号”或“文字说明”或“所在页码”有所变化时，“追踪点”可以同步更新。这和我们稍早在页眉区追踪“目前页次内的大标题”或“目前页次内的章号码和章名”意义并不相同。所以页眉区内不该放交叉引用，应该放域，否则犯下的是一个概念错误和逻辑错误——虽然有时候也可能有效（因为交叉引用由域构成）。

4.3. 插入页码

首先进入任何一节的第一页页眉。通常双击页眉区即可进入，但如果此时页眉区没有任何设置，双击并不产生效果，必须点击【视图→页眉和页脚(H)】，才能看到页眉区，彼时输入点将停在最左端闪烁。

接下来点击【插入→页码(U)】。令【页码】中的对齐方式(A)为“外侧”，准备将页码置于外侧：



上图右下角的【格式(F)...】关系到页码的编号型式：罗马数字或阿拉伯数字？接续前一节编页？或由此开始新的编页？新编页从多少号码开始？这些问题及其解答，请参考 3.4.2 节“多重页码”。

页码的显示格式由【页码】样式负责控制。您可以修改【页码】样式以决定页码的字体、颜色等。

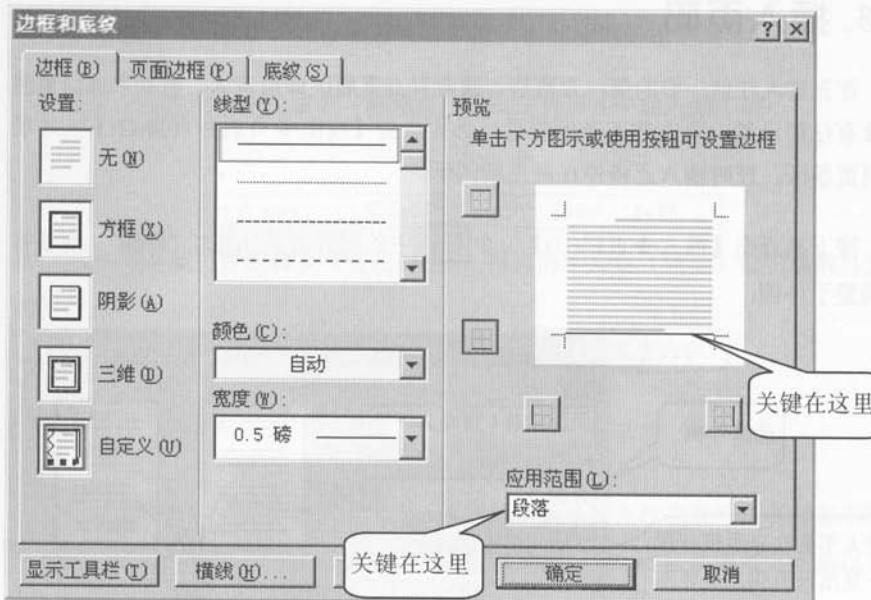
4.3.1. 同一份文档的不同页码格式

请参考 3.4.2 节“多重页码”。

4.3.2. 页眉文字下的横线

想要像本书一样在页眉下侧加一条横线吗？看起来和正文之间有比较好的视觉分隔效果。

没问题！进入页眉区后，点击【格式→边框和底纹(B)】，依照下图进行设置，即可得到您要的横线：



5. 版心之内.....	113
5.1. 样式 (styles)	113
5.1.1. 新增段落样式.....	127
5.1.2. 新增字符样式.....	128
5.1.3. 内建样式 (Built-in Styles)	128
5.1.4. 样式与快捷键.....	130
5.1.5. 将样式套用于既有文字.....	130
5.2. 模板 (Templates)	132
5.2.1. 各种设置的栖身规则.....	136
5.2.2. Word 内建的模板	138
5.2.3. 万范之本: Normal.dot	140
5.2.4. 全局模板 (Global Templates)	140
5.2.5. 模板 (Templates) 的管理.....	142
5.3. 脚注(Footnotes)	143
5.4. 多重栏位 (multi-columns)	146
5.5. 分隔设置.....	147
5.6. 项目 (清单; 列表; list)	149
5.7. 表格	150
5.8. 图片	157
5.9. 图文框.....	157
5.10. 题注 (Captions)	157
5.11. 多层次样式/自动化章节编号	162
5.12. 交叉引用(Cross References).....	172
5.13. 书签(Bookmarks)	176

5

版心之内

这里所谓的版心，是指正文活动区，不含页眉和页脚。版心区域范围的设置，请见 3.2 节。

版心内容要不是文字，就是图、表、脚注、分隔符号。其中可见的页面元素都应该以适当的样式加以驾驭和管理，而不要逐一进行低阶调整。本章的最大重点便是样式，那是一切排版动作的基础，然后才是其他诸元。

5.1. 样式 (styles)

无样式不成排版。样式是什么？让我们看看图 5-1 所示【格式→样式(S)】画面：

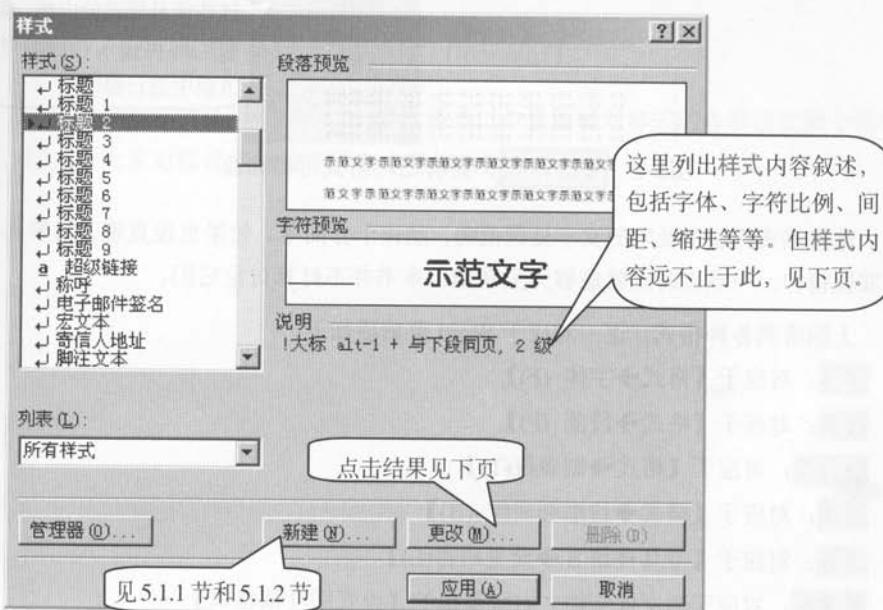


图5-1 样式对话框

修改既有样式

点击图 5-1 的【更改(M)】，出现如下画面。再点击最下方的【格式(O)▼】，拉出来的每一个选项都是样式所能蕴涵的各种格式：

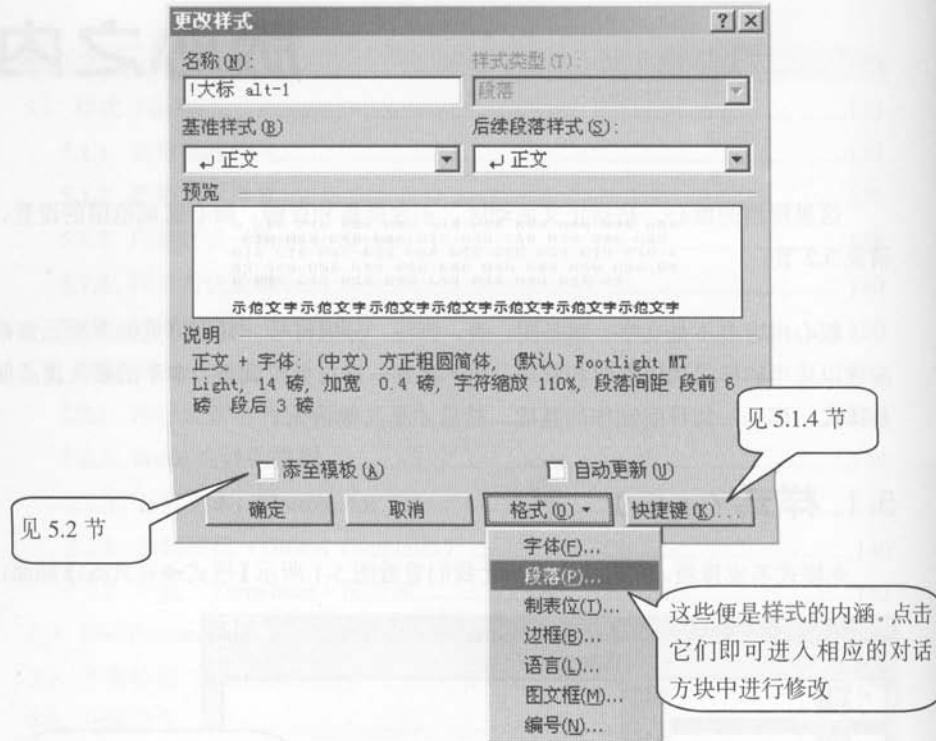


图5-2 【更改样式】。共有七种格式可供修改

这些格式都属于低阶的文字处理范畴，动作十分简单，效果也很直观，只要具备实验精神，一点点时间就能够完全掌握。本书并不打算讨论它们。

上图所列各种格式，逐一对应于 Word 菜单项如下：

- **字体**：对应于【**格式→字体 (F)**】
- **段落**：对应于【**格式→段落 (P)**】
- **制表位**：对应于【**格式→制表位(T)**】
- **边框**：对应于【**格式→边框和底纹 (B)**】
- **语言**：对应于【**工具→语言→设置语言(L)**】
- **图文框**：对应于图文框实物之右键菜单的【**设置图文框格式**】
- **编号**：对应于【**格式→项目符号和编号 (N)**】

低阶的文字处理是从点击上述菜单开始，高阶的排版作业则是把相关设置收纳起来成为样式，并在图 5-2 的对话框中加以设置或修改。对某段文字（键盘输入点所在的那一段文字）的直接格式修改（透过上述菜单项完成），影响的只是该段文字，但透过样式进行调整，其影响将遍及整份文档内所有套用此种样式的文字。

样式的存在，有以下功用：

- 使我们得以系统化地对整份文档的所有页面元素加以归类命名，例如 **章名**、**大标题**、**中标题**、**小标题**、**脚注**、**正文**、**程序代码**……事实上，Word 已经内建了用以代表各类型页面元素的各种样式（5.1.3 节），其格式设置虽然不可能满足用户的所有要求（这是当然），但它们各自带有特殊任务，您最好只是修改它们（的设置），尽可能不要自定新样式。
- 样式就是各种页面元素的形貌设置。
- 有了样式，日后打算调整全书某种页面元素的形貌时（例如将正文字体放大 2 pt、行高放大 1.5 pt 等等），只要修改对应的样式，便可影响整份文档，非常方便。
- 有了样式，Word 可以自动化制作目录并设置目录形貌、自动化制作页眉和页脚。有了样式，我们得以设置正文以外的页面元素如索引、脚注引用、脚注文本的形貌。

样式 (Style) 涵盖的各种格式 (Format)

本书虽不谈低阶文字处理，但我还是以少量篇幅对样式的各种格式做个简单介绍，让大家对样式的涵盖范围有个整体认识。

样式格式之一：字体

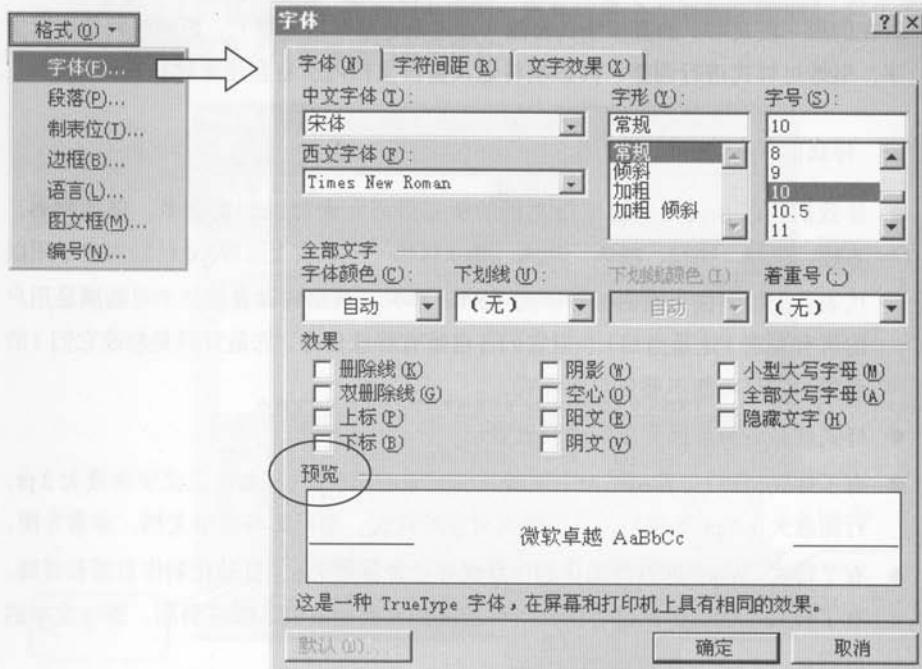


图5-3 【格式→样式(S)】 + 【更改(M)】 + 【字体(F)】 + 【字体(N)】

此图允许设置的字体相关属性皆已清楚列在画面上，实在无需多做解释。各种设置都可以在下方的**预览**框中看到效果。右上角的**字号(S)**:以 0.5pt. 为步进单位。

我认为还值得改善的一点是，Word 不允许我们针对同一样式，为其中文和英文做不同的属性设置。换句话说**字体(Y)**、**字号(S)**和**字体颜色(C)**总是同时适用（套用）于中英文身上。但中英文分开设置其实是有必要的，举个例子，如果我把**隶书 9 磅字**和常用的“Times New Roman”9 磅英文字摆在一起儿，看起来不是很平衡：

隶书 9 磅字 和 **Times New Roman 9 磅字**

如果中英文可以分开设置属性，我就可以把两者的差距拉到眼睛不易察觉的程度。

图 5-3 的另一个页签 (tag) **字符间距** 展示于图 5-4。通常我将**标题 n** 样式的字符大小比例放大为 110%，将间距加宽 0.4pt：

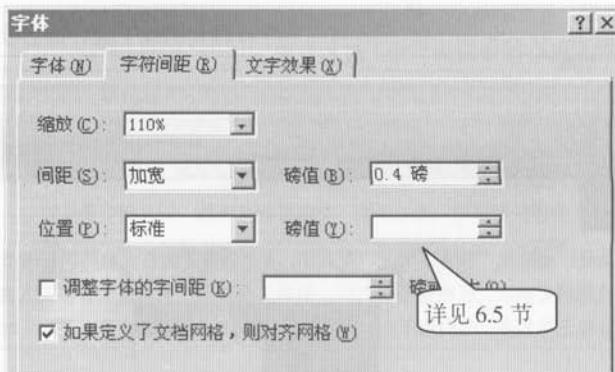


图5-4 【格式→样式(S)】 + 【更改(M)】 + 【字体】 + 【字符间距】

样式格式之二：段落

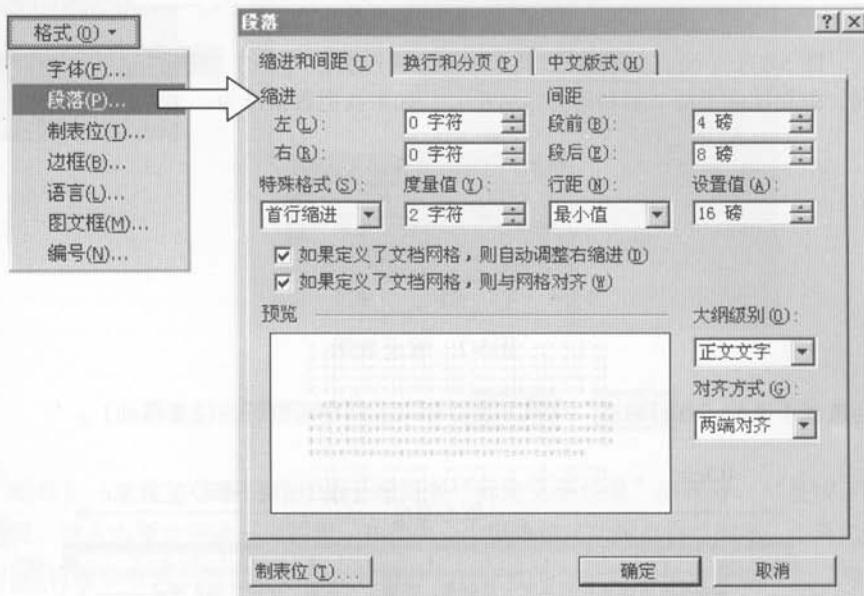


图5-5 【格式→样式(S)】 + 【更改(M)】 + 【段落】 + 【缩进和间距】

段落主要用来设置某个段落的版式属性，其中最重要的就是图 5-5 的**缩进和间距**，它所造成的效果如图 5-6。我不敢说图 5-6 的标示够精准，不过那并不重要！重要的是这些数据都需要经过用户的调整——调整到视觉满意为止就是了。

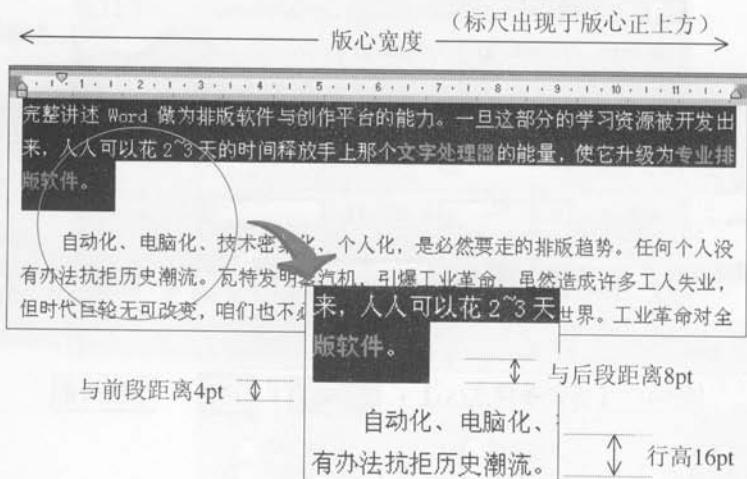


图5-6 段落的间距和行高

图5-5上半段有个缩进栏,其作用和直接控制标尺上的 作用相同,但精准度更高(可精确指定数据)。如果我把数据设为:

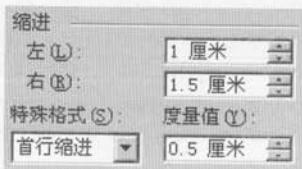
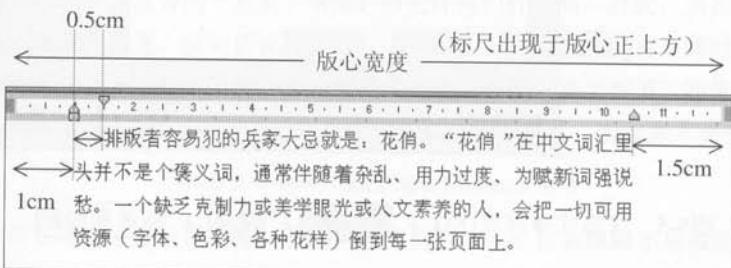
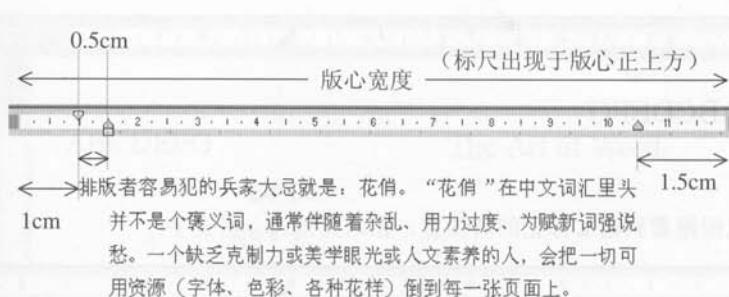


图5-7 缩进数据

会造成下面的“首行缩进”结果(注意标尺上那些定位符的位置移动):



如果保持图5-7数据不变,但将‘特殊格式’改为“悬挂缩进”,会造成下面的“首行凸排”结果(注意标尺上那些定位符的位置移动):



样式格式之三：制表位

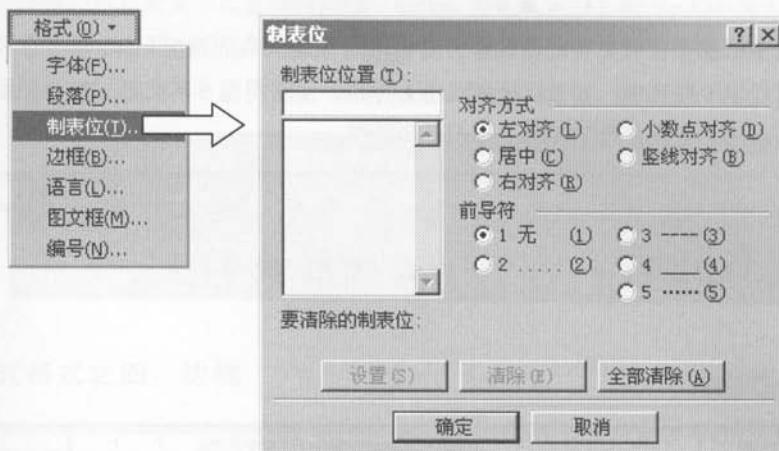
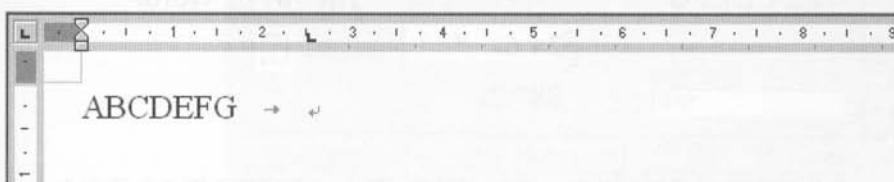
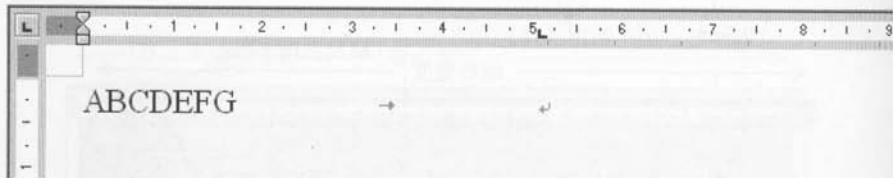


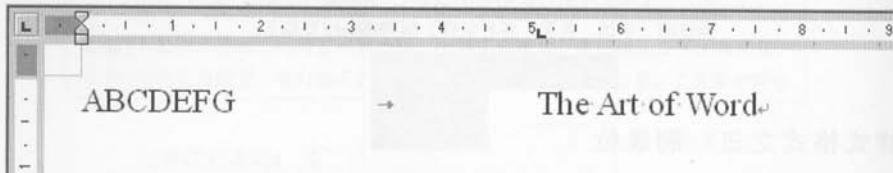
图5-8 【格式→样式(S)】 + 【更改(M)】 + 【制表位】

制表位（或称定位停驻点）是一种用来“标定文字位置”的符号。只要按下 **Tab** 键，输入点便会移动一个距离，并产生一个制表位（此时点击“常用”工具栏上的 将能够看到不可打印的制表位符号）。在此同时，将鼠标移到标尺上“与输入点大约呈垂直”的位置，点击左键，便会看到一个 **L** 标记：

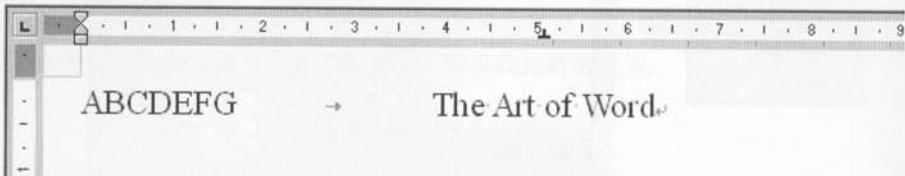




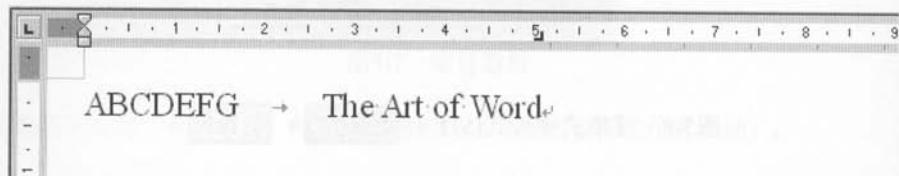
输入点也便跟着到达 L 标记的新位置。在这儿键入一些文字：



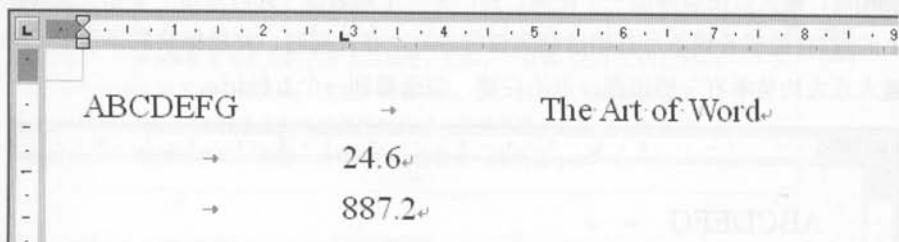
这其实完全就像在任何文字处理软件上以 Tab 控制输入点的情况一样。惟一的不同是，这里还多一些选择：如果以鼠标双击 L 标记，便获得图 5-8 画面，其中有多种对齐方式可选。上图是◎左对齐，下图是◎居中：



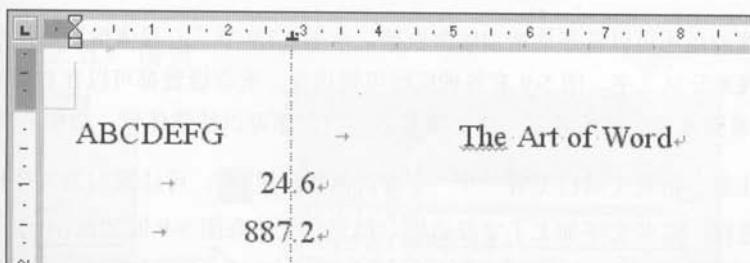
下图是◎右对齐：



现在，接下来的两行各输入一个数值如下：



并为带数值的那两行各做一次以下动作：双击 L 标记并在图 5-8 画面中点击 ◎小数点对齐方式。获得结果如下：



这样就可以把小数点位置定得很准很准。图 5-8 画面中还有一种○竖线对齐方式，和表格（5.7 节）有关，这里不讨论。

当我们打算对文字设置项目符号（bullets 如●, ■, ▶, ○）时（p.124 及 5.6 节），很可能就需要这里介绍的制表位，用以设置符号之后的文字的精准起始位置。项目文字超过一行时尤其有必要，因为项目符号往往凸排，“左缘切齐”视觉上很重要：

- Selection (选中区) 对象所代表的是文档中某个（可视的）被选中区，或 Word 之中的目前插入点（文字输入点）。此对象类似 Range 对象，但一份文档可关联许多 Range 对象，而 Word 却只该有一个 active Selection 对象。这个对象拥有“用以代表各种页面元素及各种设置”的属性（properties），包括

样式格式之四：边框

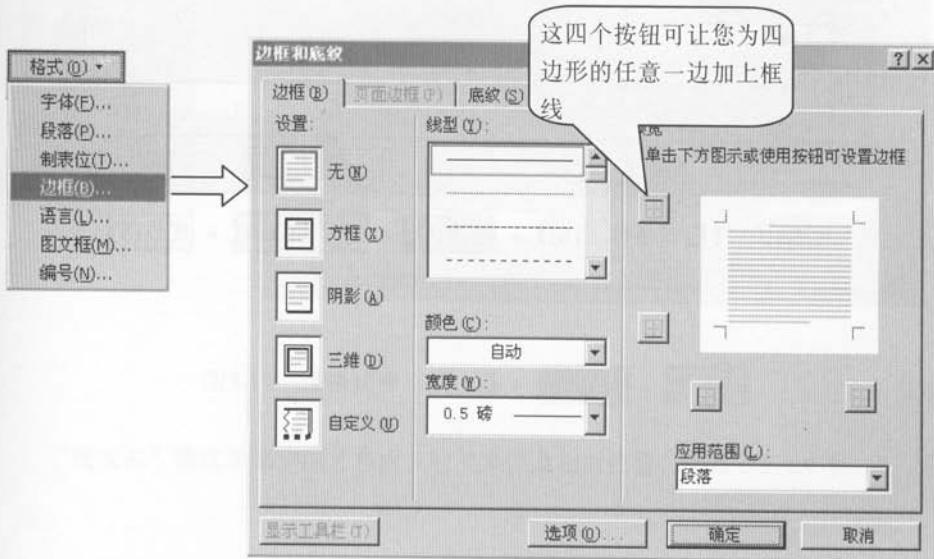


图5-9 【格式→样式(S)】+【更改(M)】+【边框和底纹】+【边框(B)】

框线的运用可以良好地增加版面质感。本书 2.3.9 节“心得偶拾”就运用了大量的框线来分隔文字。图 5-9 有各种框线可供选择。所有设置都可以在右侧的预览区直接看到效果。框线并不一定必须是封闭的，您可以任选任何一边加上框线。

请注意，格式工具栏上有一个“字符边框”按钮 **A**，可让我们为文字加上边框，像这样：这些文字加上了字符边框。但是您无法在图 5-9 的**边框(B)**中以刚才说的右侧四个按钮决定边框加于哪一边，因为字符边框一定四边都加，没得选。

图 5-10 显示**边框和底纹**中的**底纹(S)**。本书不少地方用上了底纹。

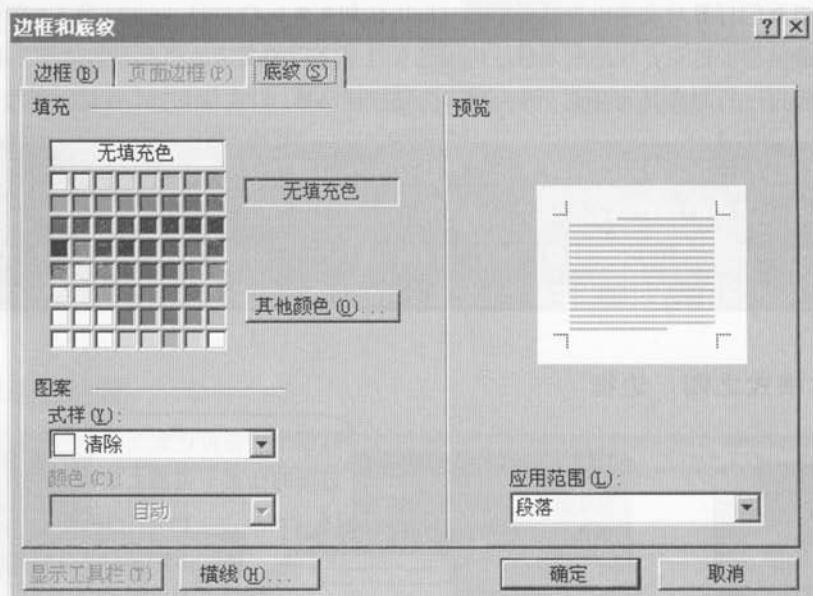


图5-10 【格式→样式(S)】+【更改(M)】+【边框和底纹】+【底纹(S)】

样式格式之五：语言

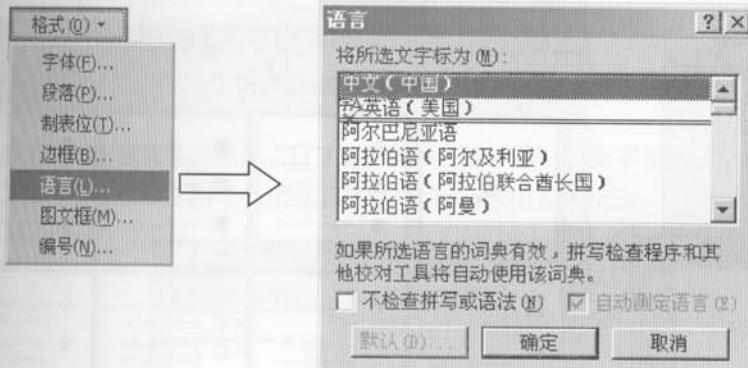


图5-11 【格式→样式(S)】 + [更改(M)] + [语言]

“语言”格式实在没什么好说的了，选择您所使用的语言便是。

样式格式之六：图文框

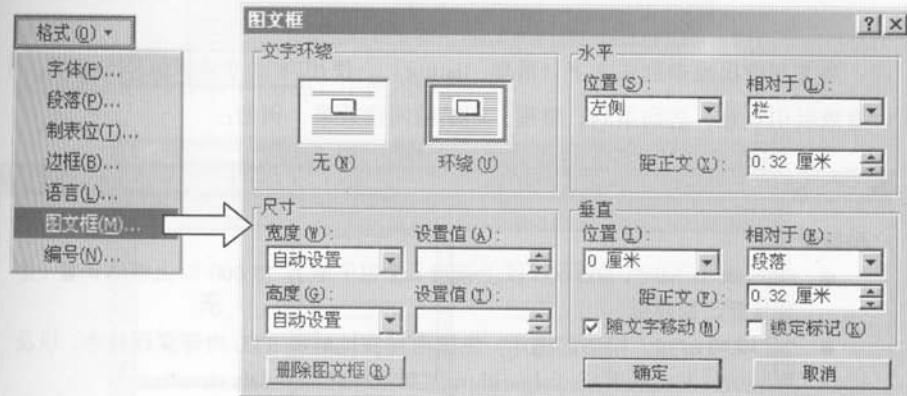


图5-12 【格式→样式(S)】 + [更改(M)] + [图文框]

“图文框”格式关系到图片与内文的关系以及相对位置，详见 6.5~6.9 节。

样式格式之七：编号

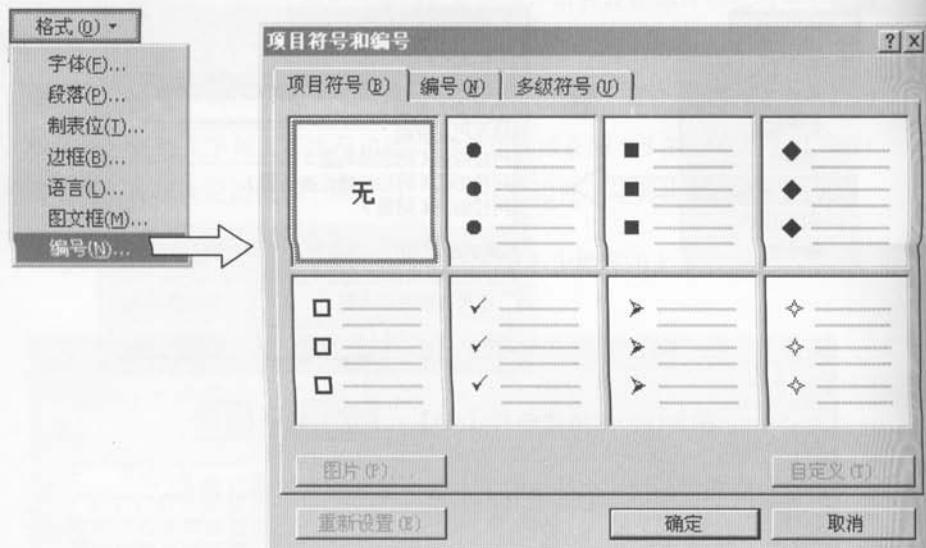


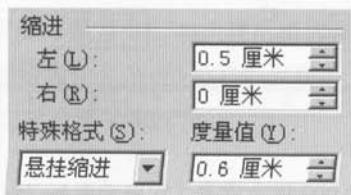
图5-13 【格式→样式(S)】 + 【更改(M)】 + 【项目符号和编号】 + 【项目符号(B)】

当文字中出现条列式文字（清单，listing），使用【项目符号和编号】最棒了，可以有效吸引眼球，达到条列、整理、归纳的阅读效果。例如：

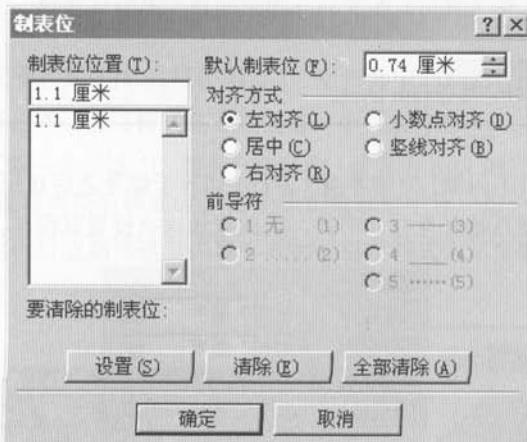
A Microsoft Word document window is shown with a bulleted list under the heading '侯捷著作：'. The list includes five items, each starting with a black circle bullet point. The text is in a standard black font, and the bullet points are aligned to the left of the text.

- 深入浅出 MFC。200 000 以上读者（含电子版），75 000 以上纸本销量（含繁体版、简体版）。
- STL 源码剖析。精彩的图示，生动而详实地解释 STL 内部实现技术，以及那些灿灿发光的算法（algorithms）与数据结构（data structures）。
- Word 排版艺术。彻底解放 Word 作为一个专业排版工具和一个辅助创作平台的强大威力。
- 大型框架设计。A dream book.

这其中比较困难的操作大概是制表位（标尺上那些小三角块和四方块）的位置。本例要求项目符号（●）内缩 0.5cm，项目文字本身（多行）切齐于项目符号（●）之后 0.6cm。整笔项目（含符号和文字）呈现凸排形式。因此应该这么设置其段落：



而为了让每个项目文字的第二行（起）能够左齐于第一行文字起始点，制表位（标尺上的那个 L 标尺，稍早介绍过）位置应该是 $0.5+0.6=1.1\text{cm}$:



下面示范如何在清单 (listing) 中使用编号:

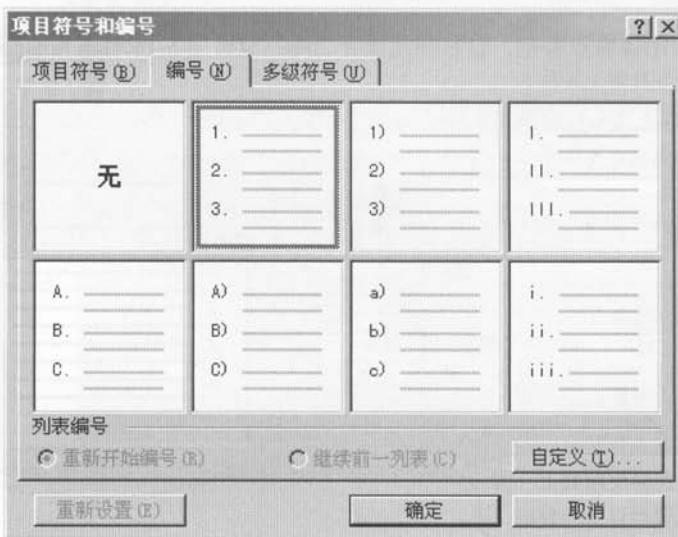
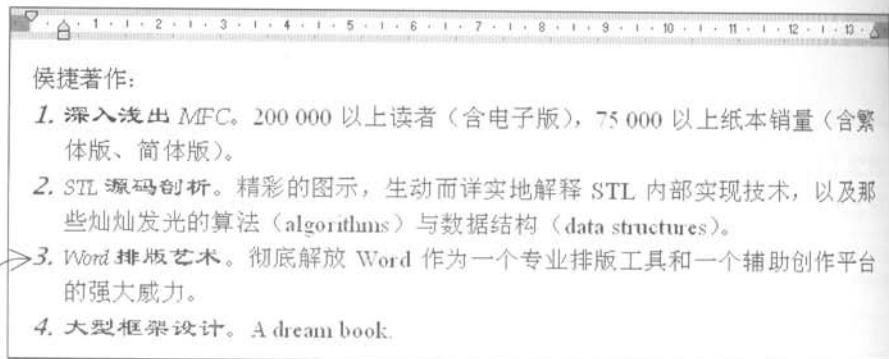
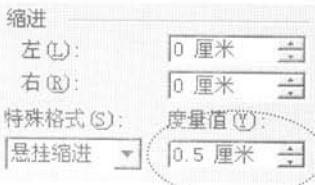


图5-14 【格式→样式(S)】 + 【更改(M)】 + 【项目符号和编号】 + 【编号(N)】

这次我示范不同的条列方式：



此次的要求是编号不内缩，文字本身（多行）切齐于编号之后 0.5cm。整笔项目（含编号和文字）呈现凸排（悬挂）形式。因此应该这么设置其段落：



无需任何制表位！因此应该点击上一页的制表位的全部清除(A)按钮。

本例的编号字体有点特别，是在哪儿设置的呢？点击图 5-14 右下角的自定义(T)：

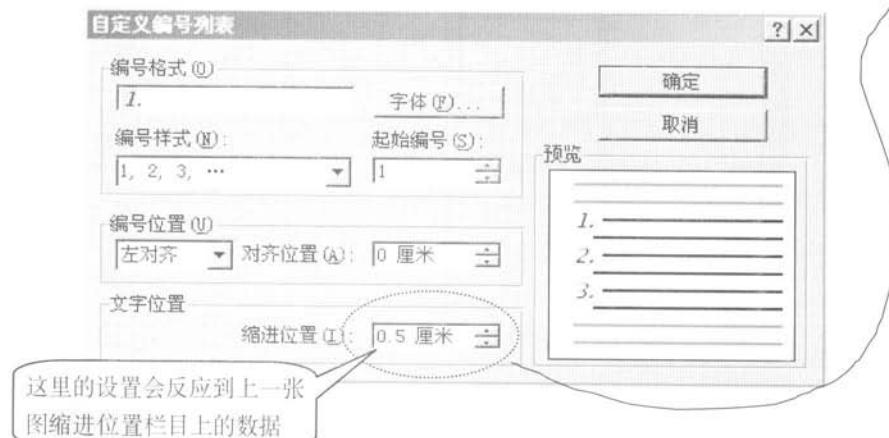


图 5-14 的另一个页签【多级符号】，与上述所说的【项目符号】和【编号】的功用和影响都不可并提。我不太理解为什么 Word 把它们放在同一个对话框中。【多级符号】关系到章节的自动化编号，用法请见 5.11 节。

5.1.1. 新增段落样式

以影响范围来划分，样式可分为段落样式和字符样式两种。前者作用于被圈选的整个段落身上，后者只作用于被圈选的一段文字（字符）身上。

点击【格式→样式(S)】后可得图 5-1，点击其中的【新建(N)】，可得以下画面。从画面上可以看出来，段落样式名称之前有一个代表符号 \leftarrow 。通常我会为自定的（新增的）样式取一个“以 ‘!’ 起头”的名称（请参考 p.87 的“样式命名考虑”），以便【样式】对话框排列所有样式时，自定样式能够集中在一起，也使我自己能够在看到一个样式名称后立刻知道是内建样式抑或自定样式，不必瞎猜。

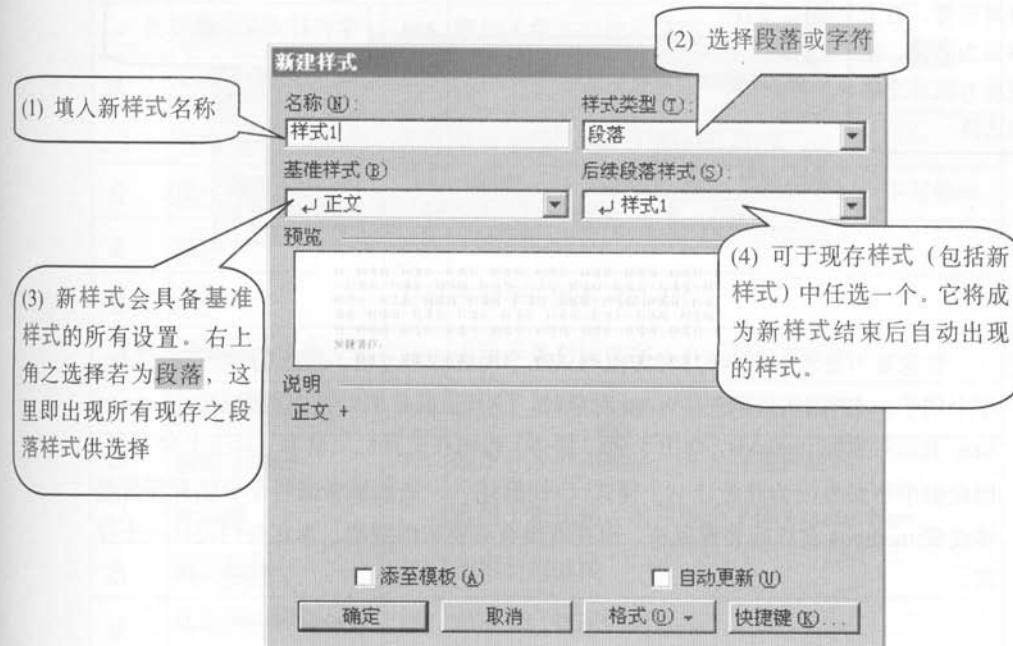


图5-15 新建一个段落样式

5.1.2. 新增字符样式

字符样式作用于被圈选的一段文字（而非整个段落）身上。以此种样式来表现计算机书籍中的各类术语，极为方便（请参考 p.86 “快捷键的取舍”）。

点击【格式→样式(S)】后可得图 5-1，再点击其中的【新建(N)】，可得以下画面。从画面上可以看出来，字符样式名称之前有一个代表符号 **a**。

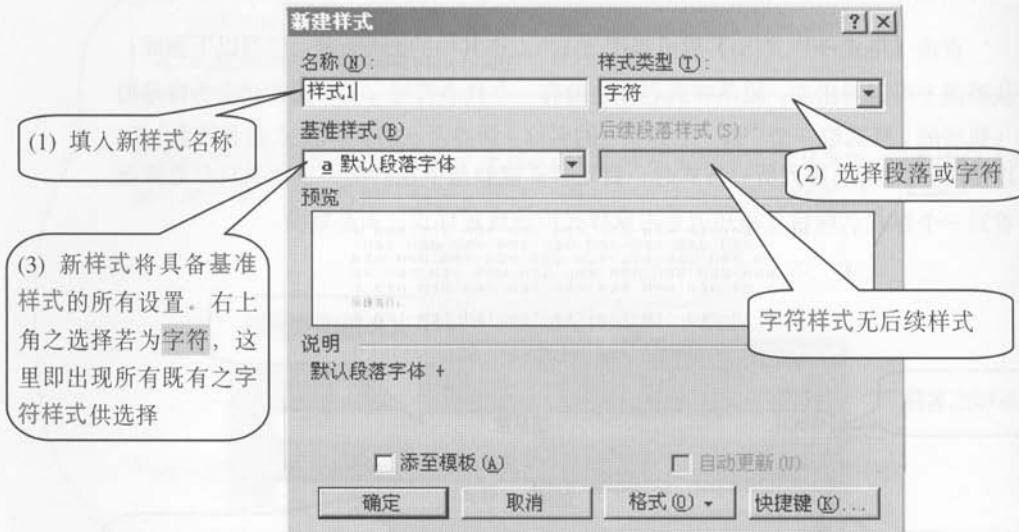


图5-16 新增一个字符样式

究竟要不要建立新样式，考虑点不只在于现在方不方便，还要考虑将来的弹性。举个例子，本书 10.3 节谈到 Word 对象时，我只是简单地以斜体字如 *Go*, *Execute*, *Item* 表示对象的 methods，而由于 **I** (斜体) 按钮在工具栏上就有，点击十分方便，因此似乎不必专门为此设计一个样式 (+快捷键)。但如果考虑将来可能有需要整体改变 methods 名称的表现风格，现在就很有必要未雨绸缪，为它专门设计一个样式。

5.1.3. 内建样式 (Built-in Styles)

Word 允许我们为任何自定样式取任何名称。除此之外 Word 也已经为我们准备了一些预先设计好的内建样式，见表 5-1。

表5-1 若干 Word 内建样式的名称及用途

种类 ¹	样式名称	参考页次	用途
↓	正文	18	套用于主体文字
↓	标题 1-标题 9	205	套用于章名、大标题、中标题、小标题……
↓	索引 1-索引 9	224	套用于 1-9 层索引
↓	目录 1-目录 9	207	套用于 1-9 层目录
↓	题注	160	套用于题注
↓	图表目录	214	套用于各种（图、表……）题注所产生之目录
a	脚注引用	145	套用于脚注的代表符号（安插于正文之中）
↓	脚注文本	145	套用于脚注的说明文字（每页最下）
a	尾注引用	145	套用于尾注的代表符号（安插于正文之中）
↓	尾注文本	145	套用于尾注的说明文字（每章最后）
a	批注引用	338	套用于批注的代表符号（安插于正文之中）
↓	批注文本	338	套用于批注视窗中的批注文字
a	默认段落字体	56	任何字符样式的根据样式（母样式）。不可修改
a	页码	109	套用于页码
a	行号	102	套用于自动行号
↓	列表编号 1-5	301	适用于条列式文字；有五种悬挂距离
↓	列标项目符号 1-5	301	适用于条列式文字；有五种悬挂距离
↓	列表-列表 5	301	适用于条列式文字；有五种悬挂距离
↓	宏文本		适用于宏代码，采用等距字体 Courier New.
A	超级链接	334	套用于超链接
a	已访问的超级链接	334	套用于已访问的超链接

这些默认样式有重要意义。例如制作目录时（第 6 章），Word 会自动抽取**标题 1**、

¹ ↓ 表示段落样式，a 表示字符样式。

标题 2、标题 3……样式的文字内容做成目录，并以**目录 1、目录 2、目录 3……**样式来表现各层目录；又以相同的目录内容做成文档结构图（图 2-1），并以**正文**样式来表现。

5.1.4. 样式与快捷键

p.86 “快捷键的取舍”提到了我对快捷键的看法。快捷键非常适合搭配“被频繁运用”的样式（而字符样式往往被极为频繁而随机地使用）。

点击【格式→样式(S)】后，在对话框（图 5-1）最下方，无论点击**更改(M)**或**新建(N)**都可以看到**快捷键**（例如图 5-2）。点击**快捷键**后得到以下对话框：

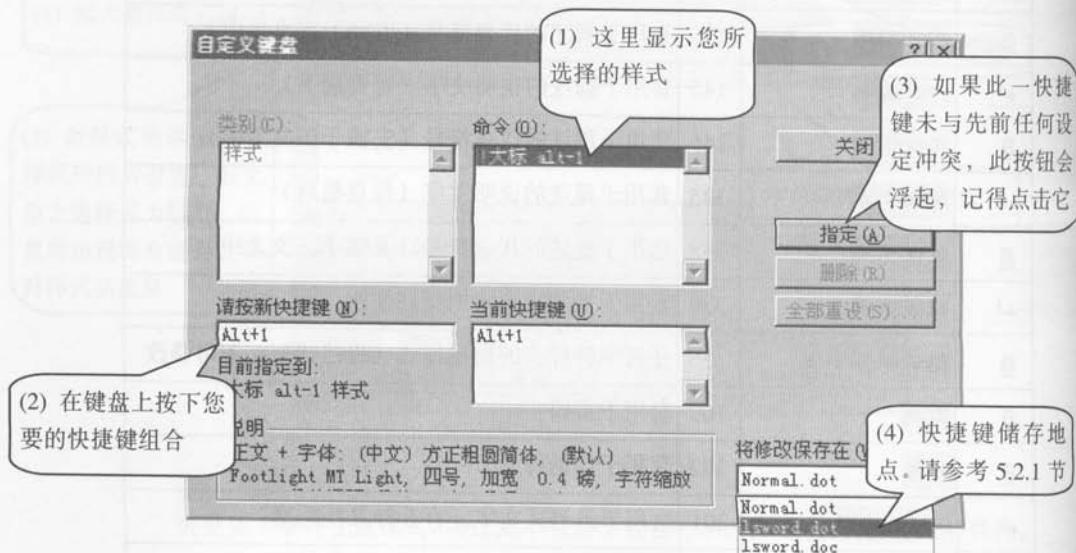


图5-17 自定义键盘

通常我会尽量将快捷键标示在样式名称中，以便将来不遗忘。

5.1.5. 将样式套用于既有文字

如果您有很多频繁出现、就版式而言类型相同的文字，例如正文、大标、中标题……（它们都是段落样式），或是面向对象编程（Object Oriented Programming）书籍中频繁出现的 class 名称、function 名称（它们都是字符样式），如何才能保持最佳工作效率呢？我的意见是，将输入点移到那些文字身上而后点击工具栏的**样式**下拉框 **自定内文** 并于其中选择一个样式，不但手续有点冗赘，Word 反应

速度也很慢，甚至即使是在高档而且内存充裕的电脑上。不信你试试。

改善速度的最佳办法就是快捷键（5.1.4 节）！

但有时候我们没有设置快捷键——或许因为有点懒，或许因为好按又容易记忆的按键组合已经没有了。那么次佳办法就是使用工具栏上的  (格式刷) 按钮。

如果要复制段落样式的格式，首先圈选原始段落。必须将整个段落圈选出来，而非只圈选其中一部分，或只是圈选所有文字：

如果要复制段落样式的格式，首先选中原始段落。必须将整个段落选中出来，而非只选中其中一部分，或只是选中所有文字：+

↑ 状况一：这样只是圈选部分文字

如果要复制段落样式的格式，首先选中原始段落。必须将整个段落选中出来，而非只选中其中一部分，或只是选中所有文字：+

↑ 状况二：这样只是圈选全部文字（将鼠标从头扫到尾）

如果要复制段落样式的格式，首先选中原始段落。必须将整个段落选中出来，而非只选中其中一部分，或只是选中所有文字：+

↑ 状况三：这样才是圈选整个段落（鼠标在段落较远左侧进行圈选）

注意，比状况二多了最后一格，那是个不可见的分段符号

接下来点击 ，光标于是变成“扫把”状。此后便可移动鼠标，在您希望设置的目标段落上按一下鼠标左键，该处的整个段落格式就会和原始段落相同。

这里所说的办法是针对“跳跃的区域”而言。我们完全可以不必对逐步进行的正文做这种设置动作——那可真是说多烦就有多烦呢。在段落样式设置画面上（图 5-15）有一个后续段落样式栏目，只要将它设为 **正文**，就解决了。

如果要复制字符样式的格式，首先请圈选原始字符(s) 然后点击 ，光标于是变成“扫把”状。移动鼠标，圈选出您希望设置的目标字符(s)，后者的格式就会和原始字符(s)相同。

的寿命只供一次使用。换句话说如果您有三个地方需要复制先前出现过的样式（不论是段落样式或字符样式），您必须做三次点击（单击） 的动作。但如果改为双击，鼠标形状就会持续变成“扫把”，您可以用它到爽为止。按下 Esc 才会使鼠标恢复原本形状。

5.2. 模板 (Templates)

我们总希望自己的努力有着重复使用性，于日后不断地被使用。面对一个已经设置好各种样式（乃至纸型、页边距、节、页眉、页脚……）的文档，将来如何重复使用那些辛苦调整好的设置呢？

办法之一是，把这份文档 (.doc) 拷贝一份，删除其中内容（留下的是各种不可见的设置），再在其中长肉。这种做法四平八稳，但不怎么灵敏，操作也嫌麻烦了点。

第二种做法是：将文档储存为所谓的模板²（样式库），如图 5-18。模板是一种以 .dot（而非.doc）为后缀名的文件。储存为模板后您可以将其内容全部移除，或只留下一些作为示范或提示（就像 p.63~p.66 所示各图），以利将来沿用或修改各种设置时有所参考。图 5-18 一旦选择保存类型为“文档模板”，储存路径会自动跳至 \Program Files\Microsoft Office\Templates（或“Word 安装路径\Templates”）处，那儿是模板的大本营，详 5.2.2 节。如果模板不放在该处，就没有一丝用途。换句话说 Word 只在那儿找模板。

作为一个模板，好处就是它成了一个“模板/范本”：用户想要产生任何新文档时，都可以挑选某个模板来用，并因此继承（沿用）了该模板中的所有设置（包括页面设置、页眉/页脚、样式、快捷键、宏、自定义工具栏）。

² Word 所说的模板，原文是 template，这个字眼在编程领域中也常称为模板。模板一词似乎容易让人联想起建筑工地那些以木板或钢架做成的协助水泥定形并坚固的外模和支撑，然而其真实意义比较接近“模具”。一个模具可做出许多一模一样的东西。



图 5-18 将.doc 另存为.dot

事实上，建立新文档（开新文件）的途径有两种，都离不开模板：

1. 点击“常用”工具栏上的【新建空白文档】。不经任何询问您便直接获得一份新文档。此新文档必定从 Normal.dot 衍生而来（或者说它继承了 Normal.dot 的所有设置）。
2. 点击【文件→新建(N)】，对话框中出现环肥燕瘦的许多模板（也许是 Word 内建，也许是用户自定）任您挑选。产生出来的新文档会继承该模板的所有设置。5.2.2 节对 Word 内建的模板有一些介绍。

图 5-19 说明了模板与文档的关系。这张图显示任何文档都衍生自某个模板（如果没有特别指定那就衍生自 Normal.dot），而所有模板又都衍生自 Normal.dot。也因此，只要在 Normal.dot 中加入任何设置，便会影响每一个模板，也就影响此后 Word 所产生的每一份文档。

模板甚至也能衍生模板（就好像各个模板都继承自 Normal.dot 一样）。子模板继承父模板的所有特性，并可加上自己的新设置。但是请注意，继承而得的各种特性设置，已经与其父模板或甚至祖模板脱离了脐带关系——那些设置都是以嵌入（embedded）方式进入子模板或文档中，不是以链接（linked）方式进入。换句话说如果用户在文档或子模板内修改某些继承而来的性质，并不会影响其父模板。

那么，如果用户将自己的文档拿到其他机器执行，而新机器上并没有该文档的父模板，会有什么后果呢？为了回答这个问题，我做了个实验。

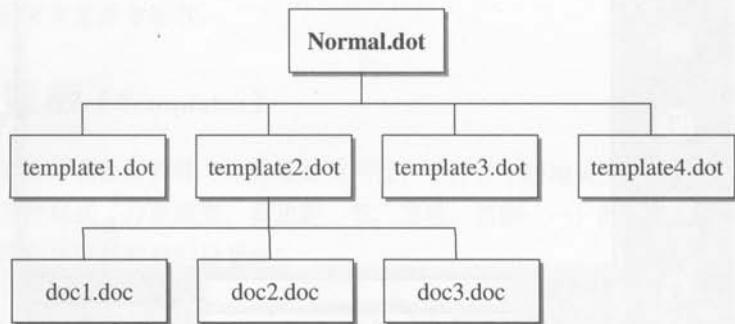


图5-19 模板（templates）与文档（documents）的关系

关于模板（Templates）的一个实验

首先点击 **□**（新建空白文档），未经任何询问即获得一份空白文档。点击【格式→样式(S)】可看到，所有样式都是 Word 内建样式，也就是置放于 Normal.dot 中的样式。接下来建立一个新的段落样式**模板测试**（做法请参考 5.1.1 节），和一个相应的快捷键（做法请参考 5.1.4 节），以及一个宏（做法请参考 10.1.1 节）和一个自定义工具栏（做法请参考 10.1.4 节）。后三者的设置过程中都允许我们选择储存地点（详 5.2.1 节），我都选择储存于文档内。

接下来将上述文档储存为 A.doc，并另存为 B.dot（模板）。

接下来将文档拷贝一份 C.doc。实际操作显示，这份复件拥有原件的所有设置，包括样式、快捷键、宏、自定义工具栏。

接下来点击【文件→新建】（图 5-22），选择稍早做出的.dot 为根据模板做出一份新文档 D.doc。实际操作显示，这份文档也有先前的所有设置，包括样式、快捷键、宏、自定义工具栏。

现在将 C.doc 和 D.doc 拷贝到另一台电脑，但不拷贝 B.dot。在新电脑上我们发现，C.doc 仍然拥有的所有设置，包括样式、快捷键、宏、自定义工具栏。D.doc 则只剩下样式，至于快捷键、宏、自定义工具栏全部流失了。如果将 B.dot 也拷贝

到新电脑，D.doc 的一切设置便获得了恢复。图 5-20 表现整个实验过程。

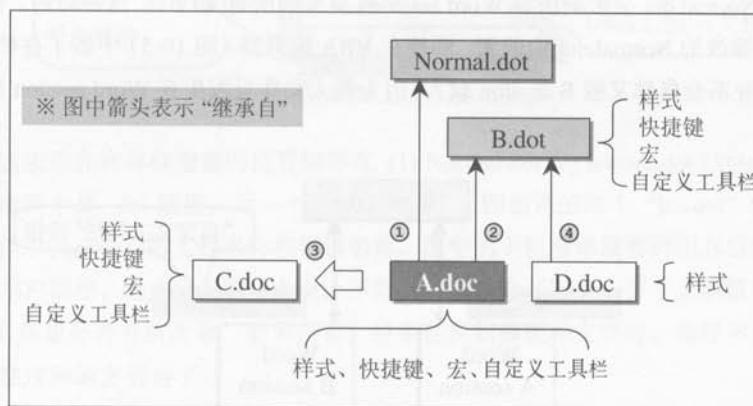


图5-20 模板 (Template) 实验之一

这个实验说明了一些事实：

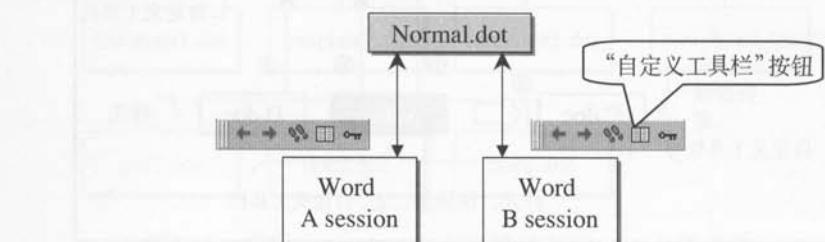
- 用户可设置的属性包括 (1) 样式、(2) 快捷键、(3) 宏、(4) 自定义工具栏。
- 用户的这些设置可存放于 (1) 自身文档、(2) 文档所据模板、(3) Normal.dot 三者。设置过程中都有机会加以选择（详见 5.2.1 节）。
- 储存于文档内的设置，随着文档而走（那当然 ☺）。
- 如果以模板为根据产生新文档，那么储存于模板内的四种设置之中只有样式会下放到文档内，随文档而走，其余都保留在模板内（但会对所生文档施放影响）。这种情况下如果模板未能和文档共同存在，文档将失去快捷键、宏、自定义工具栏。

关于模板 (Templates) 的第二实验

这个实验是我无意中的一个体会。本书 10.4 节开发“程序代码加行号”宏时，原已完成第一阶段，贴附于 Normal.dot 中，并为它设了一个工具栏按钮（见 10.1.5 节的图标）。而后由于有了某些新想法，我又增加 10.4.3 节“改变注释颜色”。当时为求方便，也为了怕互相干扰，我启动两个 Word，A session 载入本书电子档，B session 载入测试文档。两个 Word sessions 一开始便都载入 Normal.dot，也就都有了代表上述宏的那个工具栏按钮。而后我在 A session 修改宏，并以为由于 Normal.dot 会影响所有 Word sessions 里头的所有文档，因此企图在 B session 中将该宏作用于

测试文档，却发现总是不起作用。

“Normal.dot 会影响所有 Word sessions 里头的所有文档”，这是对的。但当 A session 修改的 Normal.dot 内的宏，即使在 VBA 编辑器（图 10-5）中做了存档动作，新版本并不会自动又被 B session 载入，因为载入动作只发生在 Word session 启动之际。



A session 对 Normal.dot 所做的任何修改，都不会影响 B session。除非 B session 的启动是在 A session 对 Normal.dot 的修改之后

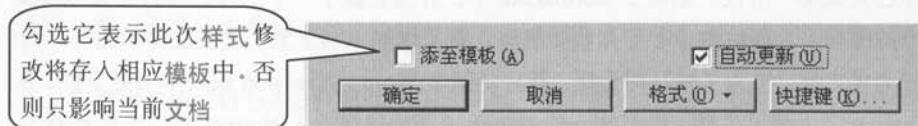
图5-21 模板（Template）实验之二

5.2.1. 各种设置的栖身规则

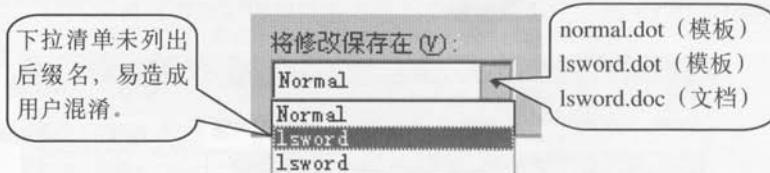
任何可被储存的设置（样式、快捷键、宏、自定义工具栏），都有以下三种可能的储存场所，每一种各有其影响范围（或说作用范围、scope）：

- 储存于文档。影响范围：文档自身。
- 储存于文档赖以为据的模板。影响范围：同一模板衍生的所有文档。
- 储存于 Normal.dot。影响范围：所有 Word 文档。

举个例子，点击【格式→样式(S)】得图 5-1，而后选择某一样式，而后点击【更改(M)】，您会看到对话框下方出现一个询问：

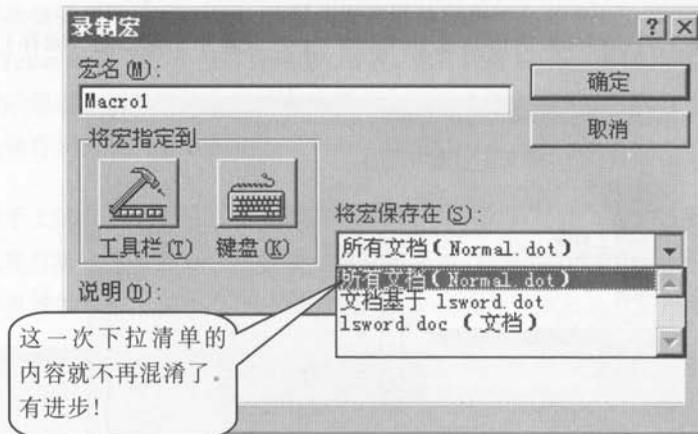


再点击图 5-2右下角的【快捷键(K)】，在【自定义键盘】右下角看到：

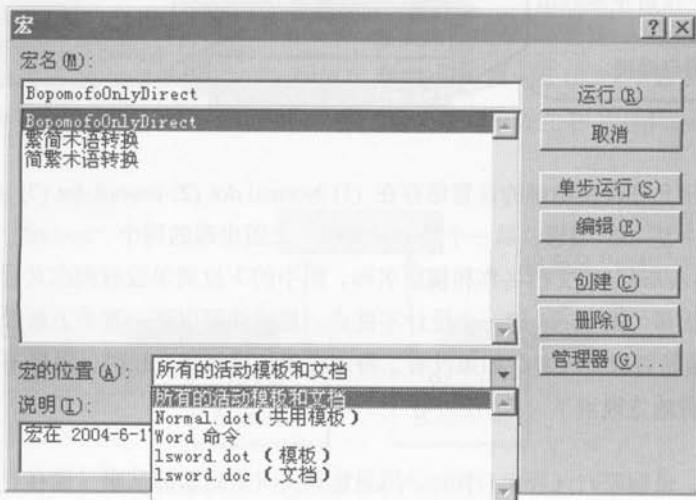


这表示允许将快捷键的设置储存在 (1) Normal.dot (2) lsword.dot (3) lsword.doc 中, 前两个是 .dot 模板, 后一个是.doc 文档。上图出现的两个 “lsword” 便是当前 (工作中, *active*) 的文档名称和模板名称。图中的下拉清单没有列出其后缀名, 易造成用户混淆, 是 Word 的一个设计不良点。您或许可以查一查手上新版本, 看看 Word 新版是否有所改善。如果没有, 将来在规划模板和文档时, 最好不要同名, 以免在这种地方混淆了。

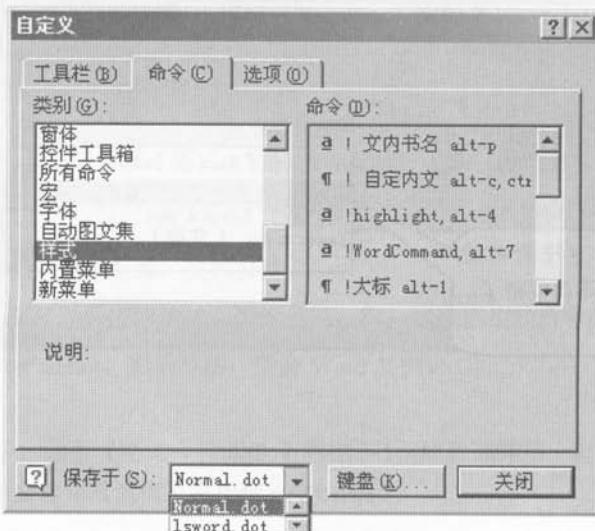
此外, 录制宏时 (10.1.1 节), 用户也被要求指定宏的贴附 (储存) 场所:



新建宏时（10.1.1 节），用户也被要求指定宏的贴附（储存）场所：



当我们自定义工具栏时（做法请参考 10.1.4 节），也被要求指定贴附（储存）场所：



5.2.2. Word 内建的模板

点击【文件→新建(N)】，可获得对话框如下：

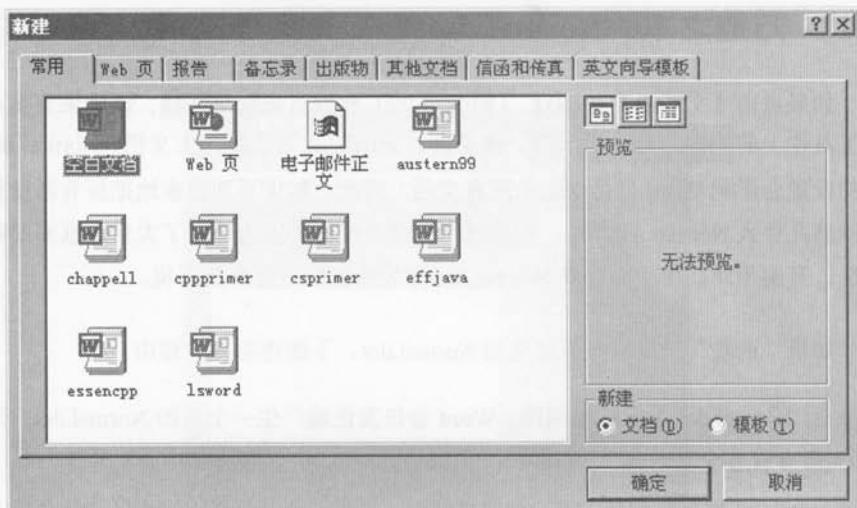


图5-22 模板集中地，在Word中观察

这些模板都被放在Word的默认安装目录（文件夹）“Program Files\Microsoft Office\Templates”中，不但有针对Word者，也有针对Microsoft Office其他软件者。先前曾经提过，如果您将文档储存为模板，储存路径会被自动设置至这里。当然您也可以储存到任何您喜欢的地方，但除了这里，模板不起作用。

我手上的资料告诉我：图5-22的每一个`页签`对应于图5-23的每一个文件夹`图标`；如果打算在图5-22建立一个自定的模板分类，只要在图5-23的“Templates”文件夹中再增加一个文件夹（子目录）即可。不过我并未实验出这样的结果⑧



图5-23 模板集中地，以资源管理器观察

5.2.3. 万范之本：Normal.dot

如果点击【文件→新建(N)】（如图 5-22）然后点击空白文档，或如果直接点击工具栏上的新建空白文档 □，便是以 Normal.dot 为依据产生文档。Normal.dot 内的设置会影响 Word 日后产生的所有文档。因此，如果不加思索地把所有新设置一股脑儿存入 Normal.dot，可以想见这个模板很快就会因为储存了大量信息而显得混乱。有鉴于此，很少人会拿 Normal.dot 作为他们的主要开发环境。

如果“混乱”不足以吓阻您使用 Normal.dot，下面还有两个理由：

1. 一旦 Normal.dot 损毁或被删除，Word 会自发性地产生一个新的 Normal.dot，原先所有设置全部消失。
2. Normal.dot 储存的信息愈多，身形愈臃肿。而我们知道每次 Word 启动都会载入 Normal.dot，也就因此花费较多的启动时间（用以载入臃肿的它）。

但是该做的还是要做，不必踌躇。第 10 章的三个宏（为程序代码加行号、两岸术语转换、文档字体统计）性质上适用于所有文档，所以我把它们都加到 Normal.dot 去。当然，那也就是说，如果用户换了台机器，Normal.dot 必须跟着文档一起拷贝过去。

5.2.4. 全局模板 (Global Templates)

有时候我们或许希望制作出“Word 启动后立即可用”的宏。Normal.dot 似乎是储存它们的惟一场所。其实另有好地点：全局模板。这东西被载入于 Word 之中（而非任何文档内），也因此其中所有设置可被任何文档使用。甚至即使尚未有任何文档产生，用户也可以使用它们，例如用来辅助（强化）开启文档的动作。

全局模板的制作与一般模板没有两样，惟一的差别在于它们被 Word 载入的时机，以及作用范围的不同。此外，用户无法将文档中的修改“上传”至全局模板而改变其内容，因为全局模板对文档而言是只读的！（它们被全民共享，不允许被单独谁谁谁改变）。改变全局模板内容的惟一办法是：开启它，修改它。

建立一个 Normal.dot 之外的全局模板，契机在于【工具→模板和加载项(I)】，如图 5-24。您可以点击其中的添加(D)，而后在文件列表中选择一个 .dot 文件，使其成为全局模板。直接把“打算成为全局模板”的那些模板搬移放进 Word 安装路

径下的“Startup”文件夹，也可以达到目的。

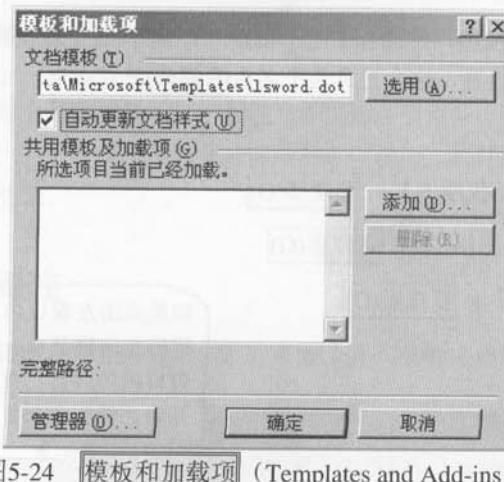


图5-24 模板和加载项 (Templates and Add-ins)

12.8 节谈到的 PDFMaker 被认为是一个 Word 宏，的确，如果您的 Word 安装了 Acrobat，便会有一个 PDFMaker.dot 出现在上图的共用模板及加载项列表中。

点击以上 **添加(D)** 之后所得画面如下，要求用户选择一个“将被当成全局模板”的模板：



5.2.5. 模板 (Templates) 的管理

5.2.1 节提过，模板内部储存了若干设置。Word 同时也提供相关工具，让用户管理模板内的那些设置。这个工具是个名为**管理器**的对话框，如图 5-25，可由数种途径进入：

1. 【工具 → 模板和加载项(I)】 + **管理器(O)**
2. 【工具 → 宏(M) → 宏(M)】 + **管理器(G)**
3. 【格式 → 样式(S)】 + **管理器(O)**

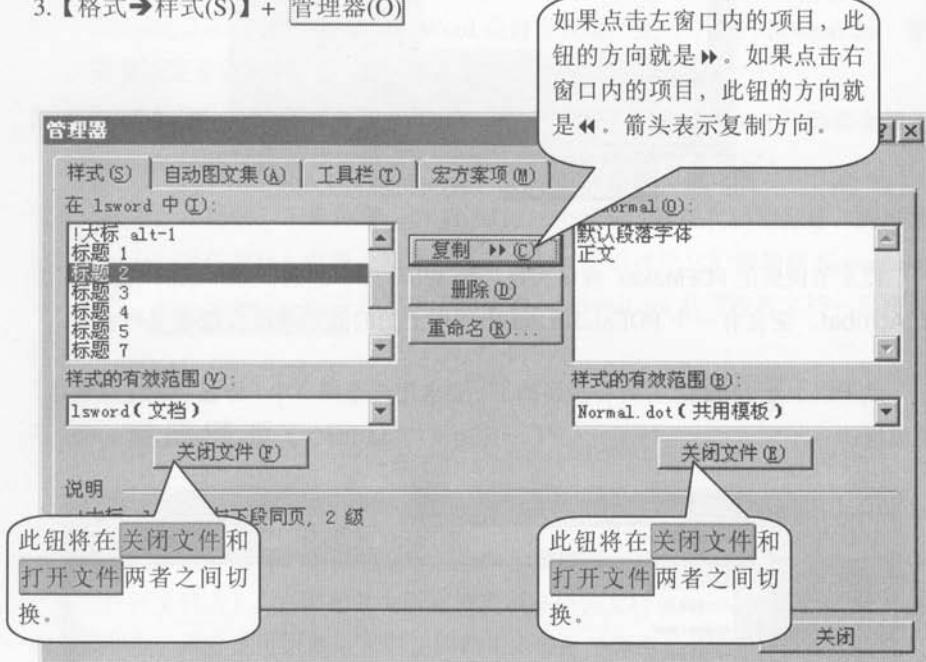


图 5-25 **管理器**可以管理模板 (templates) 内的各种设置

我们可以分别在图 5-25 左侧和右侧各指定一个文档或模板(默认情况下左边是当前文档，右边是 normal.dot)，然后点击上方不同的四种页签(**样式**、**自动图文集**、**工具栏**、**宏方案项目**)，再运用图中央的**复制(C)**、**删除(D)**、**重命名(R)**三个按钮，处理各种设置。

5.3. 脚注 (Footnotes)

脚注是正文的补充。也许您已经看过脚注。如果没有印象，我现在就给您一个：³

上一行最后那个号码就是所谓的脚注（或说脚注引用）：可以是数字，也可以是符号如 □ 或 ▷，会随文字的增减而自动对应移位。本页最下方以横线分隔的则是所谓的脚注文本。

5.3.1. 插入脚注

将光标移到欲插入脚注的地点，然后点击【插入→脚注和尾注 (Y)】，便得以下画面：

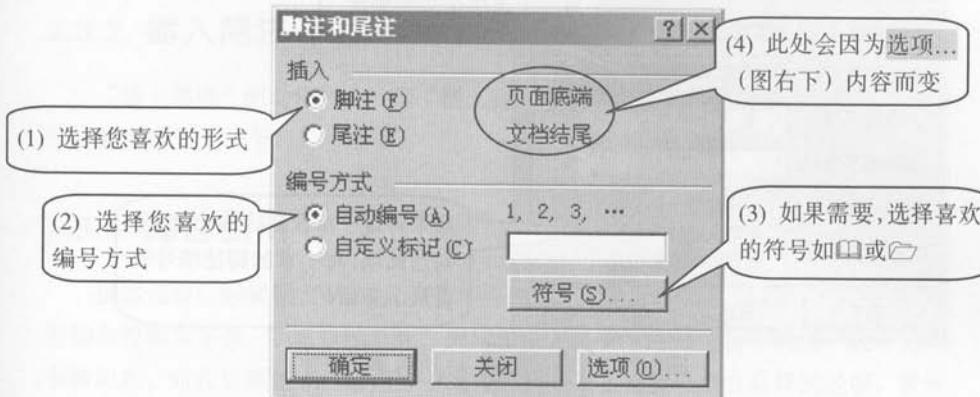


图5-26 脚注和尾注之一

这个对话框的右上半部表示：所谓脚注是放在页面底端，所谓尾注则是放在文档结尾。但这份文字其实会随着图右下的选项(O)而变化。下面是一个不同的例子：

³ 此处出现的就是所谓的脚注文本。

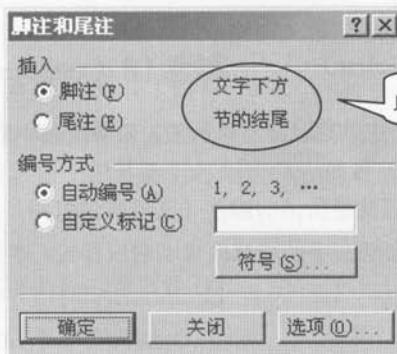


图5-27 脚注和尾注 之二

现在，点击右下角的**选项(O)**，获得以下画面（针对脚注）：

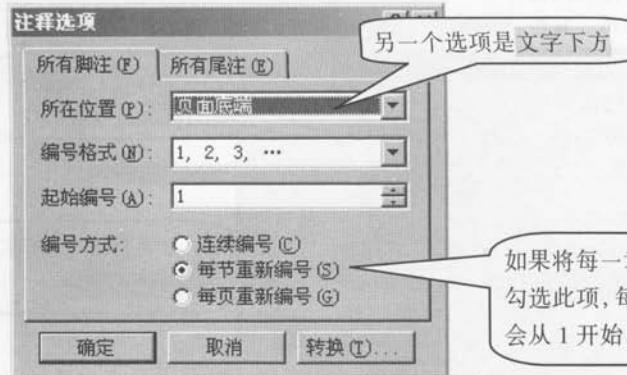


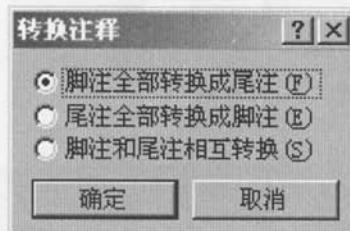
图5-28 注释选项 + 所有脚注(F)

或是获得以下画面（针对尾注）：



图5-29 注释选项 + 所有尾注(E)

点击以上二图的右下角按钮**转换(T)**，得以下画面，可转换脚注和尾注：



请注意，如果不甚清楚以上各种选项关系，很可能做出奇怪的东西。例如您决定设立一个尾注，并选择放在**文档结尾**，却又在**编号方式**中选择**每节重新编号**。结果呢，最终会出现重复编号（例如 1,1,1,2,2...）的尾注。

5.3.2. 插入尾注

“插入尾注”的动作和上一节“插入脚注”完全相同，只是某些选择有了改变。已一并说明于上一节。

5.3.3. 脚注引用

虽然先前已经说明，Word 允许我们将脚注（引用）表现为“数字”或“符号”，但如何控制其字体、字高等细节呢？例如您手上这本书的脚注编号就比一般常见的书籍粗些，而且呈蓝色——因为作为读者的我经常在阅读书籍有这样的经验：看到了脚注文本，却找不到脚注编号。

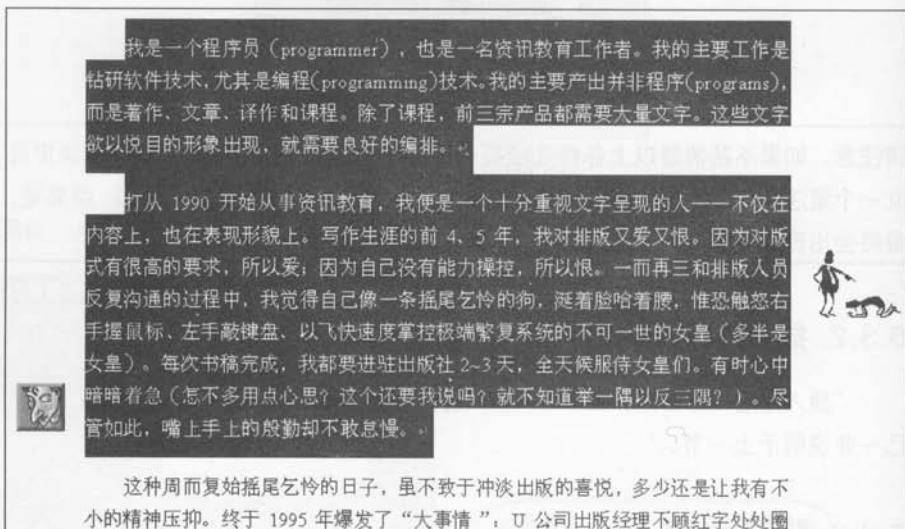
Word 内建有一个**脚注引用**样式（请参考表 5-1），只要调整它，就可以按您的心意来呈现脚注的形貌风格。本书**脚注引用**的设置记录于表 1-2。

5.3.4. 脚注文本

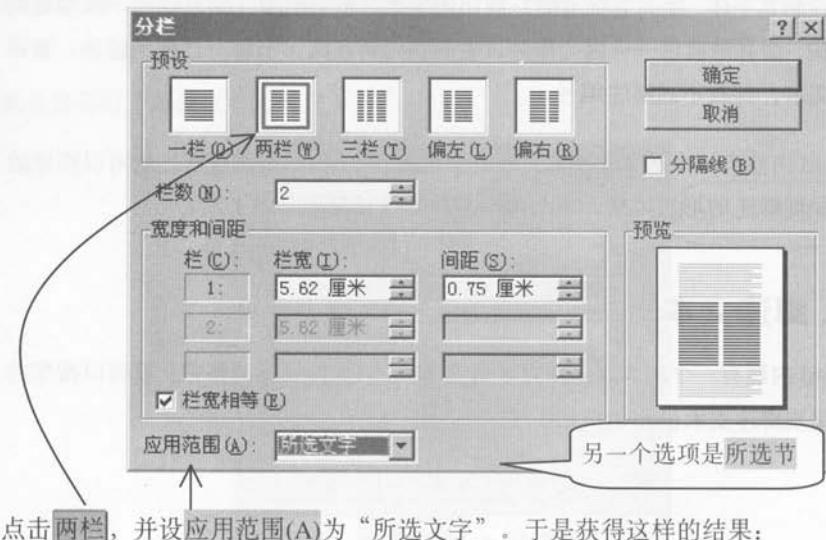
Word 内建有一个**脚注文本**样式（请参考表 5-1），只要调整它，就可以按您的心意来呈现脚注文本的形貌风格。

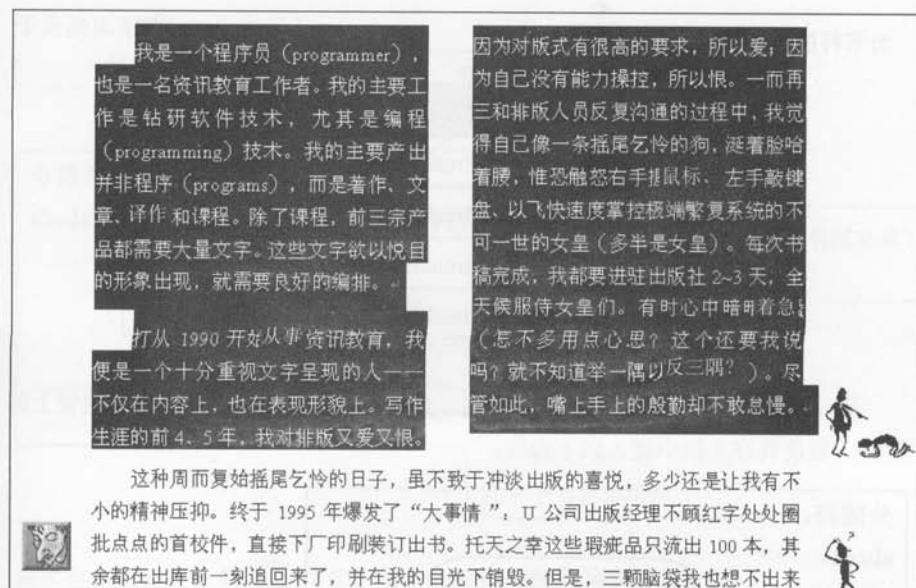
5.4. 多重栏位 (multi-columns)

多重栏位是一种比较活泼的版式，通常用于杂志、刊物、论文上头，较少出现于书籍。假设原先有这样的文字：



圈选了两个段落的文字（如上）后，点击【格式→分栏(C)】，获得如下对话框：





由于版面的变动，邻近相关的一些图片位置可能会跑位，需要再拖拉调整一下。

多重栏位的形成，是因为其中插入了分栏符。详见 5.5 节。

5.5. 分隔设置

您随时可以加上分隔符，将版面控制得更称心如意。分隔符有四种，分别是：换行、分栏、分页、分节四种。按下 **Enter** 后可加入换行符。这四种均可用【插入 → 分隔符(B)】插入于输入点：

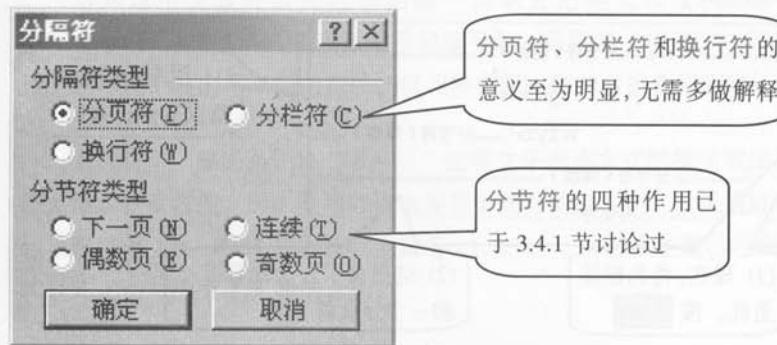
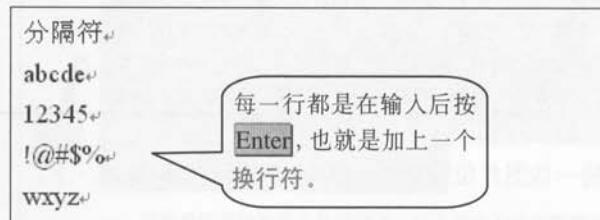


图5-30 分隔符

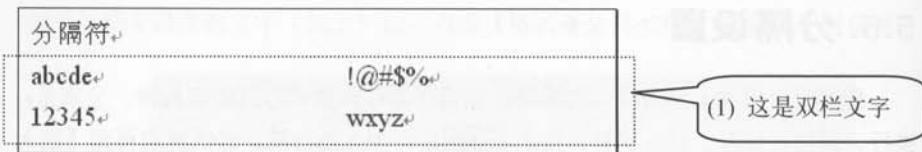
分节符的四种类型是：

分节符	Word 对象模型（10.3 节）中的代号
下一页	wdSectionBreakNextPage
连续	wdSectionBreakContinous
偶数页	wdSectionBreakEvenPage
奇数页	wdSectionBreakOddPage

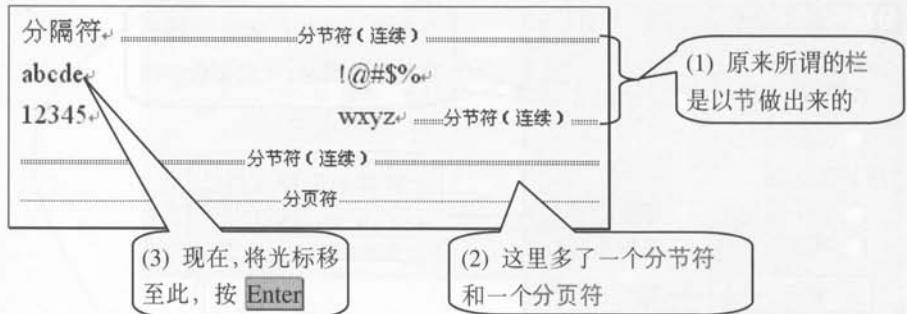
以下我把页面上的换行、分页、分栏、分节符号显示出来，让您更有视觉上的了解。假设我在文档中键入以下内容：



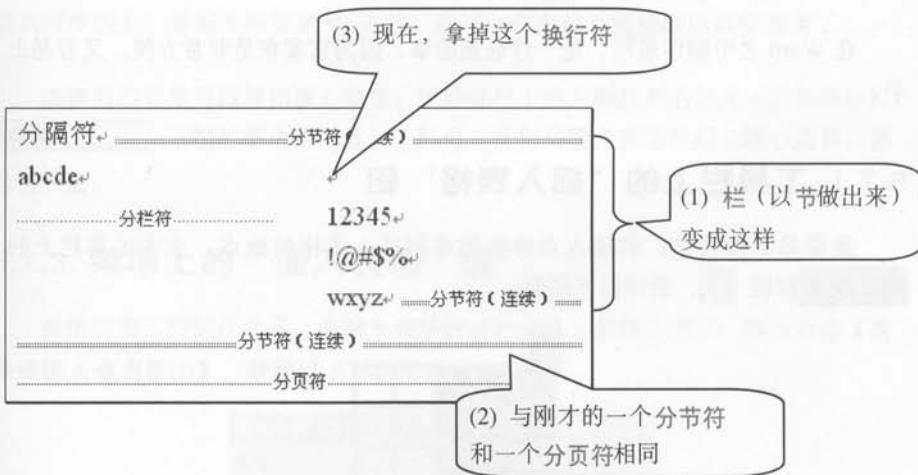
接下来将最后四行设为两栏格式（操作过程见 5.4 节）：



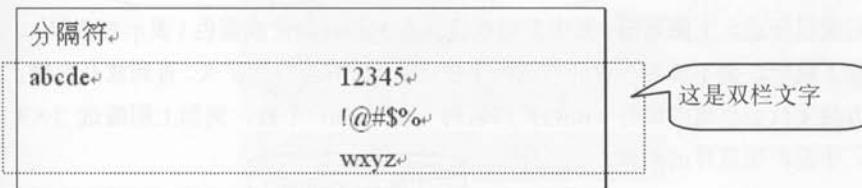
然后点击工具栏上 **显示/隐藏** 按钮  (双刀开关)，将所有分隔符显示出来：



于是画面变成：



再次点击工具栏上的 按钮，将所有分隔符隐藏起来：



当我们进行索引（通常双栏）页面的调整时，有可能需要“强制加上分栏符”。

5.6. 项目 (清单; 列表; list)

不知道中文版开发团队中的哪一位老兄把英文版【Format→Bullets and Numbering】译成【格式→项目符号和编号】。我觉得“项目”一词在这儿译得糟透了。不过，唤山不来向山走去，就习惯它吧。它的意思是指列表、清单 (list)。

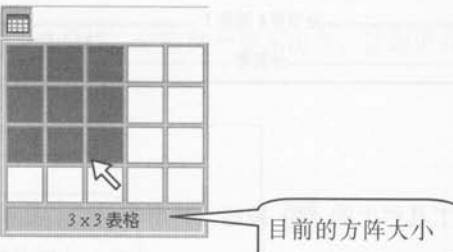
人类对于“第一条、第二条……”这种文字表达方式的接受程度，远高于对长篇大论的接受程度。因此使用条列来表现条理分明的文字，效果很好。这就需要用到项目符号 (●、■、➤...) 或项目编号 (1, 2, 3...)。这个主题已经在 p.124 的“样式格式之七”中有了完整的介绍。

5.7. 表格

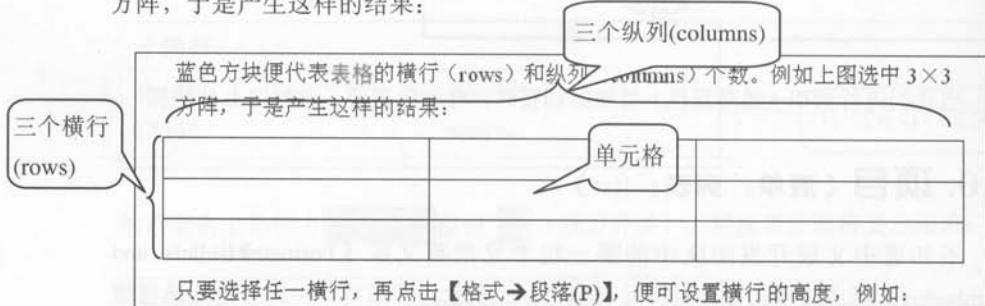
在 Word 之中制作表格，是一件惬意的事。因为它实在是非常方便，又容易上手。

5.7.1. 工具栏上的“插入表格”钮

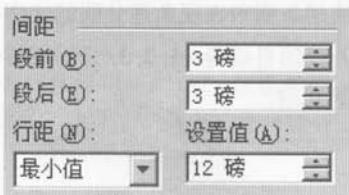
最简单的做法是：将输入点移到您希望插入表格的地点，点击工具栏上的 **插入表格** 按钮 ，获得以下画面：



只要鼠标进入上图范围，图中方格就会从左上角开始变成蓝色（表示被选取）。如果按下鼠标左键不放并一直向右或向下拉，这个方阵会持续扩大。直到放开右键，蓝色方块便代表表格的横行（rows）和纵列（columns）个数。例如上图圈选 3×3 方阵，于是产生这样的结果：



这些单元格以默认高度和均等宽度占据页面。只要选择任一横行，再点击 **【格式→段落(P)】**，利用图 5-5 所示的段落间距，便可设置横行的高度：

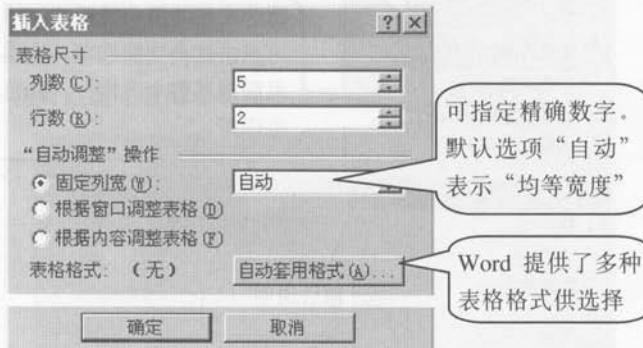


各纵栏的宽度也可以自由调整，并不是非得均等分不可。只要将鼠标光标移到纵列网格线上，看到光标变成 \leftrightarrow 形状，就可以自由拉动网格线以调整宽度了。

表格的总宽度可以超出版心宽度：拉动标尺上的左标尺和右标尺（我是指标尺下端的那个小四方块）即可。任何页面元素想要超越版心边界，都是这么做。

5.7.2. 菜单上的“插入表格”项

表格的第二种制作法是，将输入点移到您希望插入表格的地点，然后点击【表格→插入→表格(I)】，获得以下对话框：



上图设置列数 (Columns) 和行数 (Rows)，其效果和在工具栏按钮 \square 上的设置完全相同。惟一的差别是这里可以点击自动套用格式(A)...一并设置格式，如图 5-31。

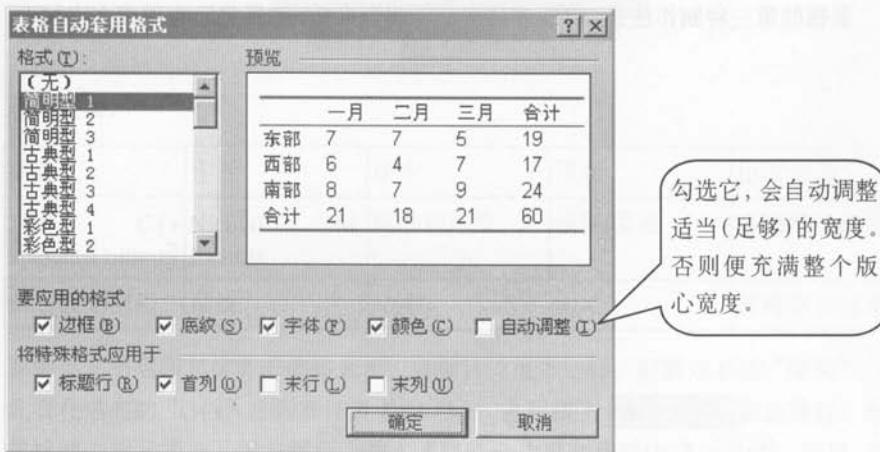


图5-31 表格自动设置 提供许多现成的表格形式

表格形成之后，随时可以再运用【表格】菜单（图 5-32）来调整列（Columns）和行（Rows）的数量，以及单元格的长度和宽度。这些功能的操作都十分直观，无需多加说明，实际操作一遍自然就会有很好的心得。其中较易被忽略的是跨页标题行重复，对于大篇幅跨页表格很有帮助——本书处处可见，例如表 1-2。

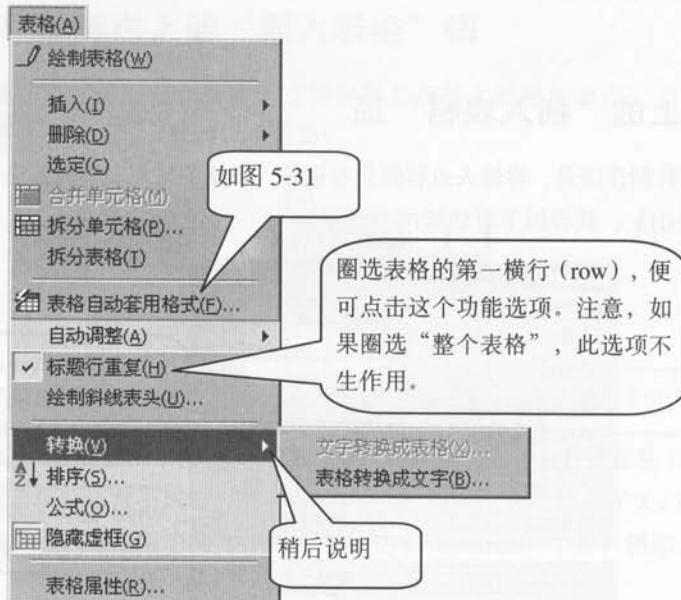


图5-32 【表格】菜单提供有许多表格调整功能

表格的第三种制作法是，将现成的文字转制为表格。如果您已有现成文字如下：

原著
作者
译本
译者
出版公司
The C++ Standard Library
Nicolai M. Josuttis
标准程序库
侯捷/孟岩
基峰图书公司
STL 源码剖析
侯捷
N/A
N/A
基峰图书公司

圈选所有这 15 行后，便可点击【表格→转换→文字转换成表格(V)】：

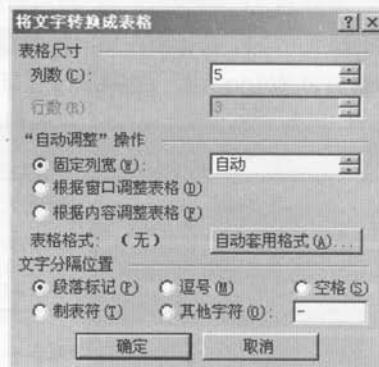
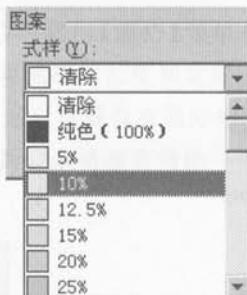


图5-33 将文字转换成表格

于是获得：

原著	作者	译本	译者	出版公司
The C++ Standard Library	Nicolai M. Josuttis	标准程序库	侯捷/孟岩	基峰图书公司
STL 源码剖析	侯捷	N/A	N/A	基峰图书公司

其中英文书名和其作者的名称太长，自动转折至下一行，但断句不甚“漂亮”，因此我在书名的“C++”之后和作者名的“M.”之后按下 Shift+Enter 强迫换行。为突显标题，我又圈选了第一横行，进入【格式→边框和底纹(B)】+【底纹】，选择 10% 灰阶：



这便获得了令人稍感满意的表格：

原著	作者	译本	译者	出版公司
The C++ Standard Library	Nicolai M. Josuttis	标准程序库	侯捷/孟岩	暮峰图书公司
STL 源码剖析	侯捷	N/A	N/A	暮峰图书公司

如果圈选整个表格并进入图 5-31，选择图左侧的**简明型 I** 格式（这是我最喜欢的清爽风格），并将第一横行设为 10% 灰阶（操作同上），便得以下表格：

原著	作者	译本	译者	出版公司
The C++ Standard Library	Nicolai M. Josuttis	标准程序库	侯捷/孟岩	暮峰图书公司
STL 源码剖析	侯捷	N/A	N/A	暮峰图书公司

相信您也看出来了，有些单元格内容因为“太满”的缘故而彼此过于靠近，这时可将鼠标移至网格线上（鼠标光标形状将有变化），个别拉动网格线调整之。

以上表格应付基本需求已是绰绰有余。如果需要更繁复的表格，例如：

英文词	中文词	重要页次	英文词	中文词	重要页次

或如表 1-2、表 10-1 等复杂表格，可多尝试【表格】菜单中的各个功能项。以下有更进一步的说明。

5.7.3. 非“单纯方阵”表格

表格内的单元格可以分割，形成“非单纯方阵”表格。做法是将输入点移至某单元格中（表示点击了它），然后点击【表格→拆分单元格(P)】：

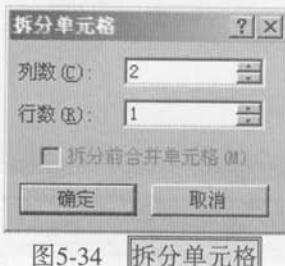


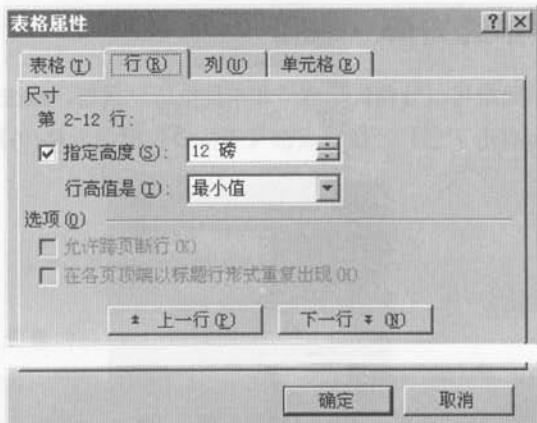
图5-34 拆分单元格

表 6-1 是个实际例子。该例将“后缀名”和“说明”两列的第 1 行分割为 11 行，如下图所示。此表格的行高（查其段落格式可知）为前距 3pt，后距 3pt，行高 12pt。由于第 1 行一开始预留的高度过大，所以均分 11 份后每一列的高度还是高于原本设计（下图灰色标题的高度）。图中黑色横条便是鼠标圈选后的高亮（highlight）区，这个高亮区只会显现横行的原始设计高度。我们可以看到，每一黑色横条之下还留有一些空隙（白色横条）。

种类	特征	后缀名	说明
点阵图 bitmap	以点（图素）形式存在，每个点（图素）存放一种颜色。所以有 1024 × 768 × 256（色）这样的表示法。相当占用储存空间。放大或缩小都会失真。放太大则会产生锯齿状。	bmp jpg/jpeg gif	
	电脑荧幕截图（hard-copy）就是一种点阵图。数码相机所得相片以及扫描器所得图片，也是一种点阵图。	tif/tiff pcx	
	点阵图可被储存为.bmp 或.jpg 或……，视软件能力而定。	png	
	许多时候 Word 把“点阵图”称为图片，把“向量图”称为图案。虽是原本意义相同的两个字眼，为避免混淆，不得不察。	pcd ico ipx ufo psd	
向量图 vector	以直线的位图或矢量图形式存在，如……	……	

将 1 行均分为 11 行。由于预留高度过大，导致均分后的行的高度还是过大。

欲消除这些空隙，一个做法是：圈选所有 11 行（就像上图那样），然后点击【表格→表格属性 (R)】+【行(R)】，设置数据如下（其中 12pt 是原本设计）：



这就获得了下面的表格：

种类	特征	后缀名	说明
点阵图 bitmap	以点(图素)形式存在,每个点(图素)存放一种颜色。所以有1024×768×256(色)这样的表示法。相当占用储存空间。放大或缩小都会失真。放太大则会产生锯齿状。	.bmp	
	电脑荧幕截图(hard-copy)就是一种点阵图。数码相机所得相片以及扫描器所得图片,也是一种点阵图。	.jpg/jpeg	
		.gif	
		.tif/tiff	
		.pcx	
		.png	
		.pcd	
		.ico	
		.fpx	
		.ufo	
		.psd	

向导图 以以前的值自动更新;只对图形和文本... F

所有11行都恢复了原始设计高度

但最后一行却留下了一些余绪……

至此,最后一行还是过高。这其实是因为左侧的那一大块单元格的高度高于右侧11行而造成的(本例那一大块单元格内的文字末尾有许多不可视的换行符号)。只要进入左侧一大块单元格最下缘,按Backspace除掉那些换行符号,就可以让整个表格“完美无瑕”了。日后在表格的单元格中进行的输入如果超出一行,文字会自动折行,单元格也将因此变得比较高,如表6-1。

5.7.4. 手绘表格

再没有比手绘表格更直观的表格制造法了。工具栏上的

--

按钮可以帮你做到这件事。既然那么直观而容易,在这本讨论排版的书籍中就不多说了,请自行实验。

5.8. 图片

图片的插入很简单，但是要对图片做到精准的定位，不太简单。请见第 6 章。

5.9. 图文框

请见 6.7 节。

5.10. 题注 (Captions)

题注是非常好用的东西。对排版或许没有帮助，对文档创作过程有绝对的帮助。

所谓题注，就是“图、表、或任何可被归类的东西”的编号。大型文档的创作过程中内容往往反复变动，前后挪移。时而有了新想法，又要增减一些图、表、或其他型式的页面元素。这些页面元素若由作者手动编号，不仅辛苦而且无趣，是名符其实的 dirty work。

Word 提供的题注正是用来解决这个问题的。只要在设想编号的地方（例如图表下方）插入一个题注，Word 便会为您维护其编号，并且自动化更新（但必须由您下达更新命令，后述）。这么一来创作者再也不必担心图表的增减，得以专心致志地把全副心神用来创作。

插入题注的操作非常简单：点击【插入→题注(C)】，在题注中选择一个标签，如图 5-35。所谓标签（label）就是出现在编号前的文字（图 5-36）。

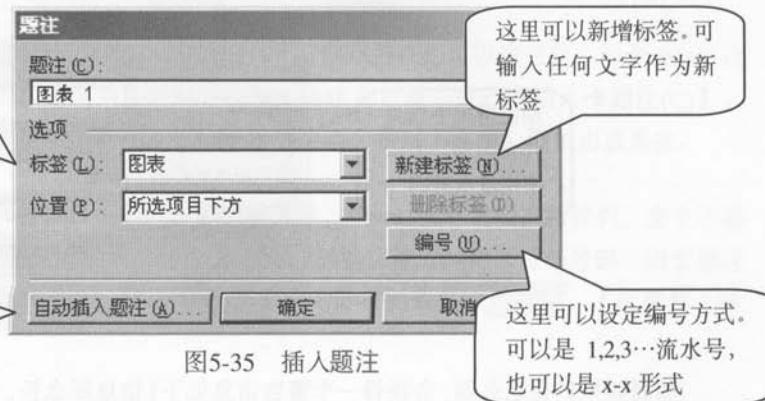


图5-35 插入题注

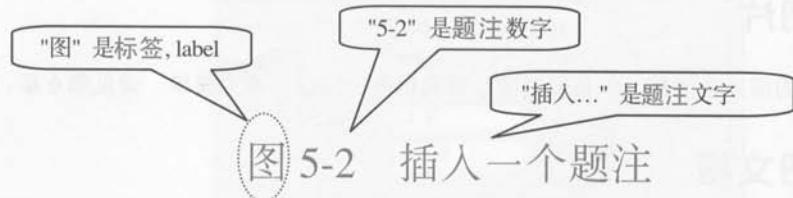
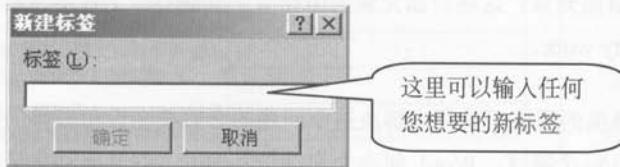
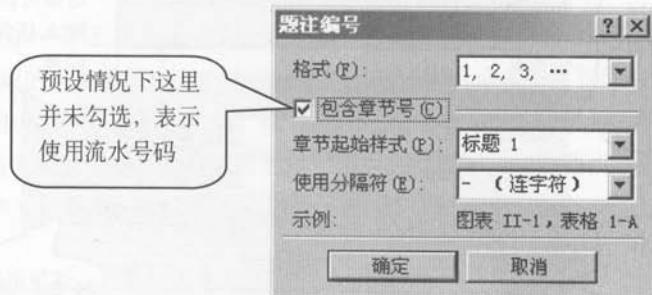


图5-36 题注标签、题注数字、题注文字

Word 内建三种标签 (labels)：(1)图表 (2) 表格 (3) 公式。如果这三种内建标签不能满足您，可以点击图 5-35 右侧的**新建标签(N)**获得以下画面，任意输入诸如“小秘诀”、“我的心得”之类标签。一般而言我喜欢把图称为“图”而不是“图表”，把表格称为“表”而不是“表格”，因此我两次使用下面这个对话框，加上“图”和“表”两种标签。



题注的编号方式可以是 1,2,3…流水号，也可以是 "x-x" 形式或其他。设置方式是：点击图 5-35 右下角的**编号(U)**，获得对话框如下，勾选其中的**包含章节号**，并在右侧选择一个章节起始样式和一个分隔符。这就表示您希望题注数字 "x-x" 的第一个 'x' 与该样式连动（第二个 'x' 则是流水号）。本书的原则是在各章之下编图号和表号，所以我让第一个 'x' 链接到**标题 1**（本书以**标题 1**为章名样式）。



然而如果只是这么做，会获得一个警告信息如下（信息那么长，为什么不折行？我不知道。这得由微软工程师回答）：

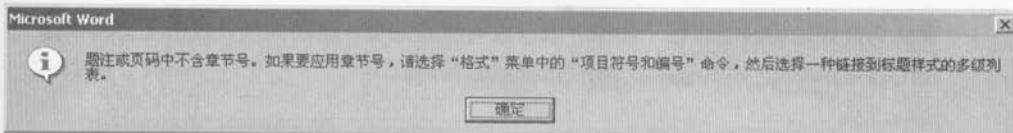
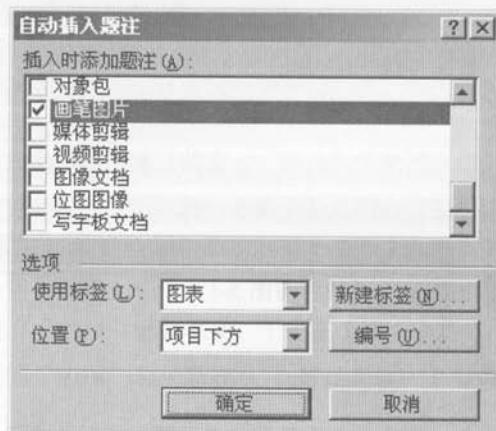


图5-37 Word 要求用户建立一个“链接至标题样式的多级列表”

并且，此后获得的各章题注都将会是 "0-x"。这是因为 Word 并未聪明到自动为我们计算章节编号。我们必须像上述警告信息所言，另外再做某种设置，详见 5.11 节。

除此之外，在书籍第 1 章之前，我们往往也把“序言、目录、前言、致谢”的地位等同于章，以**标题 1** 样式套用于这些标题名称身上。这么一来正牌的第 1 章出现时，其内的题注数字就不是 "1-x" 了，怎么办？这个问题也在 5.11 节解决。

请注意图 5-35 左下角的**自动插入题注(A)**，会产生以下对话框：

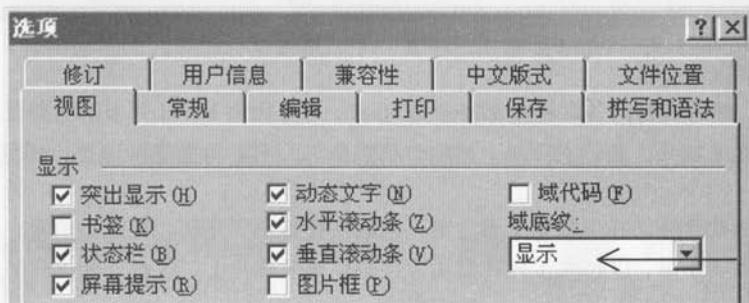


对此我有一个好消息和一个坏消息。好消息是：您可以透过它，命令 Word 自动为特定类型的页面元素加上题注，不必再由用户逐一进行【插入→题注(C)】。坏消息是：它似乎是个绣花枕头，中看不中用！我试了几次，没试出成果来。

所有题注都可以自由插队、附加或移除，Word 会对它们重新排列，完全不必创作者担心。但它不像脚注（5.3 节）那般能够保证立刻自动更新号码。由于题注是由域（第 9 章）构成，您必须在它身上点击右键菜单的**更新域**，才能更新（详 9.2.2 节）。

为避免创作过程中不必要的困惑（例如明明已是 “图 2-4” 却还标着 “图 2-3”），

我认为最好是将域明白标示出来，以求获得直观感受。这没问题，点击【工具→选项(O)】的视图，在其域底纹(E)栏选择显示即可：



这么一来域在屏幕上便都有个灰色底纹。虽然稍微有点丑丑的，实用还是重于美观啦。再说将来输出（不论打印机、输出机、或 PDF Distiller）时并不会输出这些灰色底纹。

题注样式

也许您不见得满意题注的“长相”。它的长相由题注样式控制。唔，解铃还需系铃人，这下我们知道该在哪儿动手脚了。我对于“题注与前后（上下）段落的距离”有意见，所以我进入【格式→样式(S)】并选择题注样式，然后点击更改(M)+格式(O)+段落(P) 加以修改（请参考图 5-1）。

图目录/表目录

严谨的学术论文，往往会有图目录、表目录，甚至公式目录。坊间有些书籍内含各式各样的小秘诀，也适合收集起来成为一个小秘诀目录，方便读者查找。

有了题注，这些图目录、表目录……各种目录唾手可得！请见 7.6 节。

修改题注标签

有时候我们会对先前所设置的题注标签（Caption Label）感到不满意。例如本书写作过程中，为了对第 10 章的 VBA 程序列表进行编号，我原先设了“VBA 程序代码”这个标签，后嫌冗长拟改为“VBA”。遗憾的是 Word 并未提供我们修改标签的机会。我只好尝试点击【插入→题注(C)】进入题注，在标签(L):处选择“VBA 程序代码”并点击删除标签(D)。此时原先已做好的各个题注（不论是其标签、编

号或文字) 不会有任何变化。然后我将原先存在的各个“VBA 程序代码”标签逐一改为“VBA”：

```
#024    MsgBox hdrftr.Range.Text
#025    'The Art of Word - Writing and Typesetting Large Scale Documents
#026    '(blank)
#027    'The Art of Word - Writing and Typesetting Large Scale Documents
#028 Next
```

直接在正文中将“VBA 程序代码” 改为 “VBA”

VBA 程序代码 10-7 剖析 Section 对象

此时在“VBA 程序目录”区点击右键菜单的【更新域(U)】，所有目录条目便都随之修改妥当：

VBA 程序目录

目录获得了修改

VBA10-1 宏, 将文档中的所有“软体”替换为“软件”.....	275
VBA10-2 Application 对象.....	284
VBA10-3 Document 对象	286
VBA10-4 Paragraph 对象	287
VBA10-5 剖析 paragraphs、sentences、words 和 characters	288
VBA10-6 观察 Section 对象和 PageSetup 对象	291
VBA10-7 剖析 Section 对象.....	292

但是接下来呢？我没有“VBA”标签可用，于是在【插入→题注(C)】的【题注】中加上它，但接下来新建的“VBA”相关题注从 1 开始起编，不复接续原先的“VBA 程序代码”编号：

```
#027    'The Art of Word - Writing and Typesetting Large Scale Documents
#028 Next
```

从 1 起编，不妙！

VBA10-7 剖析 Section 对象

VBA 10-1 剖析 Section 对象

我只好把所有的“VBA 程序代码”题注（其标签已被改为“VBA”）删除掉，全部换为“VBA”题注。

题注和标签的关系

上面所讨论的标签修改过程，透露出些许的吸引力，吸引我想知道题注和标签

和文字(图 5-36)究竟是什么关系。10.3.3 节所谈的 TableofFigures 对象(p.296)可带来额外的信息。

5.11. 多层次样式/自动化章节编号

对创作平台而言，这是很关键的一节。

创作平台的一个很大诉求是(1)各种编号自动化(2)交叉引用(5.12 节)自动更新。如您所见，本书用了许多交叉引用(就是“请参考第 8 章”或“请参考 5.2.2 节”或“请参考表 1-1”或“请参考 p.35”这些东西)，其中最多的引用对象就是章节。因此，如果在一份长文档中没有为其章节设立多层次样式/多级列表(本节主题)，那么章节的引用是别想玩了！

我们希望创作平台能够做到以下两点：

- 图表等页面元素的题注(captions)有着“分章编列”的智慧表现。换句话说表现出“图 x-x”或“表 x-x”，其中第一个‘x’是章序号，第二个‘x’是流水号。
- 交叉引用所指的章节如“第 8 章”或“第 5.2.2 节”，其编号由 Word 维护，由 Word 负责更新。

稍早图 5-37 曾经显示，欲令 Word 自动为我们做出“分章编列”的题注，Word 反要求我们先建立一个“链接至标题样式的多级列表”。那是什么东东？

是这样的，Word 尚未聪明到会主动为我们计算章节编号，因此如果您的章节(**标题**)全都靠自己手动编号，Word 一点儿过问机会都没有，导致所有的题注都是“0-x”(当然您也可以弃而改用流水编号 1,2,3……)。

但如果您向 Word 下达某种命令，让章节编号的权力掌握在 Word 手中，Word 就有能力在题注出现时抓取当时的章号，填入题注数字内，成为“分章编列”的题注。请参考图 5-38。

在整个多层次样式的设置过程中，我个人的经验有一点峰回路转、柳暗花明的感觉。原先我只能够做出第一层样式(代表“章”)并让 Word 有正常的表现。一旦做第二、第三层样式(代表“大标题”和“中标题”)，Word 的表现便有点涣散，非常伤脑筋。很幸运的情况下才发现是 Word 的臭虫做怪。回避那些难缠的小

鬼，阎王也就见到了。我把整个过程记录下来，其中或有值得省思或观摩的地方。

4. 页眉与页脚

4.1. 页眉/页脚的大小和位置

关于页眉和页脚的大小和位置，请见 3.3 节，该处与页面设置一起讨论。

4.2. 页眉/页脚的内容设置

看看您手上这本书，奇偶页的页眉和页脚有何不同？某些单元（例如序言、致谢、导读、附录、索引）的奇偶页页眉和页脚内容完全相同（只页码不同），某些单元（例如各章）就不是这样。以本书各章为例，偶数页眉的内侧放置章名（标题），那么 Word 便能在制作题注时抓出上述编号（代表第 4 章）放到标号的数字中。



图4-1 奇偶页页眉不同的情况

图5-38 题注与章节 标题 自动编号的关系

5.11.1. 初体验

欲令 Word 自动为我们设置章节编号，可首先点击【格式→项目符号和编号(N)】并选择其中的【多级符号】，如图 5-39，从中选择一个您喜欢的编号型式，然后按右下角的【自定义(T)】，导致图 5-40。

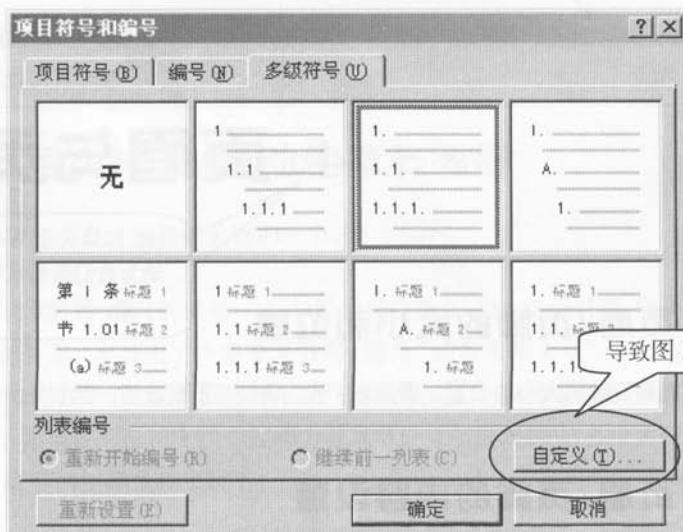


图5-39 选择一个多级符号型式，令Word做出多级章节编号

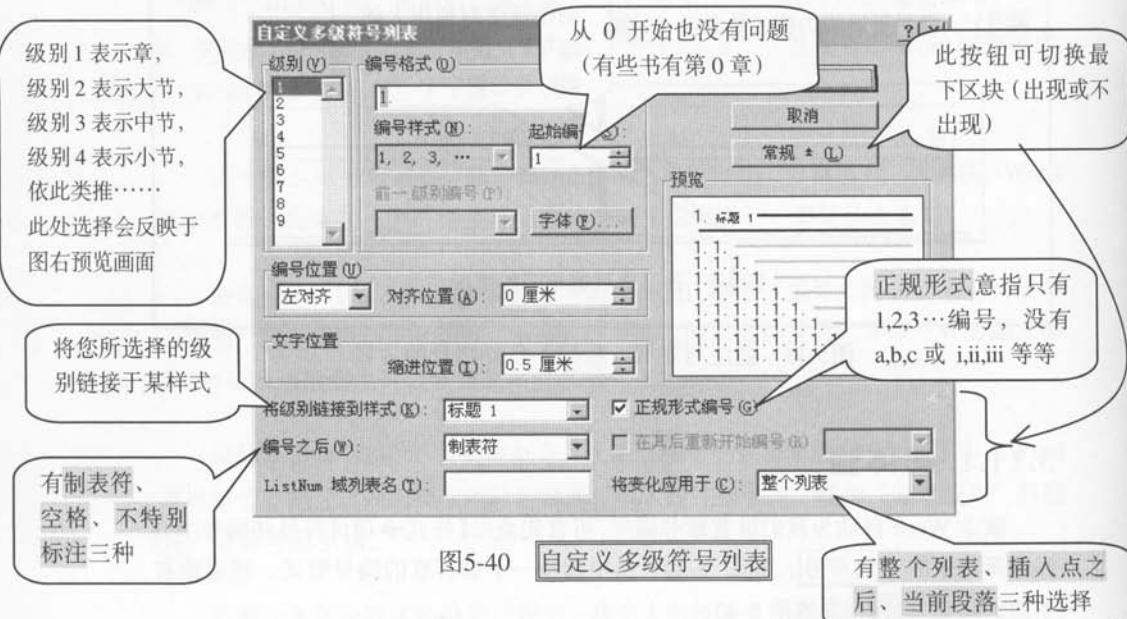
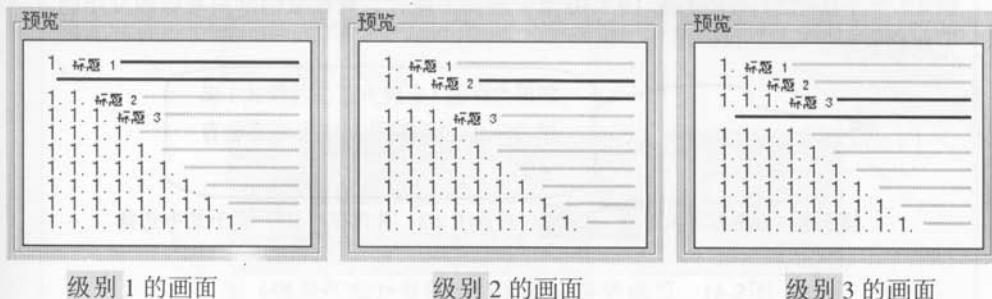


图 5-40 的 **自定义多级符号列表** 颇有玄机，乍见之下很难理出头绪。这个对话框用来让您告诉 Word：章节的第 n 层结构应该链接（对应）于哪一个样式。以本图（同

时也是本书实际情况)为例, 章是**标题 1**, 大节(x-x 节)是**标题 2**, 中节(x-x-x 节)是**标题 3**, 小节(x-x-x-x 节)是**标题 4**。然而我并不打算为小节列出编号, 因此我在图 5-40 中动作三次, 分别选择左上角的级别(V)1,2,3, 并为左下角的将级别链接到样式: 分别选择**标题 1**、**标题 2** 和**标题 3**。完成后, 对应的右侧预览画面分别是:



很好! 但还不是最好! 别忘了, 最好的办法是根本性、系统性地修改样式, 而不是无厘头似地到处沾惹。就本例而言, 既然整本书的**标题 1**、**标题 2** 和**标题 3** 都要做这样的设置, 那就不要看到一个修改一个, 最好的做法是从根本的样式下手。

首先修改**标题 1**: 点击【格式→样式(S)】+更改(M)+格式(O)▼+编号(N), 直接进入图 5-39, 然后按照图 5-40 的说明, 令“层次一”链接**标题 1**。再以相同的程序分别修改**标题 2** 和**标题 3**, 令“层次二”链接**标题 2**、“层次三”链接**标题 3**。我并且令三个层次的文字缩进位置分别为 0.5, 1.0, 1.5, 编号之后的字符皆为“空格”, 并选择**将变化应用于: “整个列表”**。

5.11.2. 尝苦果

至此, 我们有了多层次样式。Word 会自动化地为我们设置章序号和大标题、中标题序号。然而当我依序完成这三层设置时, 发现事情变得有点混乱!

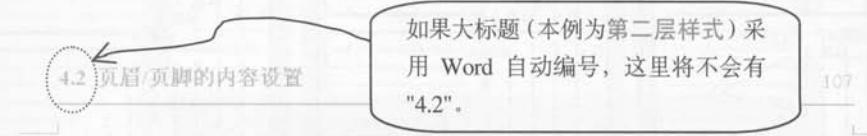
首先是标题号码虽然在页面视图中被 Word 自动做出来了, 却无法稳定地被标示于结构图(图 2-1)中, 总有一些缺漏, 这会严重妨碍结构图对创作的帮助。

其次我发现, 像下面这样一个标题:

4.2 页眉与页脚的内容设置



虽然整个（包括数字和文字）会被 Word 抽取成为目录（很好），但是当我按照 4.2 节手法把 StyleRef 域（9.3.10 节）加到页眉时，被抽取的却只有标题文字而无标题数字：



接下来，点击右下角的 **选项(O)...**，获得图 4-3，并选择**样式**，再于其中选择

图 5-41 页眉没有出现自动化编号（这不理想）

这可就糟透了！读者应该不会希望页眉只出现标题文字而没有标题编号。

更糟糕的是，每次跨越“章”（**标题 1**，第一层样式），**标题 2** 和 **标题 3**（第二层样式和第三层样式）的编号仍沿续上一章编下来（例如，第 1 章编至 1.5 节，第 2 章竟从 2.6 节开始编号）。这个情况经过各种调整，有时候 OK，有时候一团乱。混乱的程度使我无法总结心得。

为此，我终于决定放弃第二、第三层样式，改以万能双手代替（又回到了石器时代⑧）。但至少留下了表现稳定的第一层样式，让 Word 显功夫。这已经带来很大好处：我可以让图表的题注形式成为“图 x-x”或“表 x-x”，这对大型（尤其是书籍般庞大的）文档而言，读者查找图表时远比面对单纯的 1,2,3... 编号好得多！

但是放弃第二、第三层样式，意味牺牲了不少交叉引用的弹性 ⑧

5.11.3. 山不转路转

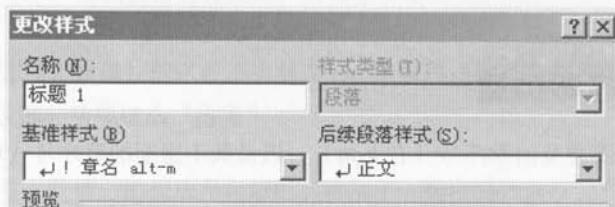
基于上述因素，本书原本打算只使用第一层样式。问题又来了：此后 Word 将所有**标题 1** 加以编号，而由于我对“序言、目录、致谢”三个单元的标题也套用**标题 1** 样式，导致第 1 章出现时，标题编号已经是 4！

为解决这个问题，可将输入点分别移至上述三个单元，透过【格式→项目符号】

和编号(N)】+【多级符号】(图 5-39)，将其多级符号全部改为“无”。现在，第 1 章的编号被正确修正为 1 了。



解决上述(从“序言”开始编章号)问题的另一种做法是：加一个中介层。这是计算机科学的惯用伎俩，想不到这儿也用得上。拿我个人的实际经验来说，在我尚未完全学得 Word 正规战之前，都是使用自定样式，并未使用 Word 的内建样式。因此我有一个!章名样式。学会正规战之后，我并不想一一重设标题 1 的众多格式(图 5-2)，所以令标题 1 继承!章名：



无心插柳地有了这个中间层，恰好在这儿派上用场。我把文前的每个单元(序言、目录、致谢)标题全部改用!章名，直到第 1 章才令各章名使用标题 1。于是 Word 的第一层样式的自动编号就正确了。

但是这又引发一个问题：我们已经知道目录和结构图都是靠抓取标题 1 来完成，如今由于“序言、目录、致谢”都不再使用标题 1，导致这三个单元名称不出现于目录和结构图中。“未出现于目录”算是大错误，“未出现于结构图”则会对创作过程带来小麻烦。

上述第一个问题有解。Word 该抓哪些东西来做成目录，其实可以由用户决定(7.1 节)，因此我点击【插入→索引和目录(D)】+【目录(C)】，于其中点击【选项(O)】，而后为【目录级别(L)】加上!章名并令其级别为 1(与标题 1 的层次同)。此后 Word 便能够同时也抓取!章名做成目录。

至于上述第二个关于“结构图内容残缺”的问题，没解！毕竟 Word 没能让用户决定结构图抓取哪些样式。这在创作过程中实在带来很大不便(在我们曾经那么方便过之后)，所以最终我并没有采取本框所说的这种做法。

不过，事后发现，所谓“结构图内容残缺”其实是 Word 的小小臭虫。稍后详述。

有了（至少）一个可正常运作的第一层样式，真是太好了，Word 在制作题注以及进行交叉引用时便都可以抓取该编号，甚至抓取该编号所在页码。这么一来作者写“请参考第 x 页，第 x 章”时就轻松多了。

Word 在这方面有两个缺失值得检讨。首先是一如前面所提，页眉所抽取的只有标题文字而没有标题编号，这当然不够好（但另有办法可解，见 9.4 节）。其次，标题编号会被抽取放进目录之中，这虽然很好，但如果编号“一定出现于章名之前”却又不是很好的消息，因为章名与章序号之间往往需要更大的设计弹性，见下一节讨论。

5.11.4. 忍辱负重

下图是我想要的章序号（右上角），以及我不想要的章序号（章名之前）。

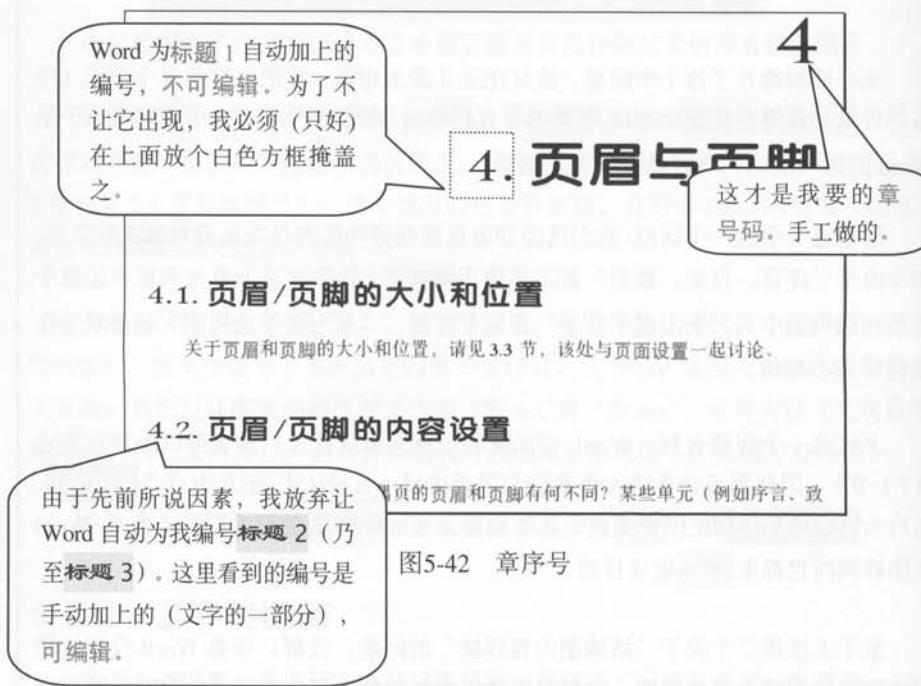


图5-42 章序号

实在说，这是难为了 Word：章号码出现在目录上，喝彩；章号码出现在正文之中，喝倒彩！天下哪有这种事，我们也太苛刻了。

但是天下没有不是的顾客，Word 该做的就是：提供一个弹性，让用户能够有分开处理标题编号和标题文字的机会。如果让用户有机会：(1) 决定标题编号要不

要出现，(2) 设置标题编号和标题文字之间的符号（例如分隔符，那就表示可以换行），(3) 设置标题编号和标题文字各自不同的格式（字体、大小、颜色、行距等等），那么目前所说的这个问题就可以圆满解决②。

好吧，Word 没有这么神武英明，所以情况依旧。我该怎么办呢？“剪一张白纸把它贴上去遮住那个不想要的编号”！不是开玩笑，我说真的！利用绘图工具栏上的□按钮画一个白色矩形，调整适当大小后拖拉盖掉我不想要的那个数字（也就是上页图中的那个虚线框）。白色贴纸不会有任何痕迹被输出，被遮掩的数字也不会被输出。所以呢，虽然过程有点麻烦，但效果完美！

不，不够完美。被遮住的东西不会被 Word 抽取到目录去（那当然不好）。真没想到 Word 这部分功能这么“厉害”。

进行到这里，我惟一的想法是：好吧，投降了。我将同时保留上页的两个章序号，制版时再来修理它（手段似乎更肮脏了③）。

5.11.5. 柳暗花明

一试再试做不成，再试一下。在锲而不舍的不断尝试中，我终于发现，先前观察到第二层、第三层样式所造成的(1)结构图失误，以及(2)编号次序不正常，其实都是Word臭虫所造成。

首先回顾目前状态。我已透过【格式→样式】将**标题1**设为第一层样式，动作如图5-40。**标题2**和**标题3**不做相同设置。我也已经将“序言、目录、致谢”三个单元以及“附录、索引”两个单元的标题（也都是**标题1**）透过【格式→项目符号及编号(N)】+**多级符号**（图5-39），将其多级符号全部改为“无”。透过【格式→样式】观察此时的各个样式说明，可看到：

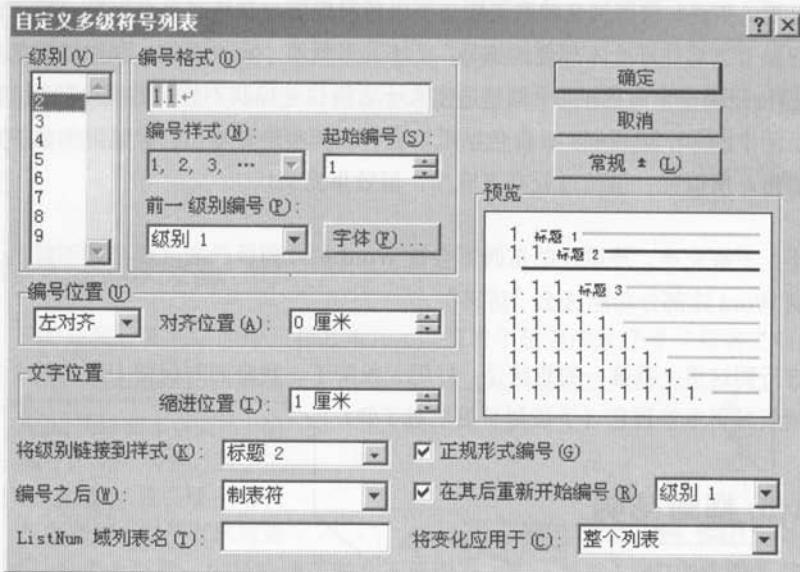
- **标题1**: 1 级, 多级符号.....
- **标题2**: 2 级.....
- **标题3**: 3 级.....
- **标题4**: 4 级.....
- **标题5**: 5 级.....

只有透过图5-40设定
第x层样式，才会有此属性。

此时“序言、目录、致谢”都没有编号，1~12章有编号，12章之后的“附录

和索引”都没有编号。

接下来进行我们的新手术：根据图 5-40，将**标题 2** 和 **标题 3** 分别设为第二层样式和第三层样式：



再度观察样式说明：

- **标题 1**: 1 级, 多级符号.....
- **标题 2**: 2 级, 多级符号.....
- **标题 3**: 3 级, 多级符号.....

只有透过图 5-40 设定
第 x 层样式，才会有此属性。

从样式下手，造成的影响是全体性的，但我并不希望“序言、目录、致谢”三个单元（甚至包括第 0 章“导读”）的**标题 2** 和 **标题 3** 有所编号。因此我一一到达这些地点，透过【格式→项目符号和编号(N)】+**多级符号**（图 5-39）将其多级符号改为“无”。整份文档这么修改的地点很多，可利用 （复制格式）完成。

进行至此，老问题出现了：（下面是问题和解法）

- 有些编号的格式（字体）不对。
- 我发现，如果（例如）**标题 2** 的一部分标题文字被我改变字体大小，那么当我在标题文字之后按下 **Enter** 产生分页符号，就会造成标题的编号格式失误：

5.1. 样式 (styles)

无样式不成排版。样式是什么？让我们看看【格式→样式(S)】所得画面：



5.1. 样式 (styles)

在这里按 Enter
造成的影响

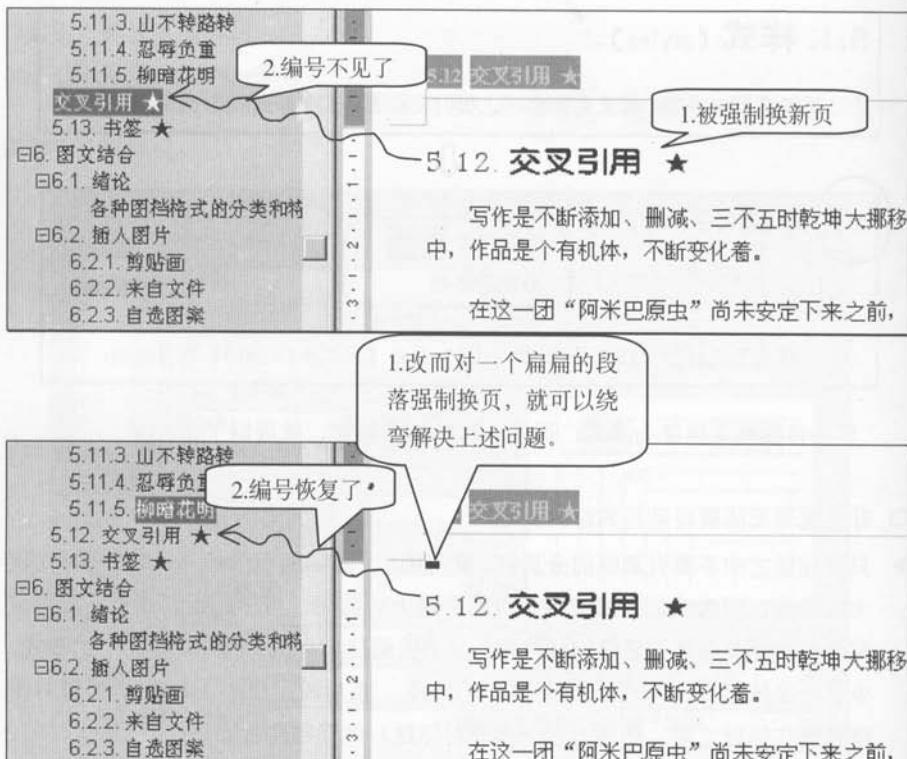
无样式不成排版。样式是什么？让我们看看【格式→样式(S)】所得画面：

只要将标题 2 恢复为正文，再设一次标题 2 样式，就可以了。

- 有些编号无法被自动抓到结构图中。
- 只要标题之前不要有紧临的分页符，就解决了。这不能不说 Word 的一只比较大的臭虫，因为我们很可能为了页面美观上的需要，将标题强制换页，因此常常会在标题前面加上紧临的分页符。为了回避这只臭虫，我特别设计一个样式，令它的字体很小 (4 pt)、行高很小 (4 pt)、前后段距很小 (0 pt)，必要时便将它放在标题之前，然后对它（而非对标题）做强制换页动作。

The screenshot shows the Microsoft Word ribbon at the top. Below it, the 'Styles' dialog box is open, showing a list of styles: 5.11.3. 山不转路转, 5.11.4. 忍辱负重, 5.11.5. 柳暗花明, 5.12. 交叉引用 ★, 5.13. 书签 ★, 由6.1. 缩论, 由6.2. 插入图片, 6.2.1. 剪贴画, 6.2.2. 来自文件, 6.2.3. 自选图案, 6.2.4. 艺术字, 6.2.5. 图表, 由6.3. 插入对象, 连结 (Linking) 对象, 6.4. 图片与对象的剪裁. A callout bubble points to the '5.12. 交叉引用 ★' style entry.

The main document area displays a hierarchical structure. The first level includes sections like '写作是' (Writing is), '中, 作' (In the middle, writing), '2. 如果我在这' (2. If I do this), and '里插入分页符' (Insert a page break here). A callout bubble points to the '1. 很好' (Very good) entry under the second level. At the bottom of the document, there is a section titled '5.12 交互参照 ★' (Interactive Reference) with the text: '在这一团“阿米巴原虫”尚未安定下来之前,' (Before this amoeba-like group has settled down,).



- 所有编号都不会被 Word 抓到页眉上
 - 只要在页眉添加一个域，并选用特殊选项 /r 即可解决。详见 9.4 节。

- 有时候编号会莫名其妙乱掉
 - 不要急也不要慌，这只是 Word 积劳成疾。通常只要【编辑→全选】，再在任何一个域身上点击右键菜单的更新域，Word 就会回神了☺

至此，功德圆满。惟一的遗憾是：我还是必须仰赖肮脏招数，在制版过程中请输出中心将那个我不喜欢的章序号（图 5-42）拿掉。

5.12. 交叉引用 (Cross References)

写作是不断添加、删减、三不五时乾坤大挪移……的过程。写作过程中，作品是个有机体，不断变化着。

在这一团“阿米巴原虫”尚未安定下来之前，章节和图表的编号随时会有变化。以人工编号这些随时可能变化的东西，实在太辛苦了，真正是 dirty work。前两节所谈的题注和多层次样式正是用来解决编号问题的。

但即使是自动化编号，它毕竟还是个变化体（只是其变化不再需要我们操心），当书籍某处需要写“请参考第 25 页”或“请参考 1.3 节”或“请参考图 4-2”时，怎么办？引用对象还没有稳定下来呢！

过去我的做法是，把上述那些尚未稳定下来的被引用对象写成“第 x 页”、“第 x.x 节”、“图 x-x”、“表 x-x”，再以一个黄色底纹提醒自己（黄色最是醒目了）。全书定稿后再以万能双手把全书走一遍，为所有未定的 x 填上正确数字。但这实在是创作者极大的负担：一方面是无聊且无趣，一方面尚未定案的引用（reference）也会对创作时的思路与灵感带来不利影响。再一方面，有时候定稿时还真想不起来当初的“请参考 x-x 节”到底是第几节。

Word 对此早已有了解决方案：交叉引用。这个宝贝可以（并且必须）配合所有由 Word 编号的东西：多层次样式编号、图表编号（亦即题注）、页码、脚注、尾注、书签。

举例说明，假设我要引用的对象是“表”（Word 有个内建的题注标签是“表格”，可我不喜欢，所以自定义一个“表”标签；做法见 5.10 节）。我可以在任何我希望的地点（引用点）写上诸如“请参考 I”文字，I 表示光标（输入点），然后点击【插入→交叉引用(R)】，获得图 5-43，并在引用类型上选择“表”，这时候 Word 便会将目前文档中所有“题注标签为“表””的东西陈列于下侧大方框中。选择其中之一，而后在右上角引用内容(R):上选择某种型式（所有型式在稍后的表 5-2 中有详细介绍）。从此，只要被引用物有了变化，引用点会同时更动；但有时需要您在其上点击【右键菜单】的更新域，才能看见最新编号。

表 5-2 的“项目编号”和“标题”，两者差别在于：前者涵盖多级符号（本书设置三层）和列表（list，也就是以 1,2,3 区分的条列），后者涵盖所有标题（**标题 1~标题 9**），不论有无设置多级符号。

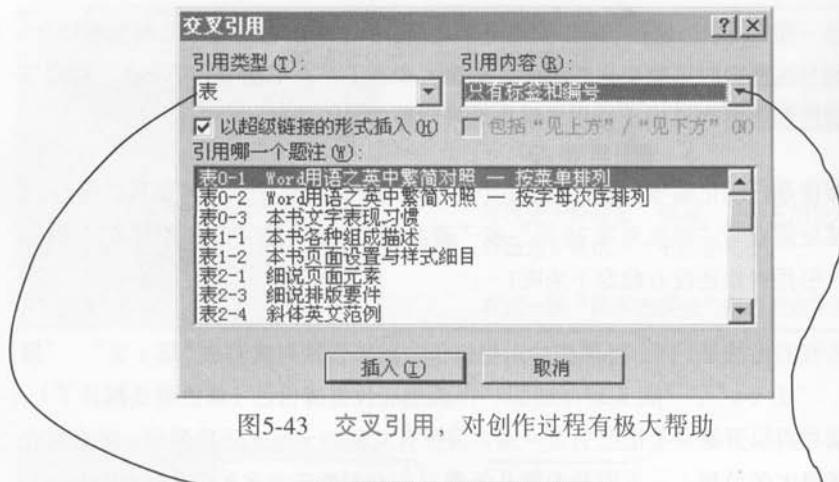


图5-43 交叉引用，对创作过程有极大帮助

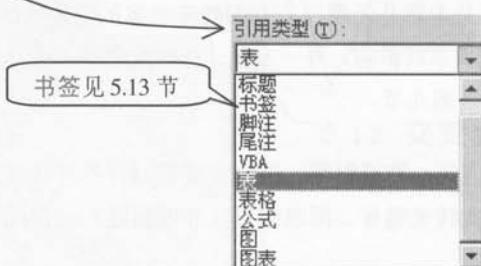


图5-44 用户可选择的交叉引用类型（内建或自定义均可）

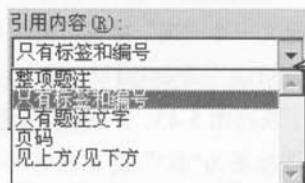


图5-45 可选择插入题注所含的信息及其所在页码

下面（表 5-2）是交叉引用（图 5-43）的所有可能选项。其中第三栏有着实际产生结果，此栏之编号项目以 6.2.1 节为例，标题亦以 6.2.1 节为例，书签以 StyleNamingConcern 为例，脚注以第 2 章脚注 1 为例，尾注以第 2 章的尾注 a 为例，VBA 以 VBA10-22 为例，图以图 1-1 为例，表以表 5-2 为例。

交叉引用在 Word 电子文档中是个超链接（hyperlink），鼠标在上面点击一下便能直接跳跃至目的地，非常有助于阅读的顺畅。不过跳回原出发点就困难了⑧

表5-2 交叉引用中的各种选项

引用类型(T)	加入引用类型的(R):	实际产生结果(例)
项目编号	页码	182
	段落号码	6.2.1
	段落号码 (部分显示)	6.2.1
	段落号码 (全部显示)	6.2.1
	段落文字	剪贴画
	见上方/见下方	见下方
标题	标题文字	剪贴画
	页码	182
	标题号码	6.2.1
	标题号码 (部分显示)	6.2.1
	标题号码 (全部显示)	6.2.1
	见上方/见下方	见下方
书签	书签文字	样式命名考虑
	页码	87
	段落号码	0
	段落号码 (部分显示)	0
	段落号码 (全部显示)	0
	见上方/见下方	见上方
脚注	脚注编号	1
	页码	59
	见上方/见下方	见上方
	脚注编号 (格式化)	1
尾注	尾注号码	a
	页码	93
	见上方/见下方	见上方

引用类型(T)	加入引用类型的(R):	实际产生结果(例)
	尾注号码 (格式化)	a
VBA (自定义标签)	整个标题	VBA10-22
	仅标签及数字	VBA10-22
	仅题注文字	找出字体发生地点，并跳至该处
	页码	328
	见上方/见下方	见下方
表 (自定义标签)	整个标题	表 5-2 交叉引用中的各种选项
	仅标签及数字	表 5-2
	仅标签文字	交叉引用中的各种选项
	页码	176
	见上方/见下方	见下方
图 (自定义标签)	整个标题	图 1-1
	仅标签及数字	图 1-1
	仅标签文字	出版流程
	页码	24
	见上方/见下方	见上方

5.13. 书签 (Bookmarks)

真实世界中所谓的书签是指您可以在书中任意点安插的一个实体标记，方便将来快速翻阅。Word 文档中的书签则是一个虚拟标记，有助于将来进行交叉引用，是引用动作的一个“标的物”。

您可以将输入点移至您想要的任何地点，然后点击【插入→书签 (K)】，获得对话框，插入新书签（自由命名）。也可以移除旧书签。详请参考 11.3 节。

6. 图文结合	179
6.1. 绪论	179
6.2. 插入图片	181
6.2.1. 剪贴画	182
6.2.2. 来自文件	184
6.2.3. 自选图形与几何绘图	184
6.2.4. 艺术字	185
6.2.5. 图表	186
6.3. 插入对象	187
6.4. 图片与对象的剪裁	188
6.5. 图片与对象的精准定位	190
6.6. 图片与对象的大小	197
6.7. 插入图文框	197
6.8. 文绕图	200
6.9. 覆盖于文档上的图片/对象	200
6.10. 图片对文档体积的影响——效率总检讨	201

6

图文结合

缺乏图片的文档，何其枯燥！本章介绍在文档中插入图片（各式图形文件）、对象（诸如 Powerpoint 简报、Excel 图表、WAV 声音、AVI 影片……）的做法。

计算机科学领域（更明确地说是编程, programming 领域）有一支称为“面向对象”（Object Oriented）。所谓“对象”，俗名“东西”。Word 将其他软件制作出来的东西统统称为“对象”。图片当然也是对象的一种，但因为 Word 对图片有较特殊的操作方式，所以我们把图片独立出来说明。6.3 节详谈对象。

6.1. 绪论

插入图片或对象的途径有两个：【插入→图片(P)】以及【插入→对象(O)】，以下分别说明。然而在介绍 Word 的相关操作之前，我先介绍各种图片的单元格式及其特性。

下面是整章的总轮廓：

- 对象种类：图片、图案、图表、造形文字、声音、影片
- 对象插入方式：链接（link）、嵌入（embed）、复制/粘贴（copy/paste）
- 对象摆放：类字符、浮贴

表 6-1 是各种图形文件格式的分类和特性。

本章主题之屏幕截图，于 Word 新旧版本中变动较大。为尽可能满足所有 Word 用户，本章采用 Word97 屏幕截图。对应之 Word2000+相关屏幕截图可于菜单或对话框中找到，敬请读者自行查找及练习。

各种图档格式的分类和特性

表6-1 图形文件格式与特性说明

种类	特征	后缀名	说明
点阵图 bitmap	以点（图素）形式存在，每个点（图素）存放一种颜色。所以有 $1024 \times 768 \times 256$ （色）这样的表示法。相当占用储存空间。放大或缩小都会失真。放太大则会产生锯齿状。	.bmp	Windows 通用图形文件。无压缩 ¹ ，支持的颜色有 16 色 ² 、256 色、1 677 万色。
	电脑屏幕截图（hard-copy）就是一种点阵图。数码相机所得相片以及扫描仪所得图片，也是一种点阵图。	.jpg/.jpeg	美国摄影协会制定。使用失真但肉眼可接受的压缩法 ³ ，压缩率最高。
	点阵图可被储存为 .bmp 或 .jpg 或……，视软件能力而定。	.gif	亦是一种压缩文件，最多 256 色。支持透明背景。采用 LZW 压缩法。
	许多时候 Word 把“点阵图”称为图片，把“矢量图”称为图案 剪贴画。虽是原本意义相同的两个字眼，为避免混淆，不得不察。	.tif/.tiff	可在 PC 和 Mac 之间轻易进行交换。
		.pcx	早期知名软件 Paint Brush 的图形文件格式。历史悠久，获得许多软件的支持。
		.png	Portable Network Graphics，压缩率比 GIF 更好，并获得 W3C 的推荐。
		.pcd	Kodak Photo CD 格式。主要用以储存 1 677 万色相片。无压缩。
		.ico	Windows 图标格式，通常画面分辨率是 16×16 或 32×32 。无压缩。

¹ 但如果将.bmp 插入 Word 文档内，保存文档时 Word 会对它进行压缩。

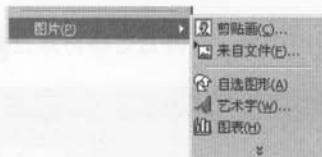
² 本书所有屏幕截图（hard-copy）都采用 16 色 .bmp 图片文件，仅有极少数为了表现 16 色以外的颜色而例外。

³ JPG 利用“肉眼对高频较不敏感”的特性，将图片高频部分以较粗颗粒的像素表示，将低频部分以较细颗粒的像素表示。所以它的压缩是一种频率概念而非空间概念。

种类	特征	后缀名	说明
		.fpx	Kodak FlashPix 格式，内含摄影重要信息。压缩率不逊于 JPG。
		.ufo	Ulead PhotoImpact 图档格式
		.psd	Adobe PhotoShop 图档格式
矢量图 vector	以必要的信息或轨迹记录图形。放大或缩小都不会失真。 Word 内建的自选图形、绘图工具栏提供的几何绘图工具所绘图形、剪贴画，都是矢量图。 CorelDraw 常附赠大量矢量图形。 Corel Mega Gallery 提供超过 50,000-200,000 个矢量图形（视版本而定）。 许多时候 Word 把“点阵图”称为图片，把“矢量图”称为图案。虽是原本意义相同的两个字眼，为避免混淆，不得不察。	.wmf	Windows Meta File，MS Windows 环境下的一种中继文件，只记录必要的图形信息或轨迹。
		.emf	增强型 .wmf 图档
		.eps	Adobe Postscript 格式，主要用于印刷。
		.cdr	Corel Draw 矢量绘图图形文件格式
		.dxf	AutoCAD 所使用的二维矢量图形文件格式 ⁴

6.2. 插入图片

将输入点移到插入目的地，点击【插入→图片(P)】：



共有五种选项如下，稍后一一说明：

1. 剪贴画：泛指 Microsoft 提供的“剪辑库”内的各种对象。
2. 来自文件：泛指外来的各种格式 (bmp, jpg, emf, wmf, gif, tiff...) 图形文件。
3. 自选图形：Word 提供的现成矢量图（箭头、流程、几何形状等等）。
4. 艺术字：Word 提供的文字艺术造形。
5. 图表：Word 提供的简易电子表格

⁴ DXF 曾经因为 AutoCAD 的红火而名噪一时。当时市场存在许多 AutoCAD 第三方协力厂 (third party)，他们首先就得知道如何解译 DXF。我也曾经解译过 DXF。

6.2.1. 剪贴画

点击【插入→图片(P)→剪贴画】可获得如图 6-1 的“剪辑库”。左侧是分类，中间呈现可供选用的图案（或其他种类的美工作品）。这些美工作品并非 Microsoft Office 的默认安装选项，所以极有可能您还必须同时放入 Microsoft Office CD 才能读取它们。点击其中任何一个图案，再点击右侧的【插入(I)】，即可将该图案插入文档内。所有图案都是矢量图，占用的空间很少。

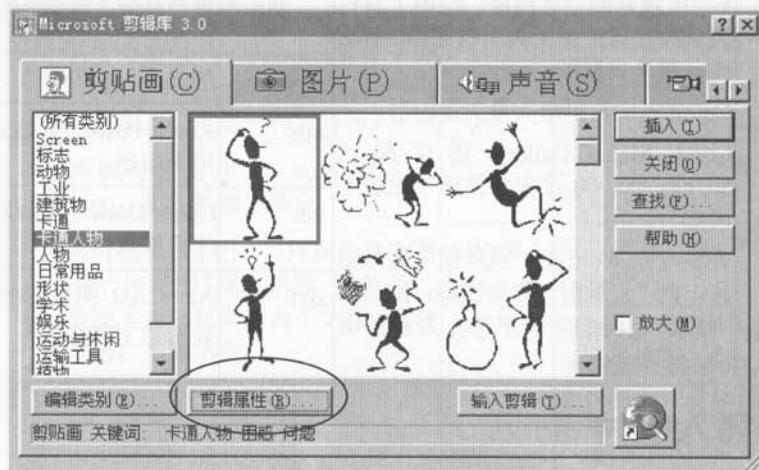


图6-1 Microsoft 剪辑库

以预览方框左上角那个卡通图为为例，点击它之后再点击【剪辑属性(R)】，获得：

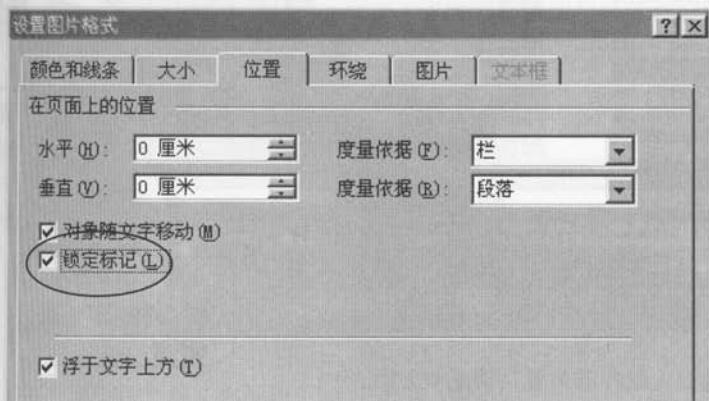


图6-2 多媒体项目（剪贴画、图片、声音、影片）属性

这个 AMCONFUS.WMF 不过区区 1494 bytes 而已。

除了剪贴画(C)之外，剪辑库另提供图片(P)、声音(S)、影片(V)，也都可以相同手法插入文档内。剪贴画(C)和图片(P)的差别在于，前者(Clipart)是矢量图，后者(Picture)是点阵图(并且几乎都是 JPG 档)。声音(S)和影片(V)在纸面文档中无法展示，所以不是基于印刷用途，而是基于简报或在线文档之用。

这些对象贴入文档的位置都是浮动的。为了固定住它和文档之间的相对位置，我必须点击此对象【右键菜单】的【设置对象格式】+【位置】，然后勾选“锁定标记”。



6.2.2. 来自文件

下面是【插入→图片→来自文件(F)】所得的对话框。中央有一块预览区。

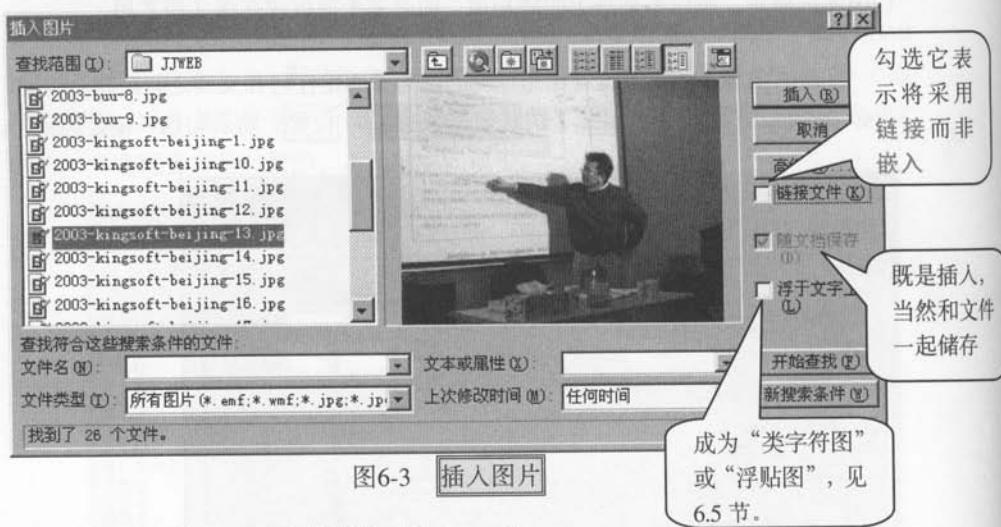
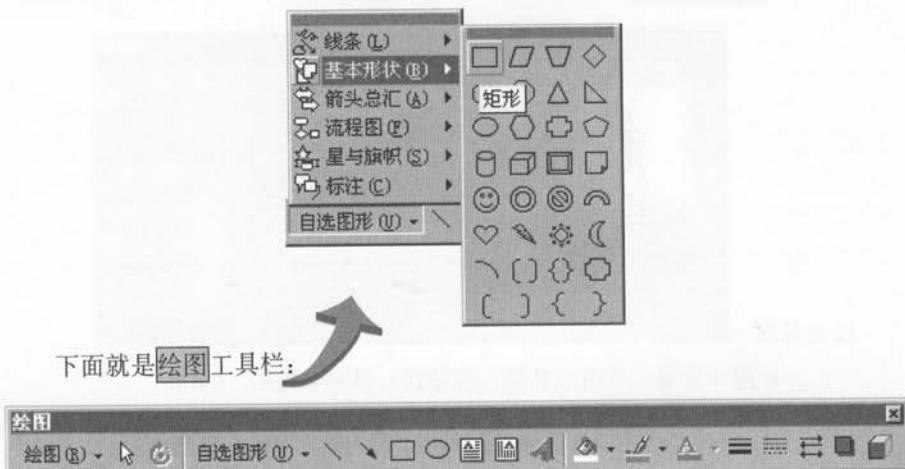


图6-3 插入图片

欲精准控制被插入图片的位置,请见 6.5 节。

6.2.3. 自选图形与几何绘图

Word 的 **绘图** 工具栏有一个**自选图形(U)**下拉菜单, 其中提供许多现成可用的图案。只要点击任何一个图案, 便可于文档内以鼠标拉出该图案:





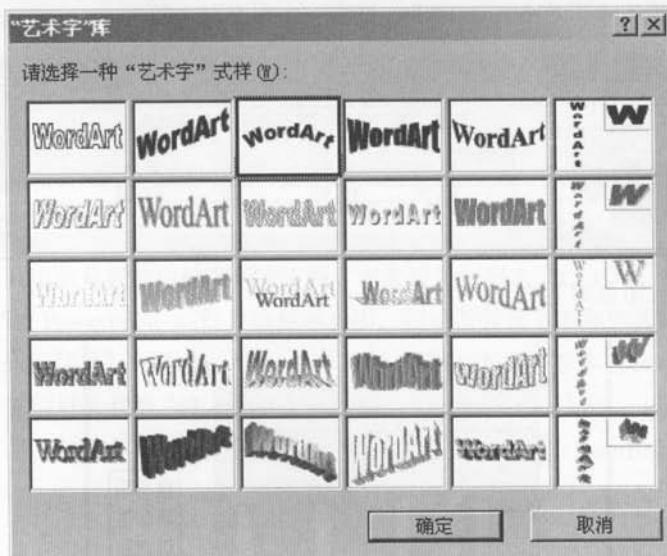
有时候我们会想拷贝 A 页已设妥格式（例如段落、字体）的一个自选图形到 B 页去。这时最常犯的错误是：在 A 页 **Alt-C** 拷贝之后，来到 B 页直接就按下 **Alt-V**，却怎么也看不到贴上来的货。运气好的话（A、B 两页距离够近）上上下下滚动鼠标滚轮，发现贴上的货竟然出现在 A 页。运气不好的话（A、B 两页很远），上穷碧落下黄泉，两处茫茫皆不见。这是因为，您从 A 页移往 B 页后，未曾在 B 页某处按鼠标左键，以至于输入点（input point）还停留在 A 页。不过，即使将输入点移往 B 页，您的货还是会被贴到 B 页中“与 A 页面相同（或邻近）相对位置”的地方，而不是 B 页输入点附近，还需动手将它拖拉到您要的精确位置上。

6.2.4. 艺术字

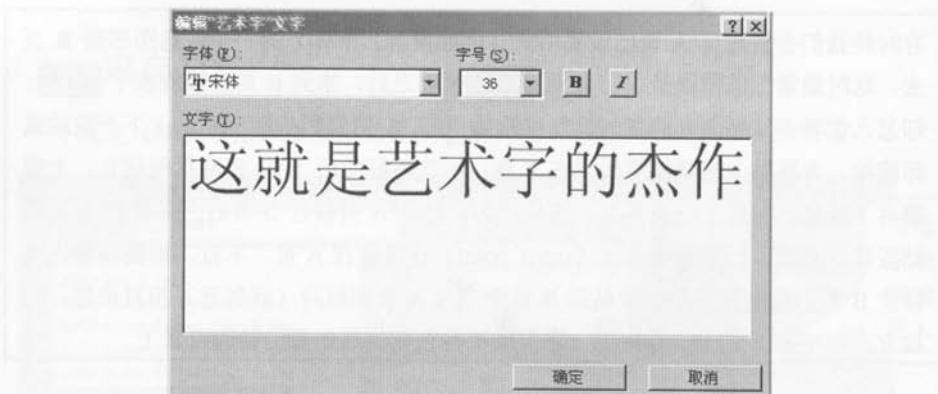
虽然我们可以选用各式各样的字体，但如果要做出广告效果，还是得求助于“艺术字”。下面是**艺术字**工具栏：



点击上述按钮之后，出现以下“艺术字”库：



双击任何一种文字样式，出现以下对话框：



填入内容之后按**确定**，这便获得以下成果：

这就是艺术字的杰作

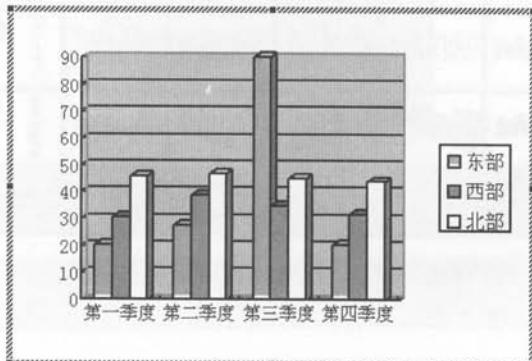
任何时候点击它立刻又会调出上述的**艺术字**工具栏。

6.2.5. 图表

点击【插入→图片→图表(H)】，跳出一个图表编辑器：

	A	B	C	D	E
	第一季度	第二季度	第三季度	第四季度	
1 东部	20.4	27.4	90	20.4	
2 西部	30.6	38.6	34.6	31.6	
3 北部	45.9	46.9	45	43.9	
4					

以及一张对应图表如下：



您可以在上述编辑器中修改内容，其结果将直接反映在图表中。

如果您所撰写的是财务报表之类的文档，Word 提供的图表应该会很有用。

6.3. 插入对象

前面所谈都是从【插入→图片(P)】的路线去插入图片。这一节的被插人物不限图片，而是任何“可插入物”。这些东西由其他程序透过所谓“对象链接与嵌入”(OLE) 技术提供。

【插入→对象(O)】会获得以下对话框：

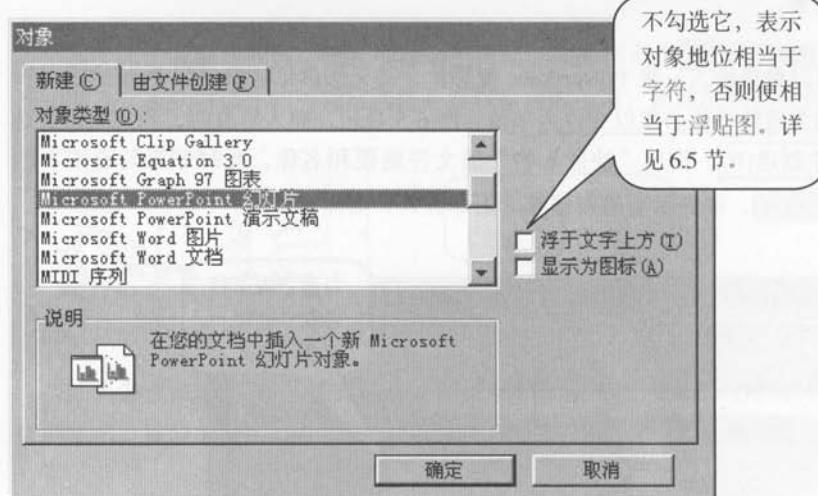


图6-4 【插入→对象(O)】 + 新建(C)

图中方框列出的是我的电脑内所有支持“对象链接与嵌入(OLE)”技术的软件。我个人常以 PowerPoint 幻灯片绘制示意图，因此常需像上图那样选择“Microsoft PowerPoint 幻灯片”，再以 copy/paste (复制/粘贴) 方式将原本在 PowerPoint 中画好的东西复制过来成为“嵌入式对象”。本章一开始提到的 .bmp、.jpg 也都是一种对象，也可以在图 6-4 的对象类型(O):清单中找到，被定义为 PaintBrush (画图) 的输出对象。

以这种方式插入的对象，不仅携带对象内容，也夹带对象来源端的信息。因此如果双击这一类对象，其编辑软件（例如 PowerPoint）会被调用，并且将菜单、工具栏、或其他任何可能必要的工具塞进 Word 视窗，使 Word “改头换面”。这正是

OLE (Object Linking and Embedded; 对象链接与嵌入) 的功能。

6.5 节对于对象 (包括图片) 的位置以及与文字的关系，另有详细讨论。

链接 (Linking) 对象

以上讨论的动作是所谓的对象嵌入 (embedded)。被嵌入物的内容完全塞入文档之中，成为文档的一部分。文件体积也因此变大了不少。

另一种图片 (对象) 插入法称为链接。意思是只塞入对象的属性、路径之类小量信息到文档内，对象主体内容仍保持在文档外部。这么一来文件体积便不会膨胀得太过厉害。

您可以单单嵌入一张 PowerPoint 投影片，却无法单单链接一张 PowerPoint 投影片，因为链接动作必须以文件为单位。图 6-5 显示，插入对象时，我们有机会选择由文件创建(F)，指出“被嵌入物”的文件路径和名称。此时如果勾选图右的链接到文件(K)，便表示要采取链接 (而非嵌入) 动作。

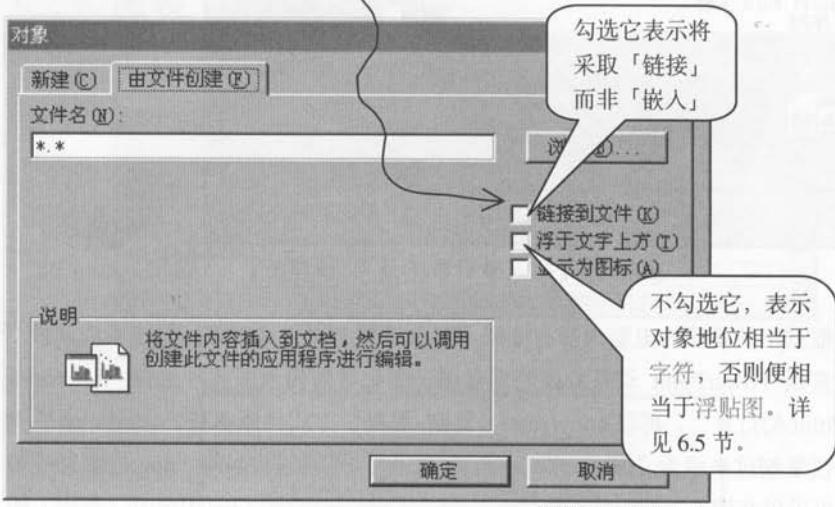


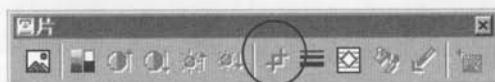
图6-5 【插入→对象(O)】 + 由文件创建(F)

6.4. 图片与对象的剪裁

一般人可能直观以为，点击图片后，图片四周出现的细框，以及八个可拉动的小黑点，就是调整图片大小的工具。不，那只会改变整张图的缩放比例，无法“裁

剪”不想要的部分。

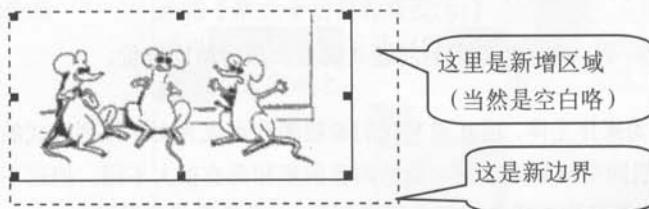
图片工具栏中有一个专门用来裁剪图片的按钮如下：



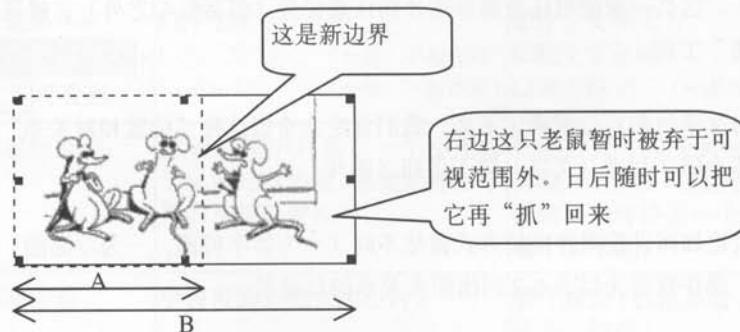
只要任何图片被点击，这个按钮便起作用。点击这个按钮后，鼠标光标会变成~~爪~~形状，此时再去拉动图片四周的八个小黑点：往内拉就是裁剪，往外拉就是扩张。在图外任何区域单击鼠标，便结束裁剪动作。

但是，图片怎么能够扩张呢？

其实无论裁剪或扩张，图片本身毫无增减。如果您点击~~爪~~而后将图片框线往外拉动，代表今后该图的可视范围将是虚线框起的范围，新增的都是空白：



如果您点击~~爪~~而后将图片框线往内拉动，代表今后这张图的范围将是虚线框住的范围，被裁剪的区域日后不再显现（但依然存在，随时可以恢复它）：



Word 摆置图片时，是以图片的可视范围作为尺寸。例如当您将上图（裁剪后剩下的两只老鼠）置中，Word 考虑的图宽是 A 而不是 B。

对于任何类图片（picture-like）对象，例如 PowerPoint 投影片，其裁剪动作和

概念与上述完全相同。

如果嫌鼠标移动不够精准（很多场合的确如此），还可以在图片上点击【右键菜单】的【设置图片格式(I)】（若为对象则是【设置对象格式(O)】），然后在【图片】的裁剪栏目中精确指定左上右下的裁剪尺寸：

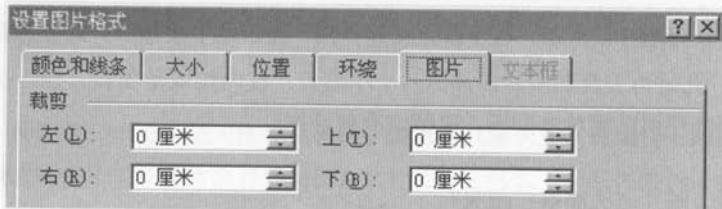


图6-6 裁剪图片/对象

6.5. 图片与对象的精准定位

将图片插入文档很简单。最困扰我们的问题其实是：如何精准定位。

文档毕竟是以文字为操作主体，因此通常我们希望图片也成为文字排列方式的一员，让Word将图片视同字符般地处理（当然两者的宽和高有很大不同，但这不影响排列原则）。Word称这种方式为“与文字排列⁵”，或称为“非浮贴”。

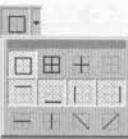
但也可能某些情况下我们希望图片不按照文字排列规矩，而是“独立于文字”，或谓之“浮贴”。这么一来便可任意搬移图片到任意位置（包括版心之外），或甚至进行“文绕图”工程。

一旦图片与文字的相对位置确定下来，我们肯定会希望这种“位置相对关系”能够锁死，才不会跑了和尚（文字）留下了庙（图片）。

稍后再来讨论如何设置两种定位方式截然不同（一为类字符图，一为浮贴图）的图片和对象。现在我要先以表6-2列出图文关系的总说明。

⁵也许说“与文字并排”，文意更清楚些。

表6-3 图文相对关系的说明

操作	与文字排列 / 非浮贴 (我称之为类字符图)	独立于文字 / 浮贴 (我称之为浮贴图)
搬移	无法运用 、、、 键移动之。因为其地位就像字符，字符是无法以箭头键移动的。	可运用 、、、 键移动位置。
设置字符格式	可，因为该图被视为一个字符。例如 【格式→字体(F)】+【字符间距】+【位置/磅值】	不可，因为概念上并不将该图视为一个字符，而是视为一张可随处浮贴的图。
文绕图	不可。因为该图地位相当于字符。把它放到文字中，它不过就是领受一个字符的权利。	可以
大小	可任意调整大小	可任意调整大小
加外框	点击 【格式→边框和底纹(B)】 ，或以 表格和边框 工具栏的 外部框线 按钮完成之： 	以 绘图 工具栏的 (线条样式) 和 (线条色彩) 和 (虚线样式) 完成之。
与文字的相对位置	锁定 (文字上下移，图亦上下移)	锁定 (文字上下移，图亦上下移)
右键菜单+ 图片格式(O)+ 位置可看到：	<input type="checkbox"/> 浮于文字上方 <input type="checkbox"/> 随文字移动 (←注：不起作用) <input type="checkbox"/> 锁定标记 (←注：不起作用)	<input checked="" type="checkbox"/> 浮于文字上方 <input checked="" type="checkbox"/> 随文字移动 <input type="checkbox"/> 锁定标记 (←亦可勾选)
以鼠标拖动：	拖动过程中只显示一个小图标于鼠标光标尖端。落地后即以字符的身份行使权利。	拖动过程中整个外框随鼠标光标而动，可掠过(覆盖于)文字之上，落地后会排挤出一个新段落，或覆盖于文字上。
移动限制	字符可到之处 (版心内)	整个版面 (包括页边、页眉、页脚、天头、地脚)
加阴影	以 【格式→边框和底纹(B)】 完成	以 绘图 工具栏的 (阴影) 完成
特定物		Word 的绘图对象和自选图形都属于此类 (浮贴图)： 

现在，请看下面这个实例。我以【插入→图片→来自文件(F)】的方式，在某行文字中插入一个小小图标：

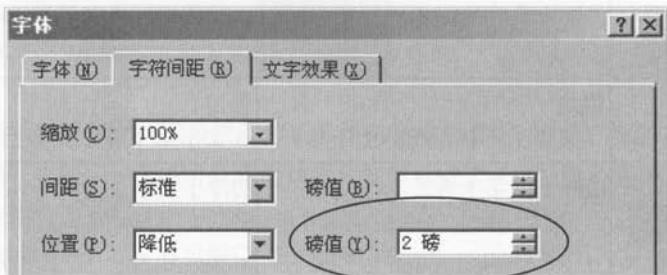
➤ 以绘图工具栏的阴影按钮 完成之 ← (这就是我刚才说的某行文字)

一开始您肯定也是做出这么一个不怎么好看的排列。哪儿不好看呢？那个小图标的位置似乎太偏上了，如果能够下降它的位置，使成为这样：

➤ 以绘图工具栏的阴影按钮 完成之

视觉效果似乎比较好。要做到这一点，必须考虑一件事：这个图是类字符图还是浮贴图？若是前者，我们就要以对付文字的方式对付它，若是后者，就要以对付浮贴图的方式对付它。必须对症下药！

先前已经说了，这个小图标以【插入→图片→来自文件(F)】的方式插入。从图6-3可知，默认情况下浮于文字上方(L)并未被勾选，因此这是一个类字符图。那么，欲移动其位置，就得点击【格式→字体(F)】的字符间距(R)：



在磅值(Y)栏目中以“向下箭头”（表示向下移动）指定移动点数。如果要向上移动，就得使用“向上箭头”。

如果面对的不是类字符图而是浮贴图呢？那就点击它之后直观地以↑、↓、←、→键移动它们。如果嫌步幅过大，希望更精准定位，可点击【右键菜单】的设置自选图形格式(O)或对象格式(O)，然后在位置页中设置水平位置和垂直位置：

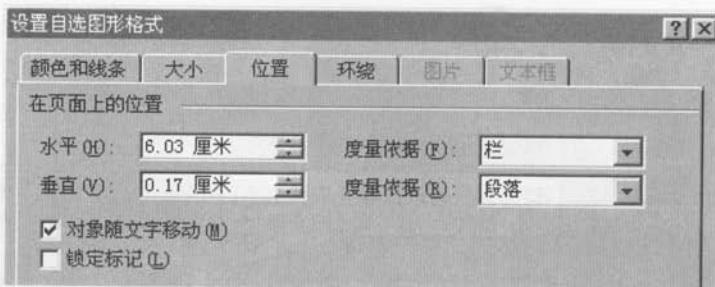
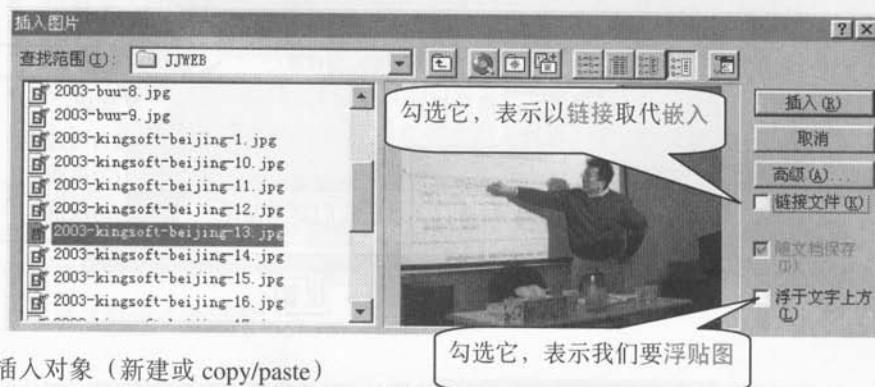


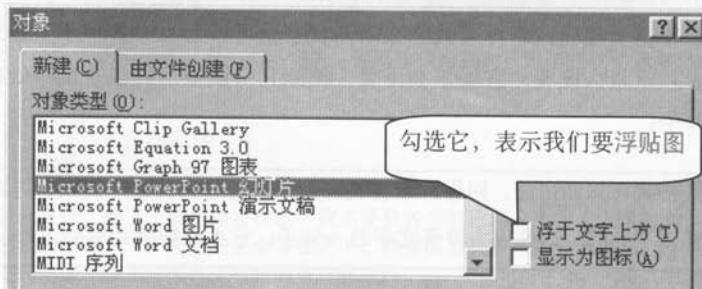
图6-7 [设置自选图形格式] + [位置] 可精确定位一张浮贴图

现在，我们终于可以仔细检阅 Word 之中混乱的用语，并总结如何在插入图片（对象）时指定为类字符图或浮贴图。下面是各个用语的出现画面：

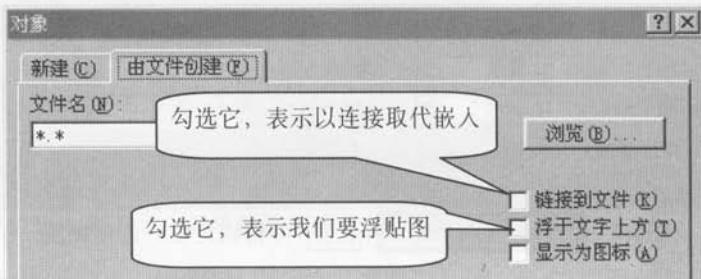
插入图片：



插入对象（新建或 copy/paste）



插入对象（从文件）



查看图片或对象的格式：

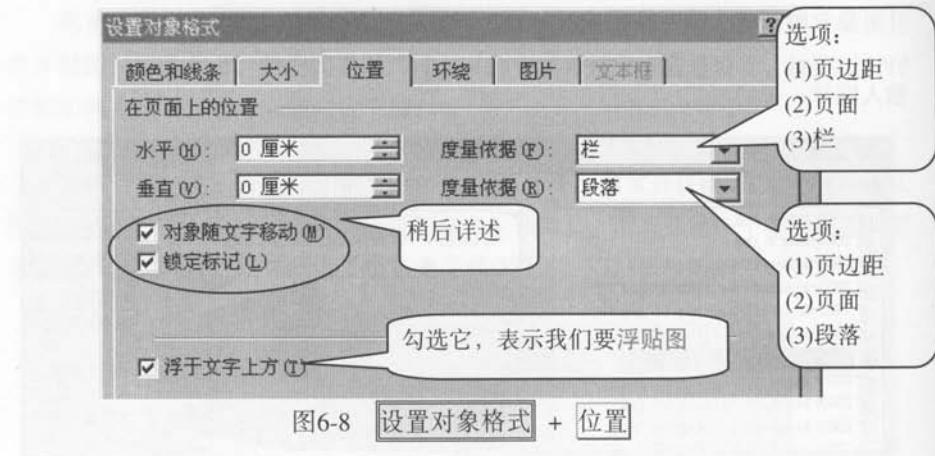


图6-8 设置对象格式 + 位置

如果不勾选**浮于文字上方(T)**，插入的图片/对象将是一张类字符图，摆放时其权利与字符完全相同：字符可以怎么动作，它就可以怎么动作；字符格式（例如边框、阴影）该怎么设置，它就怎么设置。例如它可以被放在以下的□处成为段落文字的一分子：

字符格式（例如边框、阴影）该怎么设置，它就怎么设置。例如它可以被放在以下的□处成为段落文字的一分子；它不能被移到版心之外。

类字符图。地位相当于字符

但如果勾选 浮于文字上方(T)，插入的图片/对象将是一张浮贴图。其地位与字符不同，不能成为段落文字的一分子，而是自个儿占据一个段落（除非“文绕图”，见 6.8 节）。例如，当您把这个对象移到下图的□处，会挤出一个新段落来：

但如果勾选 浮于文字上方(T)，插入的图片/对象将是一张浮贴图。其地位与字符不同，不能成为段落文字的一分子，而是自个儿占据一个段落（除非“文绕图”，见 6.8 节）。

例如，当您把这个对象移到下图的□处，会挤出一个新段落来：

浮贴图。地位不等于字符

再来谈谈图 6-8 的 对象随文字移动选项。这个选项只有在浮贴图中才能作用：

- 如果勾选 对象随文字移动，图片/对象的水平位置基准可以是页面或页边距或栏，默认为栏。垂直位置基准一定是段落。
- 如果未勾选 对象随文字移动，图片/对象的水平位置基准的选项与上述同（三个），垂直位置基准可以是页面或页边距，默认为页面。

换句话说，当浮贴图是 对象随文字移动，它会随上下文字段落的垂直移动而移动（因为它的垂直位置基准是段落）。但如果浮贴图并未勾选 对象随文字移动，它会被锁死在当页位置上⁶，不随上下文字段落的垂直移动而移动（因为它的垂直位置基准是页面或页边距）。

下面的卡通图是一张浮贴图，并且 对象随文字移动。

托天之幸这些瑕疵品多在出库前一刻追回来，没逼得我自杀。三颗脑袋我也想不出来那是什么样的出版思维！



一张浮贴图

终于我告诉自己：够了，我要亲自排版。我不要我的书的任何一个字一张图的大小型式颜色粗细位置……不在我的完全掌控之下。我要保证我能够预先、精准、而完整地看到我的读者将来在纸件上看到的每一样东西。

把输入点移到第一段末尾，按 **Enter**（插入换行符），于是画面变成：

⁶ 在文件的创作或排版上，似乎很少需要将图片/对象钉死在某页某位置。

托天之幸这些瑕疵品多在出库前一刻追回来，没逼得我自杀。三颗脑袋我也想不出来那是什么样的出版思维！



这里新键入一个分行符号；
图随文字垂直移动。

终于我告诉自己：够了，我要亲自排版。我不要我的书的任何一个字—张图的大小型式颜色粗细位置……不在我的完全掌控之下。我要保证我能够预先、精准、而完整地看到我的读者将来在纸件上看到的每一样东西。

但如果该卡通图（一张浮贴图）未勾选对象随文字移动，则上述动作会造成：

托天之幸这些瑕疵品多在出库前一刻追回来，没逼得我自杀。三颗脑袋我也想不出来那是什么样的出版思维！



这里新键入一个分行符号；图
不随文字垂直移动。

终于我告诉自己：够了，我要亲自排版。我不要我的书的任何一个字—张图的大小型式颜色粗细位置……不在我的完全掌控之下。我要保证我能够预先、精准、而完整地看到我的读者将来在纸件上看到的每一样东西。

再谈谈图 6-8 的锁定标记选项。这个选项只有在浮贴图中才能作用：

- 如果勾选锁定标记，怎么样？
- 如果未勾选锁定标记，又怎么样？

如果勾选锁定标记，*On-line Help* 的说明是：“将选用对象与其注释文字放置于同一页上。对象可以以它的锚点移动，但它总是像锚点一样停留在相同的页上”。我无法理解这段话的意思，也没有实验出什么所以然来 ⑧

6.6. 图片与对象的大小

以鼠标拉动图片/对象四周的边框黑点，可以改变图片/对象的展示大小（但由于分辨率并不会改变，所以改变的是缩放比例）。如果嫌鼠标动作不够精细，亦可以右键菜单的**设置图片格式(I)**或**设置对象格式(O)**加以设置，如图 6-9。

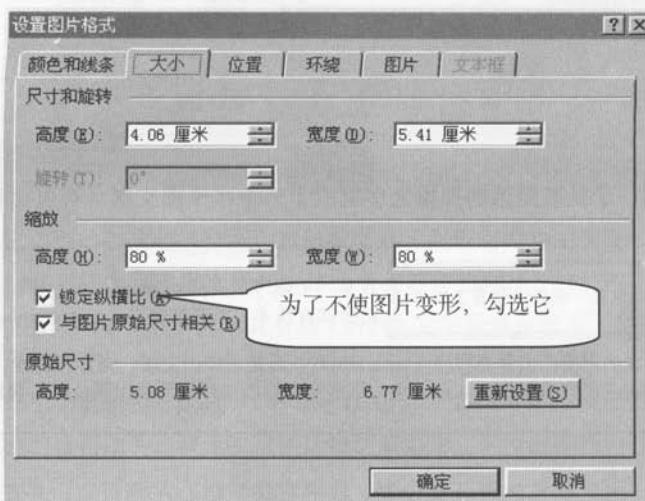


图6-9 调整图片（对象）的大小

6.7. 插入图文框



本节仅适用于 Word97，自 Word2000 以降，功能更强大的“文本框”取代了“图文框”；两者操作方式大同小异。

图片/对象分两种，一是类字符图，一是浮贴图（见表 6-2）。后者可浮贴于页面任何地方，甚至版心之外。因此如果您需要在“页面任意地点”张贴图片，选择浮贴图便是了。

文字或类字符图没有办法提供这种随遇而安的弹性。面对类字符图，将它转换为浮贴图也就是了（点击图 6-8 的**浮于文件上方**即可）。但如果面对文字，我们只有把它转换为图文框，才能解放其位置束缚。

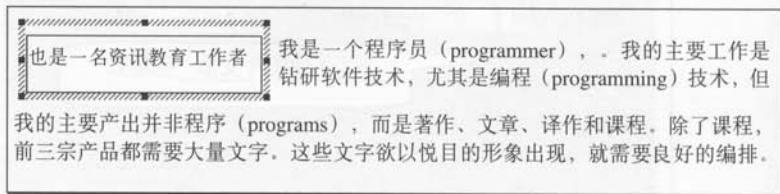
首先圈选目标文字：

我是一个程序员（programmer），也是一名资讯教育工作者。我的主要工作是钻研软件技术，尤其是编程（programming）技术，但我的主要产出并非程序（programs），而是著作、文章、译作和课程。除了课程，前三宗产品都需要大量文字。这些文字欲以悦目的形象出现，就需要良好的编排。

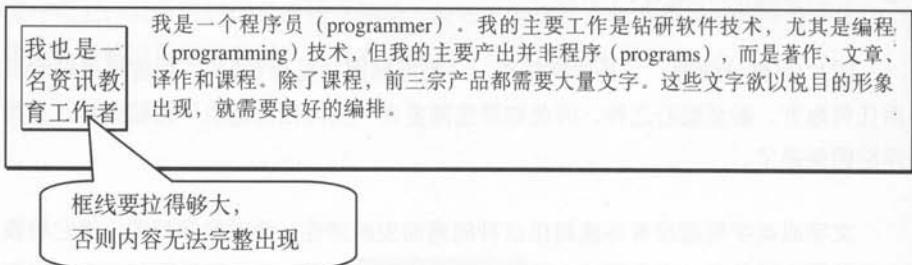
然后点击【插入→图文框(Z)】，出现横排(H)和竖排(V)两个选择：



点击横排(H)，于是被圈选的那段文字就成了一一个水平图文框（有可能移位）：



这个图文框可由鼠标拖放至任意位置。我把它移到版心左侧页边，并直接在框内编修文字（前头加一个“我”字）。而后，在图文框处于被圈选的情况下，点击【格式→段落(P)】，将段落前后距离调整为0,0（提醒您：正文为4,8），将行高调整为13（提醒您：正文为16），使图文框的布局紧凑些（就美学观点，辅助说明文字紧凑些比较好看）。再点击【格式→边框和底纹(B)】，为图文框加上阴影，于是就成了以下模样：



这样的图文框是否会随着它旁边的文字段落的移动而移动呢？点击【右键菜单】的【图文框格式(M)】，出现对话框如下：

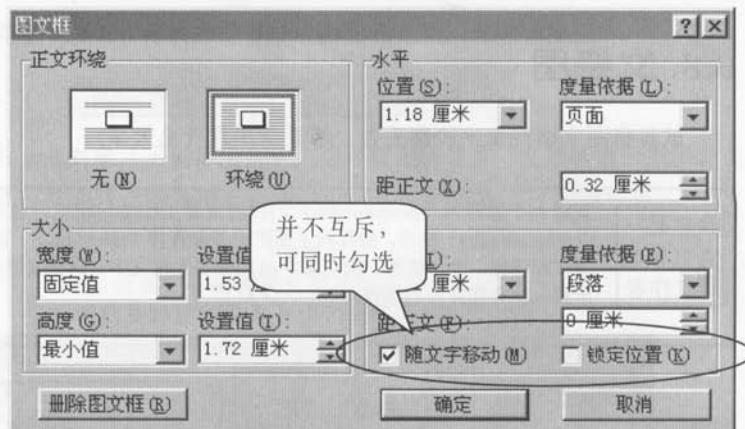
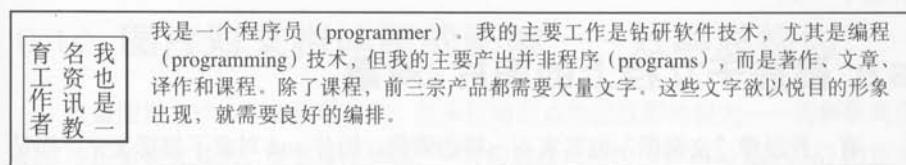


图6-10 图文框 对话框

如果勾选上图的**随文字移动**，表示图文框会跟着它旁边的文字段落移动。但如果那段文字并非因为版面垂直排列次序的变化而自然移动，而是被您硬性搬到另一个地方，图文框将留在原地，其右侧（新的）文字段落也就成了新的贴附对象。

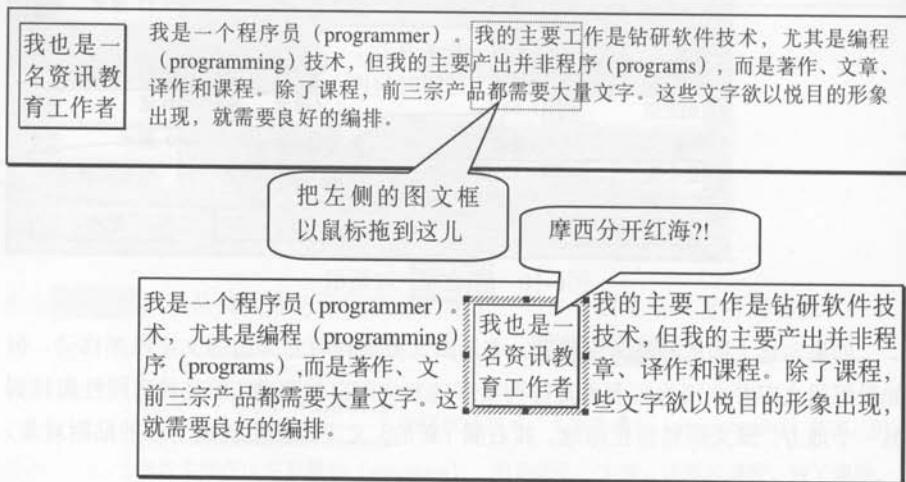
如果勾选上图的**锁定位置**，*On-line Help*如是说：“将图文框的锚点锁定到目前包含锚点的段落上”。我无法理解这段话的意思，也没有实验出什么所以然来④。

如果先前点击的是**竖排(V)**而非**横排(H)**，圈选的那一段文字就变成了一个垂直图文框，并且可能移位。把它移到版心左侧页边，并像先前对待水平图文框的方式来改变其段落格式，加上阴影，就成了下面模样：



6.8. 文绕图

试着将图文框往文字段落丢去，看看会出现什么效果：

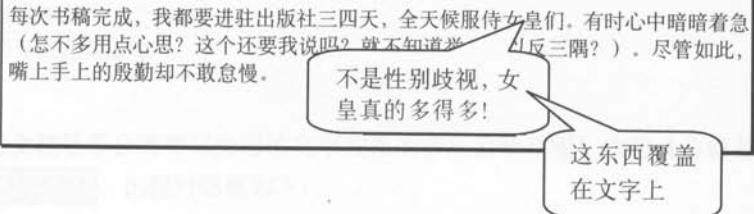


文字纷纷往外排开，腾出空间容纳图文框。这便是所谓的“文绕图”效果。您可以在图 6-10 左上角决定要不要让图文框拥有这种效果。如果取消“文绕图”效果，图文框就完全像是一张浮贴图。

有没有可能把图文框“覆盖”在文字之上呢？也就是不做出“摩西分开红海”①的效果呢？不可能，因为图 6-10 没有提供这种机会。但是图片 and 对象就有可能，详见下一节。

6.9. 覆盖于文档上的图片/对象

有一种很像“文绕图”但其实不一样的效果：图片 and 对象不排挤文字，而是直接覆盖在文字段落上，像这样：



这种特异功能在 Word 提供的所有自选图形和绘图工具都会发生：



我们自己的图片 and 对象（不包括图文框）也可以做出这种“直接覆盖”效果。关键在于图片 and 对象的【右键菜单】的图片格式或对象格式的环绕属性页。

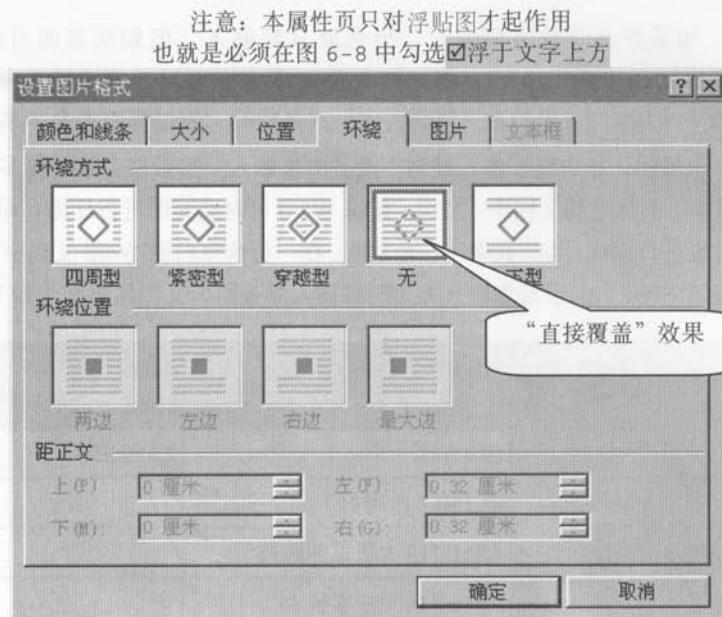


图6-11 设置图片格式 + 环绕

6.10. 图片对文档体积的影响——效率总检讨

矢量图对文档体积的影响不大，但未压缩的点阵图就影响很大——高解析高彩的点阵图影响尤其大。本书写作初期，一有构思我只想快快把相关 hardcopy 图放进来，以免打断创作思绪，因此放进文档的都是 1024×768 的 24-bit 高彩点阵图，文档膨胀得非常惊人，书才写 1/3，文件高达 62MB，硬件设备 (Centrino CPU, 512MB RAM) 渐渐有点左支右绌。于是 I 多管齐下厉行减肥措施（详后），终于瘦身为 12MB 左右。CPU 和 RAM 都得以喘口气，备份所耗时间也少多了。

将图片插入文档，有以下数种文件格式和操作类型可供选择：

- 图片压缩但失真（例如 JPG）。分辨率其实足够就好！JPG 图片对于一般书籍

或论文是绰绰有余了，所以 JPG 是不错的选择。

- 低彩度点阵图。色彩能力足够就好，否则徒然浪费。16 色对于视窗屏幕截图（hardcopy）效果已经很好，体积又不大，放进 Word 之后还可以被 Word 自动压缩。本书所有屏幕截图除了极少数例外，都是 16 色点阵图（bitmap）；如果储存为 JPG 体积会更小，但是当图中带有不少字迹时，JPG 到底还是不如点阵图清晰。
- 嵌入：如果您有对象需要嵌入，当然就只好嵌入。但如果您面对的是图片，可以选择【插入→图片→来自文件(F)】，亦可以选择【插入→对象→由文件创建(F)】（走的是“对象嵌入”这条路）。别以为两者差不多，对体积的影响可能让您大吃一惊。首先，走“对象嵌入”这条路会放进更多（但不是多很多）的信息到文档中。其次，据说 Word 可对点阵图进行压缩，却无法对嵌入对象进行压缩。为此我做了一个实验，在一份全空白的 Word 文档中（1）插入一张点阵图，（2）以对象嵌入方式同样插入那张图。文档体积变化如下。

	DOC 文件	BMP 文件 24-bit 全彩全屏幕截图
原大小	19,456	2,359,350
DOC 插入 BMP 图片	92,160 (压得好)	
DOC 嵌入 BMP 对象	4,851,712 (乖乖的隆咚)	
DOC 链接 BMP 对象	2,446,848 (好家伙)	

- 链接：如果不使用嵌入，改用链接，根据 OLE 技术原理，应可大量节约外部对象带给文档的体积膨胀。但实际情况有点怪，见上表最后横行。另，根据 Word 相关文献显示，Word 使用“绝对路径”来记录图片文件在磁盘中的位置，因此一旦图片文件更名，或文件夹绝对路径有所改变，Word 就无法正确找到图片文件了。但实际情况也有点怪，下面是实验结果：

	BMP 文件改名	BMP 文件搬移	DOC, BMP 一起搬移
(DOC 链接 BMP 对象) 还找得到 BMP 对象否？	Fail 链接错误！即使改回原名 也无效了。	OK (真厉害)	OK

7. 目录自动化.....	205
7.1. 藉由样式自动化产生目录.....	205
7.1.1. 自定目录形貌.....	207
7.2. 手工加入目录项.....	208
7.3. 更新目录.....	210
7.4. 切断目录链接，成为独立文字.....	210
7.5. 单章目录的制作.....	211
7.6. 图表目录的制作.....	214

目录自动化

在创作或排版过程中，藉由某种设置而自动化产生目录，不必再两眼发昏地进行手抄工程，真是件惬意的事。5.1.3 节谈到内建样式时，已经提过这个概念。本章将探讨目录的每一个制作细节。

7.1. 藉由样式自动化产生目录

Word 为我们准备了一些内建样式，其中和目录有关的是：

- ↳ 标题 1 ~ ↳ 标题 9：用于文档的各种标题（如章名、大中小标题名称……）。
- ↳ 目录 1 ~ ↳ 索引 9：用于文档的 1-9 层目录。

其中前导符号 ↳ 表示它们都是段落样式。

如果您真的以**标题 1**、**标题 2**、**标题 3**这样的样式来设置“希望将来能够被提取出来做成目录”的文字（例如章名、大标题、中标题、小标题……），那么，只要将光标移到想要放置目录的地点（例如书籍的目录区），点击【插入 ➔ 索引和目录(D)】，如图 7-1，再点击图右下角的**确定**，这便获得了一份漂亮而完整的目录。

本书目录就是这样做出来的。目录中的各个页码都是超链接（hyperlink），可以链接至文档内容。

这一整份目录其实是个域（见 9.3.6 节），因此文档若有更动，只要再次到目录区（可能面积庞大，其中任何一处都可以），执行 7.3 节的目录更新动作，就可以更新目录了。

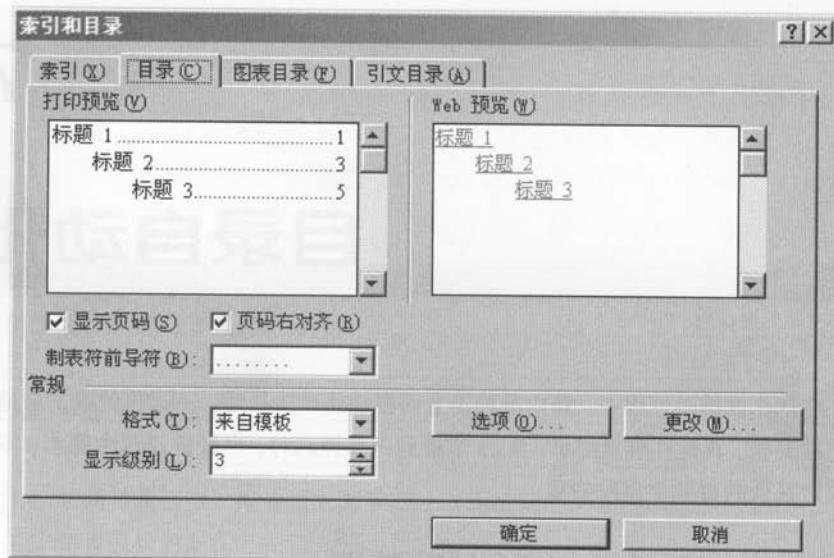


图7-1 索引和目录 + 目录

当然，您也可以在目录区外的任何其他地点做上述【插入→索引和目录(D)】动作，一样会得到最新目录，但现实大概不需要各处散布目录。倒是有些书籍会在每一章前面放一份简要的“单章目录”，7.5节介绍这种做法。

虽然Word默认抓取**标题1**至**标题9**的文字内容做成目录，您也可以改变这种默认行为。是的，请点击图7-1右下角的**选项(O)**，获得以下画面：

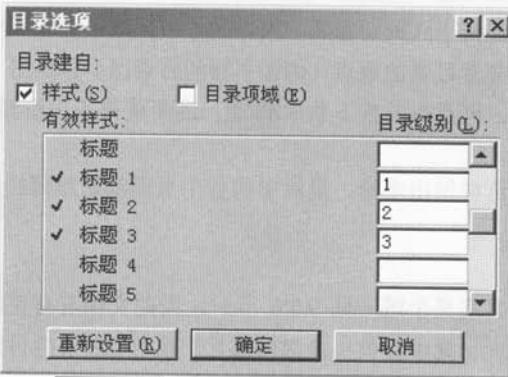


图7-2 目录选项，默认的“用以制成目录”的样式

画面所呈现的是默认情况（抽取**标题1**至**标题3**为目录）。只要将图中的1,2,3去掉（输入点移至文本框中并按`DEL`或`Backspace`即可），再将滚动条拉到您想要制成目录的样式，进行1,2,3...设置，像下图那样，就可以了：

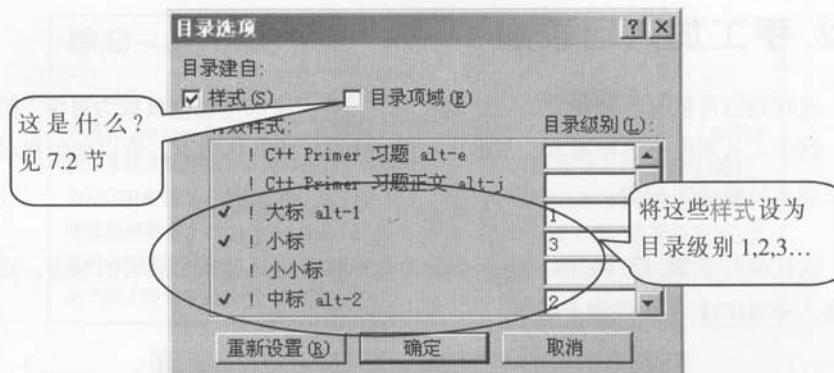


图7-3 目录选项，改设另一些“用以制成目录”的样式

7.1.1. 自定目录形貌

Word 以**目录 1**、**目录 2**、**目录 3**等样式表现各层目录。如果您想要改变目录的长相（通常您会需要 \textcircled{O} ），可在任何时候点击【插入 \rightarrow 索引和目录(D)】，获得图 7-1 画面。在左侧格式(T)方块中选择来自模板，然后点击右边的更改(M)，获得以下画面：

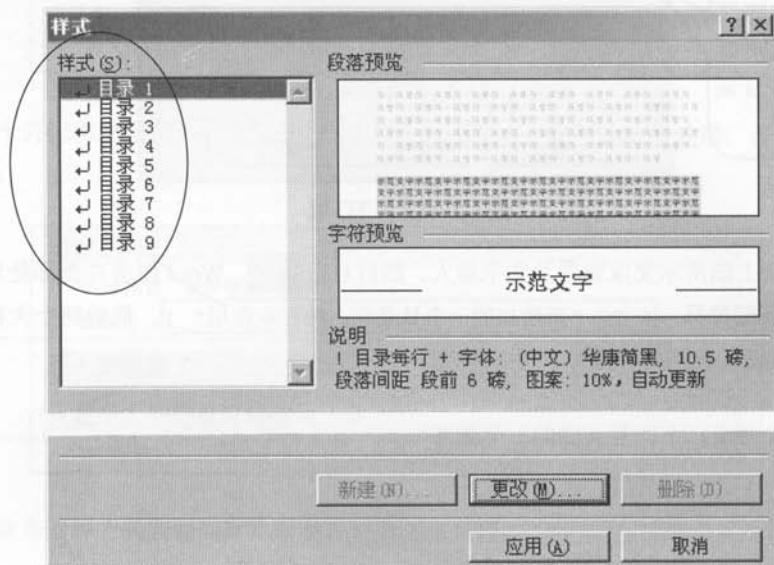


图7-4 目录 1~目录 9 样式，观察与更改

这就可以进行**目录 1**至**目录 9**的修改了。

7.2. 手工加入目录项

有时候您可能会想要在文档任意地点加上一些“将来可被抽取成为目录”的文字。我个人从来没有这种需求，不过显然 Word 设计团队认为有。所以他们提供了手工加入目录项的功能。

这就涉及了 TC 域（9.3.7 节）。首先将光标移至打算插入目录项的地方，点击【插入→域(F)】，获得如下画面：

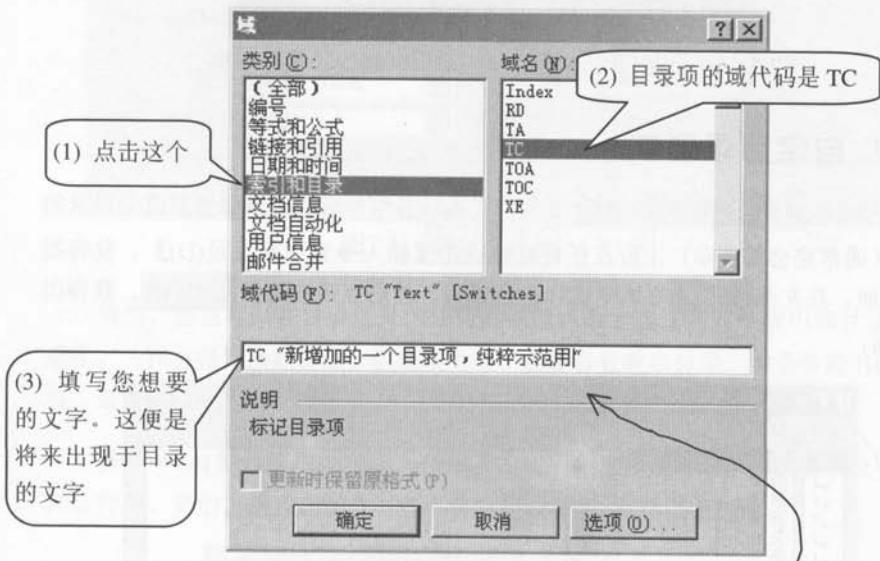


图7-5 插入 TC 域

请依上图所示完成设置及文字输入，然后点击确定。Word 便会在光标处加上这样的隐藏符号：{ TC "新增加的一个目录项，纯粹示范用" }。前后两个大括号是 Word 域的标示（分隔）符号。

举个例子，下面是文档的一个画面：

缘起

过去七 [在这里加一个目录项] 工具，排版了数百篇文章、数十本书籍，并搭配 Acrobat Distiller+Acrobat 制作了几乎相同册数的完整 PDF 电子书（有些开放，有些是私房品）。近年来写作和排版同步进行，又运用 Word 的文档结构图、自动化图表题注、交叉引用、域、宏……，为写作过程引入架构与管理上的协助，降低琐屑事务。[在这里再加一个目录项] 书，与大家分享，祈愿对于“已有若干 Word 运用基础” [TC 完成大型文档排版，或使用 Word 做为创作平台”的人带来帮助。

我在上图标题“缘起”之后加个目录项，文字为“野人献曝罢了”，并在段落结束时加上另一个目录项，其文字为“只是个人一些美好经验的分享”。

·缘起 { TC "野人献曝罢了" }

过去七八 年来我以 Microsoft Word 为工具，排版了数百篇文章、数十本书籍，并搭配 Acrobat Distiller+Acrobat 制作了几乎相同册数的完整 PDF 电子书（有些开放，有些是私房品）。近年来写作和排版同步进行，又运用 Word 的文档结构图、自动化图表题注、交叉引用、域、宏……，为写作过程引入架构与管理上的协助，降低琐屑事务。这些美好的经验使我亟思整理成书，与大家分享，祈愿对于“已有若干 Word 运用基础，希望以 Word 完成大型文档排版，或使用 Word 做为创作平台”的人带来帮助。{ TC "只是个人一些美好经验的分享" }

千万记得在图 7-3 中勾选正上方的 目录项域，否则不会产生效果。得到的结果是：

导读	1
Word 术语，英中繁简对照表	1
缘起	9
野人献曝罢了	9
只是个人一些美好经验的分享	9
本书定位	11

看来这种方式获得的目录项是以 **目录 1** 来呈现。

7.3. 更新目录

只要在目录区点击【右键菜单】中的**更新域(U)**:

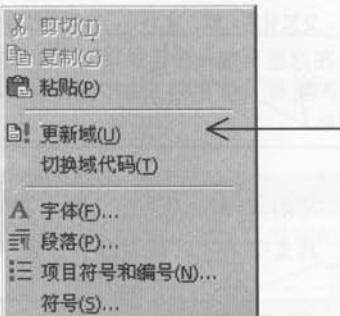


图7-6 更新域

即可更新整份目录。有时候Word还会询问以下问题：

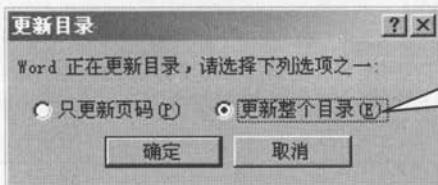


图7-7 Word询问以何种方式更新目录

本书写作过程中，我经常做上述动作，获得最新目录的生成结果。其实也没必要如此，因为如果采用文档结构图（图2-1），结构图就是最新目录，而且我在创作过程中还采用题注和交叉引用，以及章节（标题）自动编号，因此无论任何地点要参考哪个主题或哪一页码，都以交叉引用完成即可，不需实际抄写。我之所以常常更新目录，纯粹是因为操作太简单了，心痒难耐。

7.4. 切断目录链接，成为独立文字

前面已经说过，在Word之中，目录其实是域，因此这份目录必须和文档本体放在一起，否则Word抓不到正确的资料来完成目录。如果您将这整份目录拷贝或搬迁到其他文件，脱离了文档本体，获得的便是目录的当时内容，不复再能够更新。如果要硬生生切断目录和文档之间的链接（很难想象为什么需要如此吧）。7.5节有一个应用），按下**Ctrl+Shift+F9**便可办到，从此目录不复出现域特有的那种灰底显示。

7.5. 单章目录的制作

有些大型文档的设计者十分体贴读者，会在各章（各单元）之前加一份简目。通常这种所谓的章目录分层较少，而且或许不列页码，只做纲要提示。以下说明这种所谓章目录的做法。

本书主目录在章名（**标题 1**）之下列了四层：大标（**标题 2**）、中标（**标题 3**）、小标（**标题 4**）、次小标（**标题 5**）。但我打算在章目录中只列大标（**标题 2**）、中标（**标题 3**）及其页码。

第一个要考虑的是章目录的摆放位置。既曰章目录，当然是在各章之前。此前已为每一章设置一个节，并让每个节从奇数页开始，因此章目录（控制在一页篇幅内）就该放在章名之前的那个偶数页。

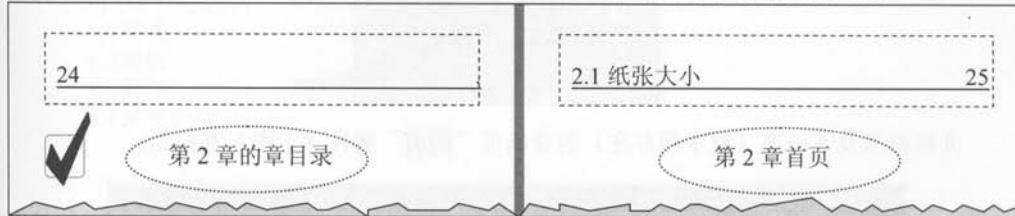
为了不扰动其他单元，将章目录设置为独立一节，似乎是不错的想法。这个节永远只有一页内容出现于偶数页（如果超过一页，由文档创作者检讨），但此偶数页之前还可能有一页空白，因为：

- 如果前一章以奇数页结束，章目录所在的节应该从偶数页开始，并且只有一页。
下图打勾者为新添之页，放置章目录。



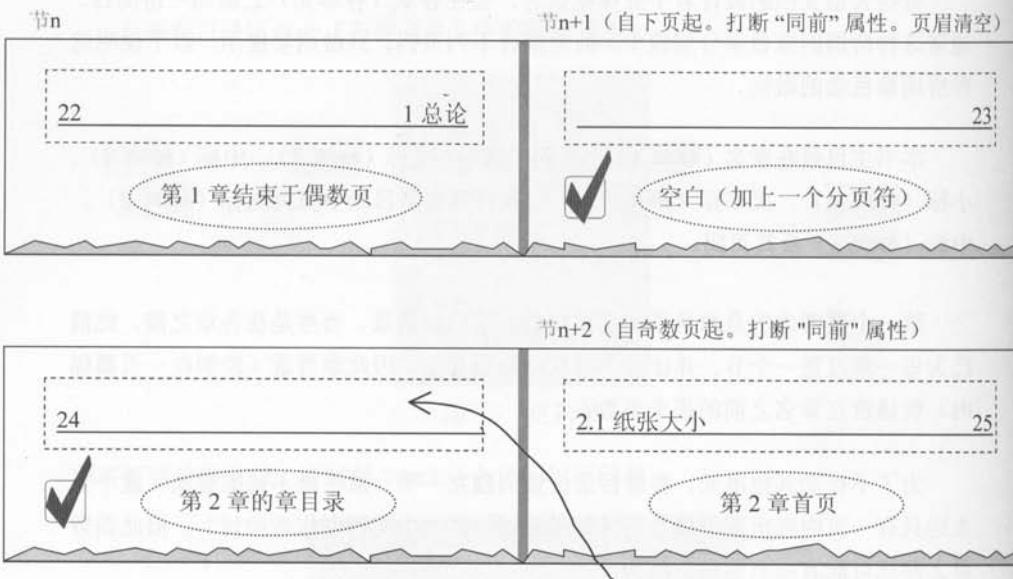
节n+1（自下页起。打断“同前”属性。页眉清空）

节n+2（自奇数页起。打断“同前”属性）



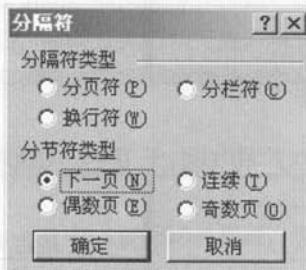
p.24 独立为一个节

- 如果前一章以偶数页结束，章目录所在的节应该从奇数页开始，并于空一页（奇数页）之后才在偶数页开始章目录内容。下图打勾者为新添之页。



请注意，不论上述哪种情况，我把章目录的页眉内容全清掉了，只留下页码，这么做既简单又可被接受。当然，在页眉上摆放任何信息，技术上都没有问题。

由于本例是在各章各自拥有节之后，才安插新节，而新节不一定从奇数页或偶数页开始，因此在【插入→分隔符(B)】时，应该选择◎下一页：

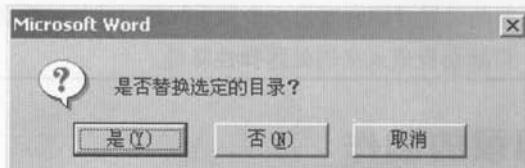


并将新节及下一节（原本即存在）的奇偶页“同前”属性（下图）都打断：



然后才开始清空新节的页眉内容。如果没有顾及上述次序，可能会带来小小麻烦，因为当清空新节的页眉时，如果下一章页眉尚未打断“同前”属性，其内容就会一并被清空，您还得把它补回来。

每一章的章目录只用到主目录的一小部分。我建议在文档全部完成后，才在主目录之外任何地点另做一份单层（只含**标题 2**）目录。只要遵循 7.1 节便可轻松做出这样一份目录。由于彼时已存在一份目录，因此 Word 要求确认：



回答**否(N)**，便可获得第二份目录。再以 **Ctrl-Shift-F9** 切断目录链接（如 7.4 节所述），然后分章切割搬移到各章的章目录区域中。

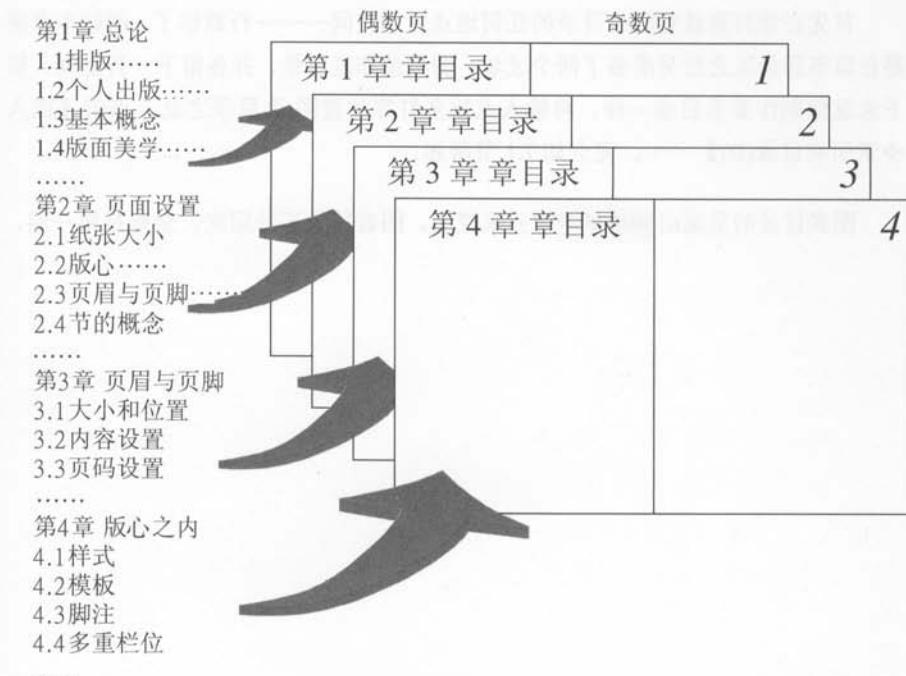


图7-8 章目录的制作示意



制作章目录也许是练习 VBA 程序（宏，第 10 章）的一个绝好的高级题目。因为章目录的不确定性——由左页开始（单页目录）抑或从右页开始（空白页+单页目录）——最是适合程序发挥。



其实如果一开始想到要制作章目录，可令每一章的独立节为“自下个偶数页起”而不再是“自下个奇数页起”。然后在第一个偶数页放置章目录（控制在一页篇幅）。这最省事了，但页眉的设置可能会成为一个棘手问题。此外，这么做会使章目录未能独立于章本身，可能会造成未来的处理弹性降低。

7.6. 图表目录的制作

5.10 节曾经介绍题注。有了题注，图目录、表目录……乃至各种页面元素的目录，唾手可得！

首先在您打算放置图表目录的任何地点留下空间——一行就够了。例如本书便是在章节目录区之后另准备了两个区域：图目录和表目录，并各留下一行空间。接下来就像制作章节目录一样，将输入点移至打算放置图/表目录之处，点击【插入 ➔ 索引和目录(D)】……，完全如 7.1 节所述。

图表目录的呈现由**图表目录**样式来控制。图表目录不分层次，全都是第一层。

8. 索引自动化.....	217
8.1. 索引相关概念.....	217
8.1.1. 索引项 (Index Entry)	218
8.1.2. 索引主项 (索引第一层)	220
8.1.3. 索引次项 (索引第二层、第三层.....)	220
8.2. 以“索引词条文件”批量标记索引项目	220
8.2.1. 索引词条文件的建立.....	221
8.2.2. 让 Word 自动化建立索引	222
8.2.3. 自定索引形貌.....	224
8.3. 逐一手工标记索引项.....	225
8.4. 索引佳例示范与讨论.....	227
8.5. 本书索引制作过程.....	229

索引自动化

8.1. 索引相关概念

索引是什么？精于看书的人都知道它并且善用它。这方面我就不多说了。

索引有多重要？那就不一定，取决于书籍的性质。通常科技类书一定要有索引，才有更高的工具书价值¹。图书馆收藏图书时也会特别列出一条：该书有无索引。任何英法德俄……科技类书的版权页（扉页之后列有出版信息、ISBN 号码、版权声明、刷次日期……信息的那一页），都会写出该书有无索引。例如《Design Patterns》的版权页上写的是：

Includes bibliographical references and index.

并非任何大型文档都需要索引。厂商的技术报告、产品使用导引、业绩报告、财务报告等等，就其性质和价值和规模而言，也许并不需要索引。但至少书籍和论文一定要有索引。书籍的索引是个大工程，论文的索引相较之下轻松一些。

索引的完成必须有以下数个条件的配合：

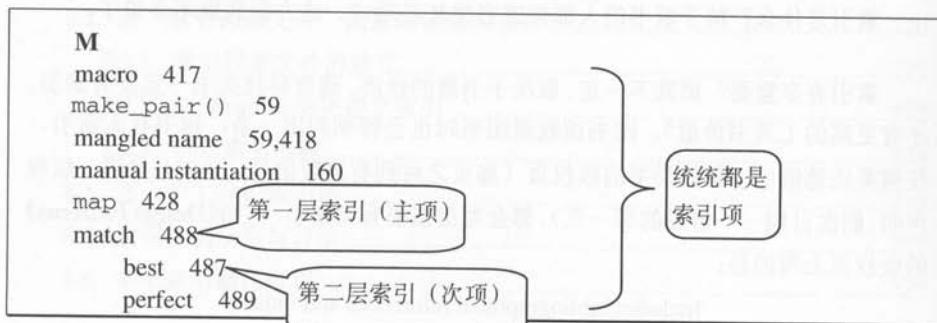
1. 排版软件必须有制作索引的功能。只要号称排版软件，都应该有这个能力！Word 有这个能力②
2. 排版人要有操作上述功能的能力。制造索引的确是比较高级的操作，但相关信息太少也是让排版人“视索引为畏途”的主要原因。如果学习资源够多，排版

¹ 但是台湾许多科技类书（例如相当大宗的计算机技术书籍和电脑应用书籍）并不怎么重视索引。可能的原因包括：(1) 排版人不会制作索引、(2) 创作人嫌麻烦、(3) 创作人和排版人之间没有顺畅的沟通、(4) 因循旧习、(5) 中文并非拼音文字，中文索引有其先天缺陷（只能以笔划顺序为之）、(6) 以上皆是。

人制作起码的索引并不太困难。

3. 文档创作者必须对索引有点概念。这个不难，没吃过猪肉也该看过猪走路。但除了概念之外还必须有能力落实一份具体的“索引词条表”（对 Word 而言；详见 8.2 节），这比较是个挑战。
4. 目前的排版软件在索引制作方面尚未融入太多的人工智慧（只是想当然尔，我并未遍历群贤），所以制作出来的索引只是硬梆梆的页次列表而已。要让索引有高超的价值，文档创作者还必须在这种硬梆梆的初级品上面加工。8.4 节有一个索引佳例，可供大家借鉴与思考。

通常，最简单的索引长相如下：



为了做出这份索引，作者必须在书稿中标出上述每一个词条，告诉排版者“这个词条将成为索引的一分子”。这些词条就是 Word 所谓的索引项（index entry）。上图索引中的 "match" 词条下有第二层索引，其中 "match" 即是 Word 所谓的主索引项，第二层的 "best" 是次索引项，"perfect" 也是次索引项。

文档创作者必须告诉排版者以上所谓的索引项、主索引项、次索引项，排版者才能够在文档中设置它们。

8.1.1. 索引项 (Index Entries)

索引项其实是一个域（见 9.3.9 节）。或者说，Word 以一个域表现一个索引项。

这个域的代码是 XE（想来应是 "indeX Entry" 的缩写），格式如下²:

```
{ XE "Text" [option] }
```

举个例子，如果您将 "generic" 标记 (mark) 为索引项（操作手法于 8.3 节详述），而且没有任何特殊设置，那么 Word 便在该词条之后（紧邻）放上一个这样的东西：

```
{ XE "generic" }
```

这东西是隐藏的，必须点击常用工具栏上的  才能看到它。

XE 域的另一种格式是：

```
{ XE "Text:SubText" [option] }
```

其中的 "Text:SubText" 将在 8.1.3 节解释。

XE 域的选项 (Options)

XE 域的选项 (options) 有以下数种。（阅读前请先建立一个观念：文档可拥有多份索引；【插入→索引和目录(D)】+索引(X)即可做出一份索引）

- \b 在索引项所得页码上套用粗体格式。例如 { XE "arrays" \b } 会显示 **arrays 56**。如果索引项的相关样式（索引 1-索引 9）是粗体，则此选项会移除粗体格式。可与 \i 同时存在。
- \f "Type" 定义索引项的类型。这些类型只会被“也指定了相同类型”的 INDEX 域 (9.3.8 节) 取出制作成索引。例如 { XE "access" \f "a" } 指定的类型是 a，那么只有 { INDEX \f "a" } 才能抓出那个索引项制作成索引（也许我们可以把这样做出来的一份索引称为“a 类型”索引）。
- \i 在索引项所得页码上套用斜体格式。例如 { XE "arrays" \i } 会显示 *arrays 56*。如果索引项的相关样式（索引 1-索引 9）是斜体，则该选项会移除斜体格式。可与 \b 同时存在。
- \r Bookmark 将“标示有特定书签”的页面范围作为此索引项的页码。例如，{ XE "base class" \r BaseClass } 会令该索引项在索引之中被显示诸如这样的型式：base class, 20-25。不过我个人从来没有正确试得这个选项功能！我的理解是：一个书签只可能置于（指向、代表、参照）某惟一页，

² 这种表示法对于无编程技术背景的排版者恐怕比较陌生。中括号 [] 表示可有可无的、选择性的。option 表示选项，通常以 \r, \e, \n...型式来表现。

那么又如何能够使得索引项的页码呈现一个区间范围呢？不解④

➤ \t "Text" 以特定文字替换页码；该文字需以引号括住。例如，

{XE "derived classes" \t "请参阅 base classes" } 会令该索引项在索引之中被显示为 `derived classes, 请参阅 base classes.`

8.1.2. 索引主项（索引第一层）

出现于索引第一层的就是主索引项，也就是 XE 域内那个以双引号括起来的词条：

{ XE "Text" [option] }

8.1.3. 索引次项（索引第二层、第三层……）

出现于索引第一层以下的，统统称为次索引项，也就是在 XE 域内，以双引号括起来的文字中，位于冒号之后的那个词条：

{ XE "Text:SubText" [option] }

若要进行第三层索引，可以这样设置：

{ XE "Text:SubText:SubText" [option] }

虽然 Word 允许用户制作多达 9 层索引，不过实用上以 2 层为主要，偶有 3 层。我个人鲜少看过 3 层以上的索引。

8.2. 以“索引词条文件”批量标记索引项目

面对索引项，Word 提供两种标记（mark）办法，一是批量作业，一是逐一作业。两相比较，当然是批量作业好：您可以先做一份完整书面资料，我称之为“索引词条文件”，并仔细推敲琢磨其中所列的 1,2,3... 层索引词条的关系¹。换句话说批量作业远比逐一作业更系统化、更组织化，比较不会迷乱。

批量作业有很高的运用价值，但其选项设置的弹性却输给逐一作业方式。两者我都会做介绍。

¹ 没有好的构思，别想做出好的索引；这不像制作目录那么容易而直观。

8.2.1. 索引词条文件的建立

首先，我们需要一个索引词条文件，记录每一笔主索引项和每一笔次索引项。这个文档必须是个双列表格——这是惟一条件，至于其内文字的格式细节如字体、字距、段落……都无所谓。第一列填写索引项目（索引词条），第二列依照 "Text:SubText" 格式填写主项与次项。如果打算制作第三层索引，则依照 "Text:SubText:SubText" 的格式填写。但是表格中无法加上 p.219 所说的选项。下面是个例子：

function templates	function:templates
class templates	class:templates

这个表格的意思是：我希望取得文档中的 "function templates" 词条的所有出现页码，并以这种方式排列：

function	templates, 4, 9, 12, 18, 28
----------	-----------------------------

我还希望取得文档中 "class templates" 词条的所有出现页码，并以这种方式排列：

class	templates, 4, 9, 12, 18, 28
-------	-----------------------------

下面是实际获得的结果（操作过程稍后介绍）：

class	function
templates, 9, 23, 24, 27, 28, 29, 32, 35, 37, 47, 51	templates, 9, 11, 15, 17, 21, 23, 24, 37, 45, 47, 53

这份结果被 Word 自动设为双栏如上——双栏正是索引的惯常表现形式。但如果分栏位置不理想，用户必须手动调整（请参考 5.4 节）。

如果我把索引词条文件中的表格改成这样：

function templates	template:function
class templates	template:class

意思是希望取得文档中 "function templates" 词条的所有出现页码，并这样排列：

template
function, 4, 9, 12, 18, 28...

还希望取得文档中 "class templates" 词条的所有出现页码，并这样排列：

template
class, 4, 9, 12, 18, 28...

实际获得的结果（当然）也由 Word 自动以双栏形式排列。

索引词条文件对中文也适用！

请注意，这样制造出来的索引，排列次序和索引词条文件中的表格内容次序无关，而是和索引词条的中文笔划多寡或英文字典次序有关，并且由 Word 自动排序。

8.2.2. 让 Word 自动化建立索引

首先如上一节所说建好索引词条文件，然后点击【插入→索引和目录(D)】+
索引，获得以下画面。

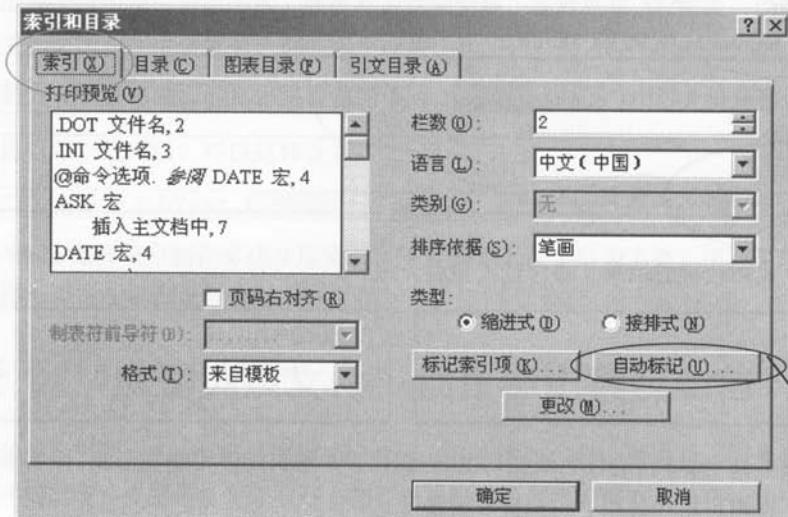
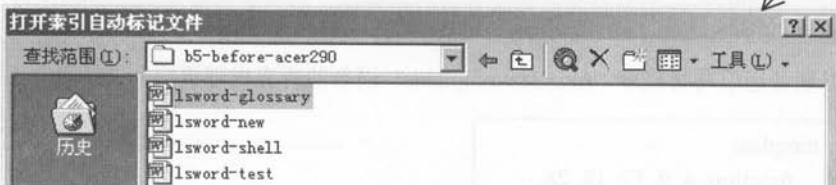


图8-1 索引和目录

点击右侧的 自动标记(U)，于是 Word 向您索求索引词条文件：



选定索引词条文件后，将光标移至文档中您希望放置索引的地点，再次点击【插入→索引和目录(D)】，如图 8-1，并点击右下角的确定，Word 便会自动为我们制造出一份索引（它会在文档中插入一个 INDEX 域，详见 9.3.8 节）。此时 Word 会自动显示所有的隐藏符号，为的是秀出域 XE 让您看见。

假设我以本节所说的批量标记方式，为“页面视图”这个词条建立对应的 XE 索引项【XE “视图:页面视图”】，而后任何时间选[¶]，将隐藏符号全部显露出来，便可看见类似这样的画面：

■ 2.3.3. 页面视图【XE “视图:页面视图”】

关于页面视图【XE “视图:页面视图”】之开启、呈现、关闭（切换至其他视图），请见图 2-1 及其说明。如果当时也处于文档结构视图中（带出了结构图），那么页面视图就出现于右视窗，否则就出现于全视窗。

页面视图【XE “视图:页面视图”】最棒的特点是，忠实呈现所有页面元素（扉页、题献、序言、目录、前言、正文、页眉、页码、附录、参考文献、书目、索引）于制版和印刷时的实际表现。想当然尔这会耗用大量电脑运算。以往硬件设备不足，专家有时候会建议您选择性地使用其他视图来进行编辑。如今速度问题已经罕见[¶]，因此我建议完全在页面视图下进行所有工作。

■ 文档结构图 + 页面视图【XE “视图:页面视图”】 = 创作者的好帮手。

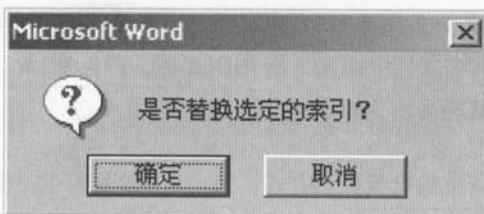
上图以框线框起的都是 Word 为我们加上的 XE 索引项。这些索引项虽然会推挤正常的内容，但实际打印时它们并不会出现，所以不会对页面带来不该有的影响。您注意到了吗，每一段落中的相同词条只有第一个会被冠以 XE 索引项。

重复建立索引，可以吗？

通常，整份文档定稿之后我们才会开始制作索引。但有时因为操之过急、有时因为疏漏，有时的确有所需要（稍后我会讨论“一份文档拥有两份索引”的情况），必须重新或再次制作一份索引。怎么办？

这时候我们必须重复上一页的所有动作：重新点击一次自动标记(U)（也就是重新载入“索引词条文件”以便重新标记索引项）、重新点击一次确定（用以生

成索引)。按下**确定**按钮时, Word 会询问您:



如果按下**是(Y)**, 旧索引便获得了更新。但是请注意, 虽然新制作出来的索引会反映索引词条文件中的新增项目, 但索引词条文件中被删除的项目仍会出现在新制作出来的索引中(那当然不好) — 因为原先标记的索引项(XE 域)还在。

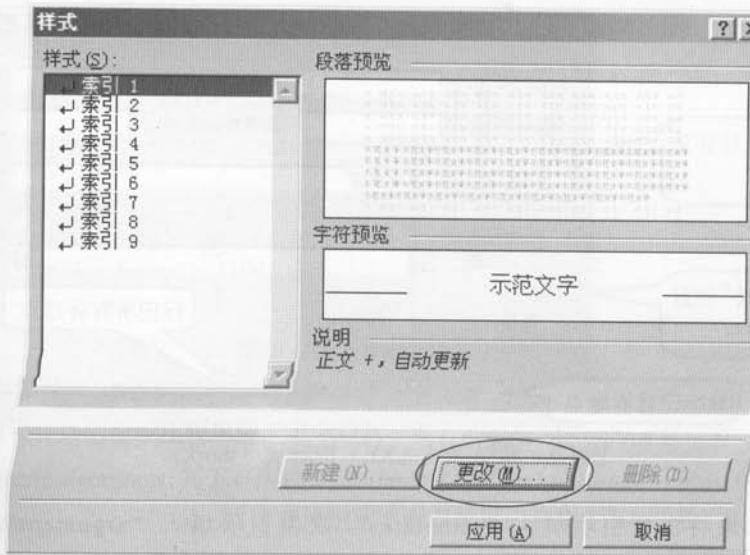
如果按下**否(N)**, 旧索引会保留不动, 另做出一份新索引。

一份文档带有两份索引, 可以吗?

当然可以。有些书籍的确带有两份索引。例如《More Effective C++》就有一份 General Index 和一份 Index of Example Classes, Functions, and Templates。

8.2.3. 自定索引形貌

Word 以**索引 1**、**索引 2**、**索引 3**……样式来表现各层索引。如果您需要改变索引的长相(通常您会需要②), 请在任何时候点击【插入→索引和目录(D)】, 获得如图 8-1 的画面, 然后在其左侧**格式(T):**方块中选择**来自模板**, 而后点击该图右下角的**更改(M)**, 获得以下画面:



再点击下方的**更改(M)**，获得如图 5-2 的画面，这就可以进行**索引 1** 至**索引 9** 样式的修改了。

8.3. 逐一手工标记索引项

逐一加入索引项，并不是个好差事，在工作流程上会把事情弄得很杂乱，又没有留下一份可以回顾检讨的书面资料。如果您大事底定（使用先前介绍的“索引项目批量标记”方式），最后才临时需要加上一两个索引，倒是可以这么办（手工加入）。

在文档中手工标记（mark）索引项的做法是：圈选希望作为索引项的文字，例如“templates”，或是将输入点移到打算插入索引项的地点，然后点击【插入→索引和目录(D)】+【标记索引项(K)】或按下 **Alt+Shift+X**，出现一个 modeless 对话框¹，此时您所选取的文字会自动出现于图 8-2 对话框的主索引项栏（可改动它，因为索引项不见得就必须是主索引项）。

¹ 所谓 modeless 对话框，就是那种“不会阻碍程序其他部分之操作”的对话框。例如【编辑→查找...】、【编辑→替换...】所产生的对话框都是。另一种所谓的 modal 对话方块，就是那种“会阻碍程序其他部分之操作”的对话方块，例如【文件→打开...】、【文件→另存为...】对话框都是。

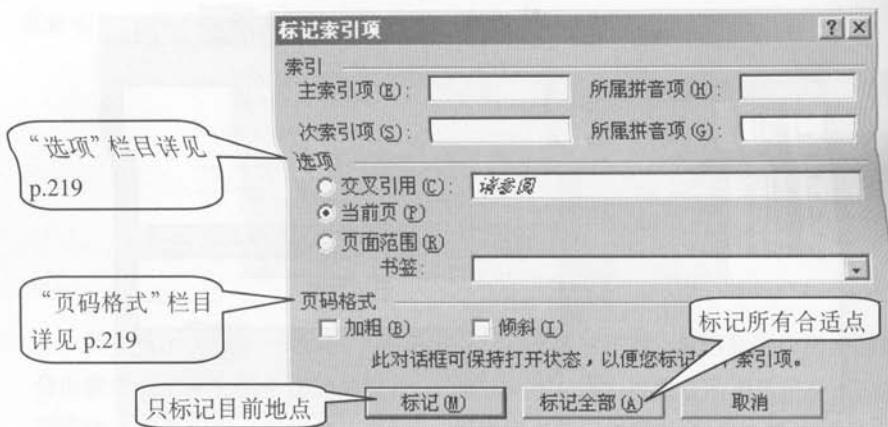


图8-2 索引项目 (XE) 的标记 (mark)

如果将主索引项填入 "templates"，次索引项填入 "argument"，按下 **标记全部(A)**，文档中的所有出现的 "templates" 词条便会被 Word 加上隐藏的 `{ XE "templates:argument" }`。此时如果立刻再一次对 "templates" 进行 **标记全部(A)**，Word 会回复“无标记的索引项”²；如果只是进行 **标记(M)**（并将主索引项填为 "jjhou"），那么被圈选（欲作为索引项目）的文字后方会被加上 `{ XE "jjhou" }`。如果随后又在文档某处加了一个 "templates" 词条，然后又再一次对 "templates" 进行 **标记全部(A)**，Word 会在新添加的那个 "templates" 词条后方加上隐藏的 `{ XE "templates:argument" }`。

最后，回到 **索引和目录** 对话框，按下 **确定** 钮，产生类似这样的结果：

templates
argument, 3, 4, 5, 7, 9, 10, 11... (后略)

Word 进行标记 (mark) 动作时，只考虑文字内容（区分大小写），不考虑其样式。如果 **标记全部(A)** 的是 "temp"，那么文档中所有 "templates" 的前半段 "temp" 也都会被标记，变成 `temp{ XE "temp" }plates`。此外，**标记全部(A)** 会对目录文字也进行标记，这通常不是我们所要的，必须事后清理^②

也可以使用 9.2.1 节所说的方法 (**Ctrl-F9**)，并参考 9.3.9 节的描述，加入 XE 域，作为索引项。

² 这个信息会不会有误呢？会不会应该是“找不到未标记的索引项”？

8.4. 索引佳例示范与讨论

下面是摘录自《C++ Primer》的一小部分索引：

generic algorithms,
...
heap generic algorithms, 1194-1197
make_heap(), 1194
pop_heap(), 1195
push_heap(), 1195
sort_heap(), 1195

这种索引词条的组织比较单纯，上下层次分明：generic algorithms 之下有一类称为 heap generic algorithms，其下有四种 algorithms 分别是 make_heap()，pop_heap()，push_heap()，sort_heap()。欲做出这个索引效果，只要在 Word 的“索引词条文件”中安排双栏内容如下：

make_heap()	generic algorithms:heap generic algorithms:make_heap()
pop_heap()	generic algorithms:heap generic algorithms:pop_heap()
push_heap()	generic algorithms:heap generic algorithms:push_heap()
sort_heap()	generic algorithms:heap generic algorithms:sort_heap()

以及为“heap generic algorithms”词条加上 `XE \r` 选项。

下面是摘录自《C++ Primer》的另一小部分索引：

base class,	以特定文字取代页码
<i>See Also</i> classes, class members, class scope,	
constructors, derived classes, destructors,	
inheritance	
abstract base classes, 885-889, 927	
access,	
protected members, 891	
to base classes, 977-984	页码区间范围
to members, 900-908	
to private base classes, 982-983	
to protected base classes, 1060	

以上索引告诉读者：

- 关于 base class，可参考 classes, class members, class scope...

- 关于范围更窄化的 abstract base class，可看 p885-p889, p927
- 欲访问 base class 的 protected members，可看 p891
- 欲访问 base class，可看 p977-p984
- 欲访问 base class 的 members，可看 p900-p908
- 欲访问 private base class，可看 p982-p983
- 欲访问 protected base class，可看 p1060

这些词条并不像上个例子有那么严明的上下层次。其中显现的“以 *See Also...* 取代页码”、“页码区间范围”等等，都是 Word 可以做出来的效果（见 p.219 的 XE 域选项），所以技术层次不是问题（虽然……呃……我在 p.219 说过，我自己才疏学浅，尚未能够正确试出 \r 选项），关键在于如何完善设计出“词条与词条之间的关系”，并正确插入对应的索引项目。

再举一个例子，仍然摘录自《*C++ Primer*》索引：

constructors, 691-702
 for base classes, 908-919
 in multiple inheritance, 970-971
 in single inheritance, 908-914
 in virtual inheritance, 998-1002
 memberwise initialization, 943-948
 as conversion functions, 780-782

这个例子和上个例子一样，各词条之间并非有严明的上下层次关系。

从后两个例子可以看出来，显然，排版人不可能单方面决定索引词条，更不可能单方面设计出“词条与词条之间的关系”。这些都超出了排版人的责任范围。至于文档作者，虽然可以不太困难地整理出索引词条，却也不可能在对索引制作工具毫无所悉的情况下设计出正确而完善的“词条与词条之间的关系”，因为诸如 for,to,with,in,as,of 等修饰词都必须在“指定文档中的主索引项（8.1.2 节）和次索引项（8.1.3 节）”时逐一设置。

好的索引，绝对是排版者和文档作者充分合作下，才有可能呈现的产物。

8.5. 本书索引制作过程

我个人专注于电脑技术的钻研和写作已逾十年，在海峡两岸各有多本书籍出版，这样的经历或足以让我谈谈台湾和大陆的计算机书籍。两岸计算机书籍不论应用类或技术类，只要是学术圈以外的，鲜少有索引，包括我自己的作品（惭愧！）。其原因或可归纳为 p.217 脚注 1 的六大项。

虽然我曾经做过索引³，却是 100% 人工苦哈哈做出来的。撰写本书之前，我从未藉由 Word 的帮助制造索引（顶多只用它来寻找词条出现页码）。老实说，几近三个星期的摸索，才让我对本书的索引制作有了足够的概念和进展。这一节记录的便是这份经验。

索引的精细有着级别的区分。最普通的一级是单纯将词条（索引项目）出现页码陈列出来。再上层楼则是对这些页码做筛选工作，只留适当者。更好一些的是为某些词条加上 "See Also"（“参阅”）。再好一些则是在某些词条（经过筛选后的）出现页码中加上诸如 as, with, in, of, to, for... 与其他词条间的关系描述。

数十本计算机书籍的翻译经验使我深信，索引的制作肯定是一种 know how，国外大出版社肯定有专业团队负责这个工作⁴。以我的最大部头译作《C++ Primer 3/e》（by Stanly B. Lippman & Josée Lajoie）为例，该书总页数 1237，索引占了 39 页，估计涵盖 3,500 个词条，其中对于各词条的描述、词条间的关系，都相当详尽。作为一本工具书，又是一本大书，这样精细的索引非常有价值。

我以做出上述水准的索引为目标，并在此将整个实作心得分享给您。

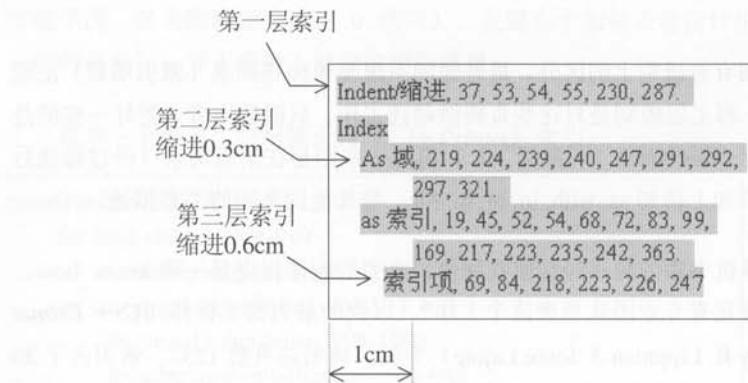
■ 首先确定索引的层级数目，进而决定各层索引的表现方式

本书采用 3 层索引，每一层索引都在 1.0cm 处折行，每一次层索引退缩 0.3cm。我以下表的一个自定样式和 3 个内建样式完成上述要求，视觉效果请见本书 p.385 索引。

³ 1998 以前我的著作没有索引！1998 以后的惟一著作《STL 源码剖析》虽有索引表，却是以笨功夫（100% 人工）做出来的。近 5 年来我把大部分心力放在翻译上头，虽然每一本译作都保有索引，却是直接拿原文书索引来使用。这么做需要三个条件配合：(1) 中英页面对译 (2) 保留诸多英文术语 (3) 补充中英术语对照表。

⁴ 我很羡慕国外作家可以不必一肩扛起索引的重责大任，也希望有朝一日了解作家和索引团队之间的合作模式——我相信作家无法完全置身于索引事外。

类型	样式名称	细目描述	说明
↓	索引文字	正文 + 字体: 8.5 磅, 行距 最小值 11 磅	
↓	索引 1	索引文字 + 缩进: 悬挂 1 厘米, 段落间距 段前 2 磅 段后 0 磅, 自动更新	搭配 第 1 层索引
↓	索引 2	索引文字 + 缩进: 左 0.3 cm 悬挂 0.7 cm, 段落间距 段前 0 磅 段后 0 磅, 自动更新	搭配 第 2 层索引
↓	索引 3	索引文字 + 缩进: 左 0.6 cm 悬挂 0.4 cm, 段落间距 段前 0 磅 段后 0 磅, 自动更新	搭配 第 3 层索引



■ 利用 p.220 的 XE 域 \t 选项，制作 See Also

假设我希望，当读者检索“视图→大纲视图”或“视图→页面视图”时，这份索引能够带引读者去看“Outline”或“Print Layout”这两个词条。欲达到这个目的，只要在文档任何地点（稍后续论）写下两笔 XE\t 即可：

```
{ XE "视图:大纲视图" \t "参阅 Outline" }
{ XE "视图:页面视图" \t "参阅 Print Layout" }
```

至于最终真正记录页码的“Outline”和“Print Layout”两个词条，其 XE 索引项目既可采用 8.2 节的批量作业方式也可采用 8.3 节的手工逐一作业方式加入文档之中。稍后我将采用批量作业方式。

一开始所有这类 XE\t 都被我放在“索引”章前。为了便于检视和检讨，每一笔 XE\t 都独立一行（尾端有换行符）。但是这么一来，虽然这些设置本身并不影响页码（只要它们被隐藏起来，☒），换行符却总是有作用，会影响其后

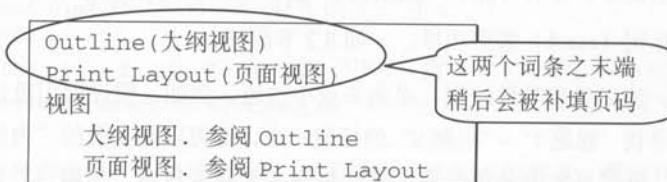
文档（本例为“索引”章）的页码。如果把换行符都拿掉，就不会影响页码，却又造成不易检阅与检讨。为此，我把所有 `XE\t` 放到“索引”章之后（也就是整份文档最后），这就无论如何不影响其他内容的编页了。最终完成全部索引后，切断索引的域关系（按 `Ctrl+Shift+F9`），再删除所有 `XE\t`，就可以让整份文档干干净净。不过也很可能您会想要保留这些 `XE\t` 设置，以备将来修改，那么就别忘了在删除它们之前先复制一份储存入其他文件。

■ 利用 p.220 的 `XE` 域 `\t` 选项，模拟应有的词条关系

接续以上动作。为提醒自己已经设了两个“稍后必须做出对应之 `XE` 索引项目”的词条，我又加上两个 `XE\t`：

```
{ XE "Outline(大纲视图)" \t }
{ XE "Print Layout(页面视图)" \t }
```

此时如果点击【插入→索引和目录(D)】+【索引(X)】+【确定】，可获得索引如下，对于稍后制作“索引词条文件”有助忆效果：



这种“以 `XE\t` 模拟日后索引之实际所含项目”的手法，可对索引的预先整体掌握带来帮助。这么做的另一个主要原因是，将来（以批量手法）真正为每个词条加上 `XE` 索引项目时会给文档带来“破坏性”的影响⁵，因此在正式插入 `XE` 之前，先以“较单纯且位置得以集中”的 `XE\t` 模拟索引全貌，很有好处。

`XE\t` 也可以模拟多层索引，例如：

```
{ XE "Style:as Word 对象" \t }
{ XE "Style:as 样式" \t }
{ XE "Style:as 样式:内建样式:目录 1-9" \t }
XE "Style:as 样式:内建样式:索引 1-9" \t
```

⁵ 我的意思是它会星罗棋布地改变文件结构，并且在又做了一些操作之后便难以全面回复文件原貌。

```
XE "Style:as 样式:内建样式:标题 1-9" \t
XE "Style:as 样式:内建样式" \t
XE "Style:as 样式:自定样式" \t
```

这便做出下面这样的索引：

```
style
  as Word 对象.
  as 样式.
  内建样式.
  目录 1-9.
  索引 1-9.
  标题 1-9. ←
  自定样式.
```

- 制作“索引词条文件”，一如 8.2.1 节所述。请注意，两个独立的索引项，如果其“主项目”或/和“次项目”的内容（包括空格）完全一致，会被 Word 视为同一笔项目，进而被合并为同一个索引词条。正因为如此，先前所说以 XE\t 模拟出来的索引词条，才有可能和“索引词条文件”中的 XE 设置合而为一。
- 以批量方式标记（mark）索引项目，一如 8.2 节所述。

有时候，为了呈现某种结果，必须来点小伎俩。例如，假设索引设计者希望读者无论寻找“标题 1”～“标题 9”的任何一个，都可以从上图的“内建样式 → 标题 1-9”所列页次中获得线索，那么也许必须插入所有“标题 1”、“标题 2”……“标题 9”的对应索引项，再将取得之页码整理于上述所列的“标题 1-9”词条。

- 点击【插入→索引和目录(D)】+索引(X)+确定，获得一份由 Word 为我们自动化生成的索引。
- 逐一修改每个词条的 XE 内容。请注意，也许您会想要按下 **Ctrl+Shift+F9** 切断索引的域关系，而后直接修改索引内容，然而我想最好还是保留域关系，因此应该修改的是 XE 设置内容。如果这样，这份索引便可重制、可更新（见 9.2.2 节）。但如果采用第一种做法，索引内容将成为“飘零的落花”，其成分从此只是一般文字，不复能够进行任何自动化工程。
- 先前当我们以批量方式针对某些词条插入 XE 索引项，讨厌的是，竟连出现于目录中的那些词条也被自动标记 XE 索引项。这些 XE 的出现，并非因为目录项的对应标题有了变化（固然那是可能的）而连带影响目录内容，而是因为目录内容受了“批量插入 XE”的影响。但是您知道，目录页码有什么必要被记录到索

引去呢？任何词条在目录中都只是“出现”，毫无解释或提供额外信息的可能。因此有必要去除那些XE设置。很简单，只要在目录区更新域（如9.2.2节）即可，这将使目录区内的XE索引项消失。至于“目录项所对应之标题被插入了XE索引项”，该XE索引项并不会被抓进目录区来（算它聪明◎）。

- 显示XE或不显示XE（利用 F3 钮），会影响整个文档的页码，进而影响索引区内所收集的页码。因此，如果您在显示XE之后，更新索引区内的域，您会发现整个索引区记录的页码全都改头换面了。别紧张，那只是反映当时现况罢了。显示XE时，文档画面可能会变得很糟糕，像这样：

```
#001-Dim[!-XE-"Dim·As(VBA指令)"]·!-startParag → '区块中的第一个段落(代表第一行程序代码)
#002-Dim[!-XE-"Dim·As(VBA指令)"]·!-endParag → '区块中的最后一个段落(代表最后一行程序代码)
#003 → '注：采前闭后开区间[],所以endParag其实是代表最后一行的下一段落
#004-Dim[!-XE-"Dim·As(VBA指令)"]·!-nLineNum → → → '行号(数值)
#005-Dim[!-XE-"Dim·As(VBA指令)"]·!-sLineNum As String[!-XE-"String(VBA型别)"]·! → '行号(文字)
#006-Dim[!-XE-"Dim·As(VBA指令)"]·!-i
#007-
#008-For [!-XE-"For·To(VBA指令)"]·!-startParag = 1 To ActiveDocument.Paragraphs[!-XE-"Paragraphs(Word对象)"]·!-Count
```

不过没关系，一旦隐藏XE，画面又会恢复原本的清爽，像这样：

```
#001 Dim startParag      '区块中的第一个段落(代表第一行程序代码)
#002 Dim endParag        '区块中的最后一个段落(代表最后一行程序代码)
#003      '注：采前闭后开区间[],所以endParag其实是代表最后一行的下一段落
#004 Dim nLineNum        '行号(数值)
#005 Dim sLineNum As String    '行号(文字)
#006 Dim i
#007
#008 For startParag = 1 To ActiveDocument.Paragraphs.Count
```

9. 域.....	235
9.1. 所谓域 (Fields)	235
9.2. 域的插入与更新.....	242
9.2.1. 插入域.....	242
9.2.2. 更新域.....	242
9.2.3. 显示或隐藏域代码.....	243
9.3. 本书使用的域.....	244
9.3.1. Page 域: 目前页次.....	244
9.3.2. Section 域: 目前节次.....	245
9.3.3. NumPages 域: 文档总页数	245
9.3.4. NumChars 域: 文档总字符数	245
9.3.5. NumWords 域: 文档总字数	246
9.3.6. TOC 域: 目录	246
9.3.7. TC 域: 目录项.....	246
9.3.8. Index 域: 索引.....	247
9.3.9. XE 域: 索引项.....	247
9.3.10. StyleRef 域.....	247
9.3.11. PageRef 域	247
9.3.12. Ref 域	248
9.3.13. Seq 域.....	248
9.4. StyleRef 域选项 (options)	249

9

域

9.1. 所谓域 (Fields)

域的目的之一是用来为文档发展过程中“内容可能有所变更”的资料保留位置¹。这些“内容可能有所变更”的资料包括目录、索引、页码、打印日期、储存日期、编辑时间、作者、文件名、文件大小、总字符数、总行数、总页数……

大型文档不可能没有用上一丁点儿域，为的便是希望能够获得这些“内容可能有所变动”的资料的最新状态，使文档创作者或排版者无需以人工方式完成繁琐的工作。让 Word 忙去吧！

域的代码、结果、底纹

域有点像 Microsoft Excel 中的“公式”。域代码就像 Excel 公式，域结果则类似公式所产生的数值。您当然希望在文档中看到域结果，那就必须先隐藏域代码，做法是点击【工具→选项(O)】+视图，取消勾选□域代码，如图 9-1。

¹ 另一个功能是在“合并打印文档”时为“套印信件和标签”保留位置。本书并不讨论这个用途。

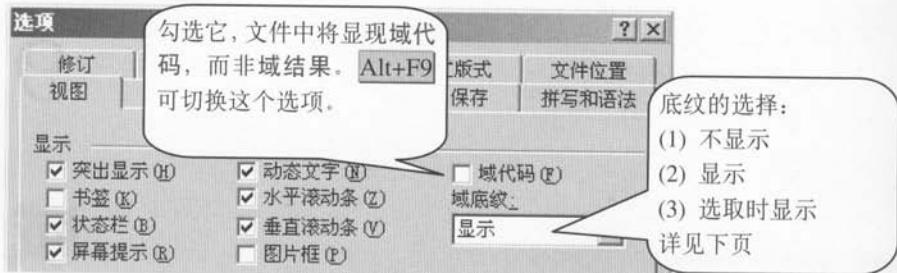


图9-1 选择显现域的代码或结果，以及底纹

如果您勾选了**域代码**，就会在文档中看到域代码，出现于一对大括弧{}之间。我试了一个例子，利用9.2.1节所说的**Ctrl+F9**，在文档中插入许多域，并勾选上图的**域代码**，得画面如下，其中的域代码意义请参考表9-1：

```
{
{ AUTHOR }
{ CREATEDATE }
{ DATE }
{ EDITTIME }
{ FILENAME }
{ FILESIZE }
{ NUMCHARS }
{ NUMPAGES }
{ NUMWORDS }
{ PRINTDATE }
{ REVNUM }
{ SAVEDATE }
{ SECTION }
{ SECTIONPAGES }
{ SET BookMarkBySet "Hello" }
{ SUBJECT }
{ SYMBOL 211 \F "Courier New" \s 16 } // ANSI 211号字符，16 pt.
{ TEMPLATE }
{ TIME }
{ TITLE }
{ USERADDRESS }
{ USERINITIALS }
{ USERNAME }
}
```

这些都是域代码

意义：设置一个书签名 BookMarkBySet，其所代表的文字内容是“Hello”

此行执行后，检验【插入→书签】，的确多出一个书签

字体 大小为 16pt

图9-2 域代码（部分）。不分大小写

一旦取消勾选上图的**域代码**，并更新所有域（办法如9.2.2节所言），获得

结果如下 (与上述的域一一对应) :

j.j.hou	// AUTHOR:作者
2004-6-23 17:35	// CREATEDATE:生成日期
2004-6-24	// DATE:日期
588	// EDITTIME:此次编辑时间(分钟)
lsword	// FILENAME:文件名称
12866560	// FILESIZE:文件大小
166604	// NUMCHARS:总字符数
362	// NUMPAGES:总页数
125015	// NUMWORDS:总字数
2004-6-24 10:23	// PRINTDATE:上次打印时间
44	// REVNUM:版本号码
2004-6-24 14:41	// SAVEDATE:上次保存时间
16	// SECTION:本节编号
13	// SECTIONPAGES:本节页数
三天内彻底理解 Word 排版和创作平台	// SET:(设置一个书签)
○	// SUBJECT:主题
lsword.dot	// SYMBOL:ANSI 字符
14:59	// TEMPLATE:模板文件名
《Word 排版艺术》	// TIME:时间
台湾 300 新竹市建中一路	// TITLE:标题
jjhous	// USERADDRESS:用户地址
Hou Jei	// USERINITIALS:用户名缩写
	// USERNAME:用户名

这些都是
功能变数结果

这些是我另外加
上的说明

图9-3 域结果

再举一个例子，用户可在页眉内点击【插入→域(F)】，放置一个 StyleRef 域，并在其对话框(图 4-2、图 4-3)中指定某个样式 A，于是置入域 [STYLEREF "A"]。将来当 Word 在页面视图中显示所有页面元素时，读到这个域，便将当页第一个²“以此样式格式化”的文字置入页眉。

如果想在文档中一眼看出某份资料是域的运行结果而非一般文档内容，惟一的办法是点击该份资料，看看是否出现所谓的域底纹(灰底；实际打印时不会出现)。

您也可以让域结果“无论如何不显示底纹”或是“不需被点击就自动显示底纹”。

² 也可以是最后一个，如果加上特殊选项的话。见 9.3.10 节和 9.4 节。

图 9-1 右下角提供了三种选择 (1) 选取时显示 (2) 显示 (3) 不显示。其中“显示”的意思就是：无论如何都显示底纹——这正是我个人喜欢的形式，可以在编排过程中有别于一般的文档内容。

顺带一提，图 9-3 的许多域运行结果也出现在以下三个 Word 对话框中。由于取图时间不同，以下各图的若干数据可能和图 9-3 略有差异。

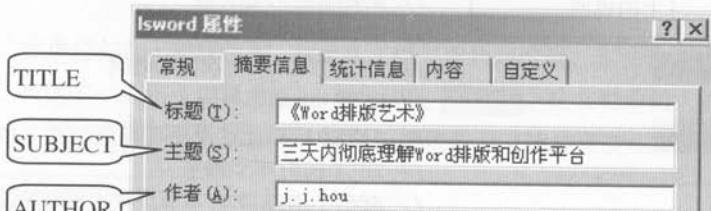


图9-4 【文件→属性(I)】 + 摘要信息

lsword 属性	
常规 摘要信息 统计信息 内容 自定义	
创建时间:	2004年6月24日 10:46:00
修改时间:	2004年6月24日 15:21:58
存取时间:	2004年6月24日
打印时间:	2004年5月25日 14:37:00
上次保存者:	Hou Jie
修订次数:	49
编辑时间总计:	85 分钟
NUMPAGES	统计名称 值 页数: 420
NUMWORDS	段落数: 6114
NUMCHARS	行数: 11380 字数: 154725 字符数: 213578 字符(带空格): 237401 字节数: 13624

图9-5 【文件→属性(I)】 + 统计信息

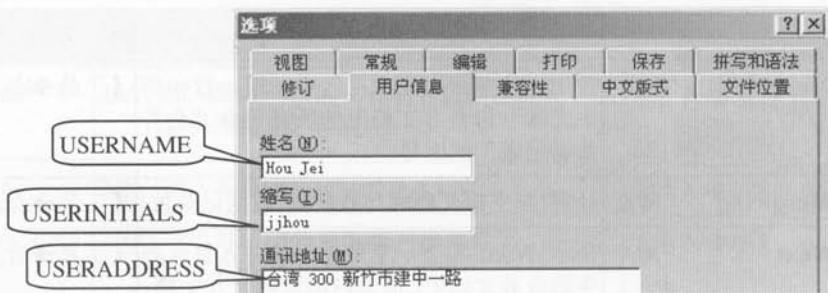


图9-6 【工具→选项(O)】+【用户信息】

Word 域总览

表 9-1 是我所收集到的 Word 域。为什么说“我收集到的”而没有使用更肯定的语气呢？因为不同的文献对于 Word 域的含括范围似乎有着不同的见解。此外某些域需要参数（选项，options），本书基于篇幅和定位之故，无法涵盖它们，但是它们非常重要，可带来许多功能上的弹性与变化。详请参阅《Word Online Help》，9.4 节则特别对于 StyleRef 域的参数做了完整说明。

表9-1 Word 域一览表

域代码	页次	意义	相关操作
= (Formula)	245	使用数学公式来计算数值	【表格→公式】
BarCode	N/A	“美国邮政所用机器可判读”之邮件条码。 用于邮件处理。	【工具→信封和标签】
Database	N/A	在 Word 表格中插入数据库查询结果	
Data	N/A	(Word 6.0+ 不再需要)	
Date	236	当天日期	【插入→日期和时间】
EditTime	236	本次编辑耗用时间 (分钟)	【文件→属性】+【统计信息】+【编辑时间总计】
FileSize	236	文件大小 (单位: bytes)	【插入→域】
Index	247	建立并插入索引。会收集 XE 域所指定的索引项	【插入→索引和目录】+【索引】
ListNum	N/A	与多层次样式 (亦即章节编号) 有关 (多层次样式见 5.11 节)	【插入→域】或 【工具→自定义】+【命令】+【类别: 插入】+【命令: 插入】 ListNum 域

域代码	页次	意义	相关操作
MergeRec	N/A	使用“合并打印主文档”中的域，打印所生之每一份合并文档上的所有符合“合并数据记录”的编号	【工具→合并文档】
MergeSeq	N/A	计算“与主控文档成功合并”之数据记录数	【工具→合并文档】
Next	N/A	指示 Word 将下一笔数据记录并入目前产生的合并文档中，而非启动新的合并文档。	【工具→合并文档】
NextIf	N/A	比较两个表达式。若结果正确，将下一笔数据记录并入目前的合并文档中。	【工具→合并文档】
NumChars	245	文档总字符数	【插入→域】
NumPages	245	文档总页数	【插入→域】
NumWords	246	文档总字数	【插入→域】
Page	244	当前页码	【插入→页码】
PageRef	247	“作为交叉引用之用”的书签的页码	【插入→交叉引用】 例如本表的所有页次都是 PageRef
Print	N/A	将“打印机控制字符”传送到选定的打印机。文档被打印时会发生影响。	
PrintDate	N/A	最近一次的文档打印日期和时间	【插入→域】
Private	N/A	储存由其它文件格式转换而来的文档资料。会成为隐藏文字，不影响版面。	
Quote	N/A	在文档中插入指定文字	举例：配合嵌套的 IF, =(Formula) 和 DATE，可产生上个月月份名称。
RD (Reference Document)	222	确认“引用文档”。引用文档就是建立目录、引文目录或“带有 TOC、TOA、INDEX”之索引时所含入的那个文档。	【插入→索引和目录】+索引+自动标记+打开索引自动标记文件
Ref	248	在文档中插入交叉引用	【插入→交叉引用】 例如请参考表 9-1 其中的表 9-1 就是 Ref
RevNum	236	文档版本编号。资料来源见右栏→说明，代表该文档的储存次数。	【文件→属性】+统计信息+修订次数

域代码	页次	意义	相关操作
SaveDate	236	插入最近一次的文件储存日期和时间 资料来源见右栏说明→	【文件→属性】+【统计信息】+【存取时间】
Section	245	插入本节编号	【插入→域】
SectionPages	236	将“目前此节总页数”插入某节。使用时应令各节由1开始重新编页	【插入→域】
Seq	248	依序为文件的章节、表、图和其它页面元素编号。若新增、删除或移动页面元素及其Seq域，宜强制更新其余Seq域以求反映新顺序。插入ListNum域也能自动编号。	【插入→题注】 例如题注表9-1中的‘9’是由【StyleRef 1】取得，‘1’则由【Seq】取得。
Set	236	定义书签及其所代表的文字	【插入→书签】
SkipIf	N/A	比较两个表达式。如果比较结果正确，便取消目前的合并文档，并移到数据源的下一笔数据项，开启一份新的合并文档。如果比较结果错误，继续目前的合并文档。	
StyleRef	247	插入特定样式所含之文字内容	【插入→域】 完整参数请参考9.4节
Subject	236	插入主题。资料来源见右栏说明→	【文件→属性】+【摘要信息】+【主题(S)】
Symbol	236	插入字符或字符串（使用ANSI字符集）	【插入→符号】
TA	N/A	定义引文目录项的文字和页码	
TC	246	定义文字和页码，使成为章节目录和图表目录内的目录项。	【插入→域】
Template	236	插入模板文件名。资料来源见右栏说明→	【文件→属性】+【摘要信息】+【模板】
Time	236	插入目前时间	【插入→日期和时间】
Title	236	插入指定之标题（标题将以文字参数的型式指定）。此标题可被右栏引用→	【文件→属性】+【摘要信息】+【标题】
TOA	N/A	建立并插入引文目录。它会集结TA所标示之索引项。	【插入→索引和目录】+【索引】

域代码	页次	意义	相关操作
TOC	246	建立并插入目录。Word 将以多级列表、指定样式、TC 目录项三者集结成为目录。	【插入→索引和目录】+ 【目录】
UserAddress	236	插入使用者地址。资料来源见右栏说明 →	【工具→选项】+ 【用户信息】+【通讯地址】
UserInitials	236	插入使用者姓名缩写。资料来源见右栏 →	【工具→选项】+ 【用户信息】+【缩写】
UserName	236	插入使用者姓名。资料来源见右栏说明 →	【工具→选项】+ 【用户信息】+【姓名】
XE (indeX Entry)	247	定义索引项的文字和页码。该索引项会被纳入索引之中。	【插入→索引和目录】+ 【索引】+【标记索引项】

9.2. 域的插入与更新

上一节已经告诉您域的用途和细部（代码、结果、底纹）。这一节将告诉您域的实际操作：如何插入、如何更新、如何显示或隐藏代码。

9.2.1. 插入域

我们无法利用键盘直接键入图 9-2 的域大括弧。必须按下 **Ctrl+F9**，才能在输入点插入一个域大括弧（此时输入点将停在大括弧内），然后才能在域大括弧之间键入适当内容。如果点击特殊命令，例如【插入→日期和时间(T)】或【插入→域(F)】（图 9-7），也能够在文档内插入域。

9.2.2. 更新域

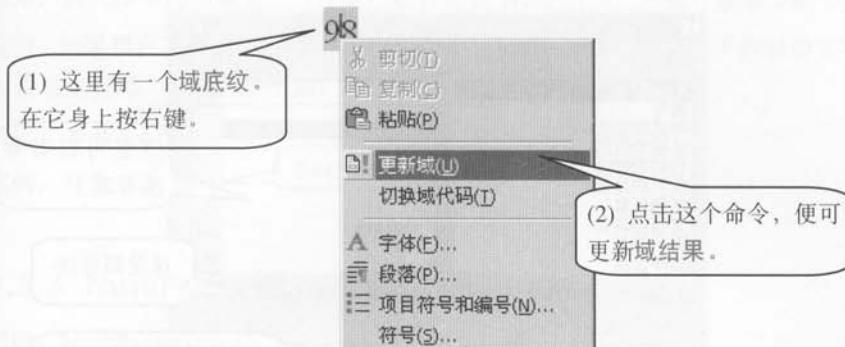
域既然是用来为文档发展过程之中“内容可能有所变更”的资料保留位置，显然需要不时地更新。是的，我们必须手动更新域，才能显现最新内容。

首先是圈选欲更新的域：

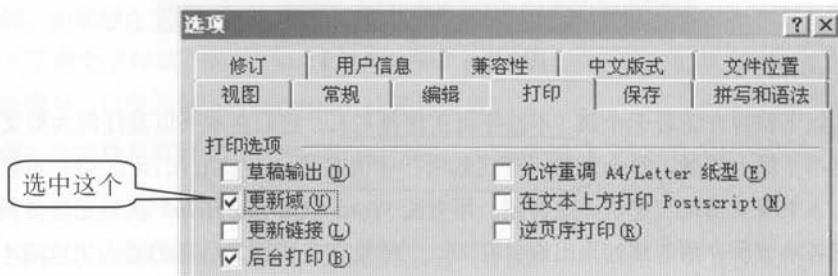
- 一次圈选所有域。这当然是最方便的了，稍后更新时毕其功于一役。而且 Word 更新所有域（包括目录和索引）的反应速度不恶，不会让您等太久。圈选方式是：点击【编辑→全选(L)】。

- 一次只选一个域。圈选方式是：以鼠标点击你的标的物。

圈选之后接下来是更新动作：直接按 F9 便可。亦可从域³的【右键菜单】中点击更新域(U)：



如果您担心自己一时疏忽忘记更新域结果，导致错误的文档内容，因此希望将来打印之前 Word 能够自动更新每一个域，可点击【工具→选项(O)】+ 打印，勾选其中的 更新域(U)：



9.2.3. 显示或隐藏域代码

如果想要显示或隐藏某个域代码，请点击该域代码或域结果（视当时显现什么而定），然后按 Shift+F9（或【右键菜单】中的切换域代码），即可在代码和结果之间切换。如果要显示或隐藏文档中的所有域代码，请按 Alt+F9。

³ 如果先前做的是全选动作，那么此处选择任何一个域的【右键菜单】皆可。

9.3. 本书使用的域

Word 支持的域很多，您可以观察表 9-1，或是从【插入→域(F)】所得的对话框（图 9-7）中获得全貌。图 9-7 是绝大多数域的插入途径。

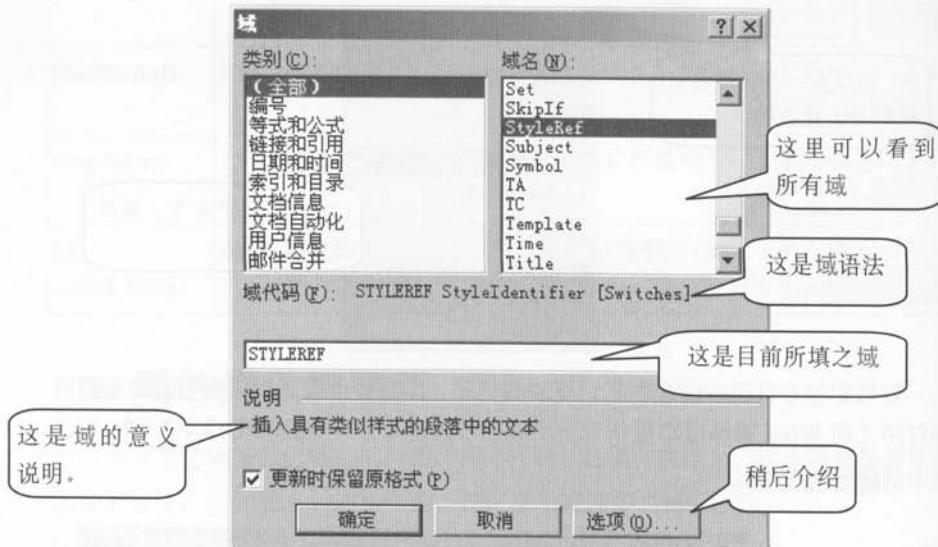


图9-7 【插入→域】

以下我将介绍若干个域（不按字母次序排列），它们在本书以及任何大型文档中免不了派上用场。每个域的“格式选项”（诸如 `\l`, `\n`, `\p`, `\r`, `\t`, `\w` 之类）非常繁多，本书限于篇幅与定位无法详述，可查阅 Word《On-line Help》获得完整资料。以下各小节所介绍的域如果出现“实例：”，便是我在冒号之后真的插入了该域（前带若干说明文字），读者可从实际结果获得一些体会。

9.3.1. Page 域：目前页次

代码：PAGE

形式：{ PAGE [* 格式参数] }

功用：Word 会在自动分页过程中将页码安插于此域内。只要点击【插入→页码(U)】，或点击页眉和页脚工具栏（图 3-6）上的页码，Word 就会为您插入这一域。

参见：本书 4.3 节

实例：目前页次 244

Page 域的结果

9.3.2. Section 域：目前节次

代码: SECTION

形式: { SECTION }

功用: 插入本节的节编号。(注意: 这里是指 Word 的节, 不是一般章节的节)

范例: 如果想在文档的每一页打印“第 n 节之第 m 页”字样, 请将下列域和文字插入页眉或页脚: 第 {SECTION} 节之第 {PAGE} 页

参见: 本书 3.4 节

实例: 目前节次 28

Section 域的结果

9.3.3. NumPages 域：文档总页数

代码: NumPages

形式: { NUMPAGES }

功用: 将总页数插入文档中。总页数也出现在【文件→属性(I)】所得之对话框中的统计信息内(图 9-5)。

范例: 如果想在文档每一页打印“第 12 页共 45 页”之类的话语, 可在页眉或页脚插入下列文字和域: 第 { PAGE } 页共 { NUMPAGES } 页。如果页码不是从 1 开始编号, 可使用域 = (Formula) 计算总页数。

实例: 本文档总页数 420

NumPages 域结果

9.3.4. NumChars 域：文档总字符数

代码: NumChars

形式: { NUMCHARS }

功用: 将总字符数插入文档内。这份信息也出现在【文件→属性】所得之对话框中的统计信息内(图 9-5)。

范例: 运用域 = (Formula), 透过“字数除以字符数”便可计算出每个字的平均长度(字符数): { ={ NUMCHARS } / { NUMWORDS } }

实例: 本文档总字符数 214991, 总字数 155989, 每个字的平均长度(字符数)为 1.38

NumChars 域结果

NumWords 域结果

9.3.5. NumWords 域：文档总字数

代码: NumWords

形式: { NUMWORDS }

功用: 将总字数插入文档内。这份信息也出现在【文件→属性(I)】的
统计信息内(图 9-5)。

范例: 同 NumChars

实例: 同 NumChars

9.3.6. TOC 域：目录

代码: TOC

形式: { TOC [选项] }

功用: 建立并插入目录。此一域抽取**标题 1**、**标题 2**、**标题 3**样式或指定样式，或
TC 域之目录项，集合成为目录。只要点击【插入→索引和目录(D)】，Word
就会为您插入 TOC 域。

备注: 如果 TOC 域所建立的目录会影响文档编页，您需要再次更新域，以反映正
确的页码。

参见: 本书 7.1 节。本书共有四个目录，章节目录的域是 { Toc \o "1-5" }，
图目录的域是 { Toc \c "图" }，表目录的域是 { Toc \c "表" }，VBA
程序目录的域是 { Toc \c "VBA" }。

9.3.7. TC 域：目录项

代码: TC

形式: { TC "文字" [选项] }

功用: 允许在文档任何地点放置“日后可被 Word 收集成为目录之一部分”的文字。
可在**标题 1**、**标题 2**、**标题 3**样式或指定样式之外，辅助制作目录内容。

备注: TC 域会被格式化成为隐藏文字，而且不会在文档中显示其域结果。如果要
查看这个域，请点击**显示/隐藏**钮。

参见: 本书 7.2 节

9.3.8. Index 域：索引

代码: INDEX

形式: { INDEX [选项] }

功用: 建立并插入索引。INDEX 域会以 XE 域为对象, 收集所有的索引项。只要点击【插入→索引和目录(D)】 , Word 就会为您插入 INDEX 域。

参见: 本书 8.2 节

9.3.9. XE 域：索引项目

代码: XE

形式: { XE "文字" [选项] }

功用: 定义索引项文字(显示于索引的单层或多层文字)和页码。所有经过 XE 域定义过的文字(词条), 都会被收集到以 INDEX 域制作出来的索引中。XE 域会被格式化成为隐藏的文字, 而且不会在文档中显示其域结果。如果要检视这个域, 请点击 **显示/隐藏** 钮。

参见: 8.1.1、8.2 节、8.3 节

9.3.10. StyleRef 域

代码: STYLEREF

形式: { STYLEREF StyleIdentifier [选项] }

功用: 此一域可抓取特定(指定)之某种样式的文字内容, 放在域所在位置上。如果 STYLEREF 域被插入页眉或页脚之中, 它会取得当页的第一个(或最后一个; 可选)指定样式的文字。

参见: 4.2 节和 9.4 节

9.3.11. PageRef 域

代码: PAGEREF

形式: { PAGEREF 书签 [* 格式参数] }

功用: 插入“作为交叉引用之用”的书签的页码。

说明：下图说明 PageRef、Ref、Seq 三个域的运用场合。

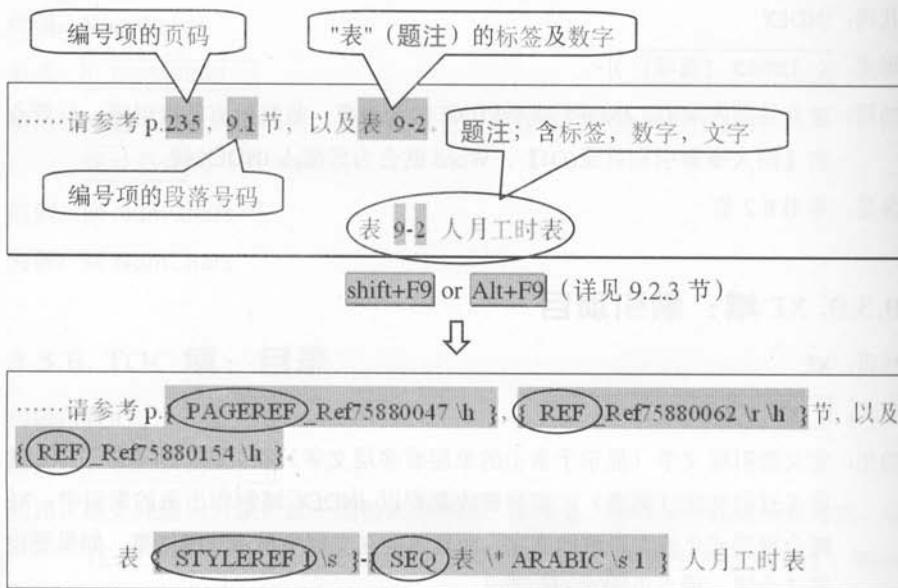


图9-8 域 PageRef、Ref、Seq 的用途

9.3.12. Ref 域

代码：REF

形式：{ REF 书签 [选项] }

功用：在文档中插入交叉引用

参见：图 9-8

9.3.13. Seq 域

代码：SEQ

形式：{ SEQ 名称 [书签] [选项] }

功用：依序为文档中的章节、表、图、以及其他页面元素编号。若新增、删除或移动页面元素及其 Seq 域，宜强制更新其余 Seq 域以求反映新顺序。

参见：图 9-8

9.4. StyleRef 域选项 (options)

Word 域的选项太多了，而且并非本书焦点，所以我不打算详细说明它们。但是举几个实例使您获得一些具体印象还是很有必要。以下我将（只）介绍本书编排过程中曾经试过的 StyleRef 域（9.3.10 节）的两个选项。

点击【插入→域(F)】并选择 StyleRef (如图 4-2)，点击【选项(O)】之后，在【域选项+域专用开关(P)】中有 6 个选项，如图 9-9。

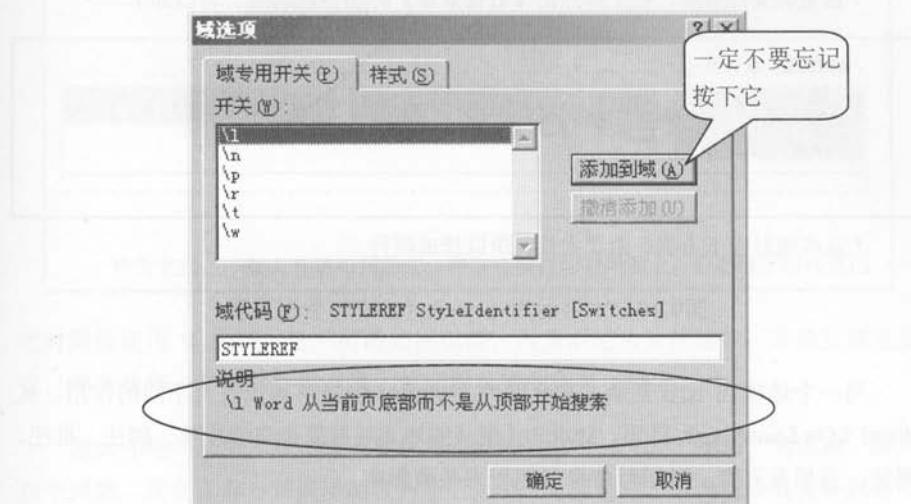


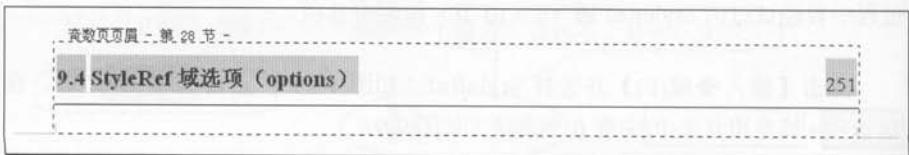
图9-9 域 StyleRef 的 6 个选项

域 StyleRef 会以“指定的方式”抓取“指定样式”所套的内容。我曾经试过 \l 和 \r，下面是两者的意义：

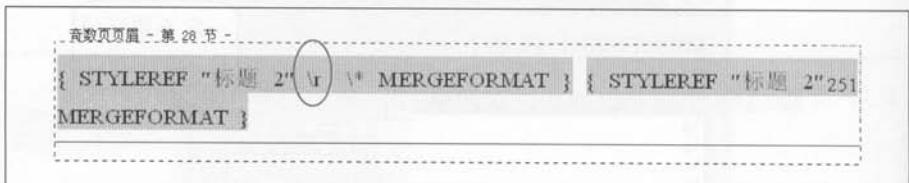
表9-2 域 StyleRef 的 \l, \r 选项意义

选项	意义	说明	相关讨论
\l	由本页下方往上搜寻，再由本页上方往下搜寻	插入本页最后一个（而非第一个）以指定样式进行格式化的文字。Word 会从尾到头（而非默认的从头到尾）搜寻出现于该页的样式。	4.2.1 节
\r	插入被引用段落的段落编号	多层次样式（章节编号）是由 Word 自动产生的，需以 \r 参数才能抓到。	5.11.3 节 (p.168)

\r 选项比较单纯，我先来谈它。5.11 节谈到多层次样式时，最后总结 (p.168) 提到：页眉区的 StyleRef 只能取得标题文字，取不到编号。但如果我们再加一个 StyleRef 并添加 \r 选项如下，就可以取得完整的信息。



↑ 这是我要的效果。在上面点击【右键菜单】的切换域代码，得图如下



↑ 这些域对应于上图。由于太长，所以挤成两行

图9-10 本书运用 StyleRef/r 于奇数页页眉

另一个选项 \l 比较复杂，它会因为 StyleRef 的坐落地点而有不同的作用。从 Word《On Line Help》得知，StyleRef 坐落的地点（可能是文档段落、脚注、批注、尾注、页眉及页脚）对运行结果可能带来些微影响。

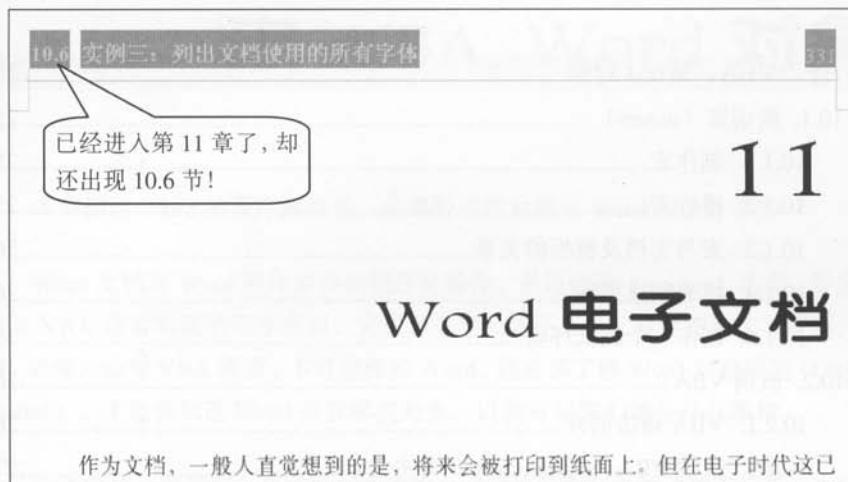
Word《On Line Help》内曾经举出 \l 选项的一个实用例子。假设我们打算打印会员通讯簿中每一页的第一个及最后一个会员姓氏，可以先将某种字符样式套用于每位会员姓氏身上，然后在页眉区中插入一般（无参数）的 StyleRef，后面再加一个带有 \l 选项的 StyleRef：

```
{ StyleRef "Name" }-{ StyleRef "Name"\l }
```

所得便是诸如 Anders-Bendal 这样的结果。这种形式的信息也可以运用在索引的页眉身上——虽然似乎有点华而不实，毕竟从“下一页页眉所记录的起始索引词条”便可轻易判断本页的索引词条的大致结束范围。

由于 \l 会改变 Word 对指定样式的搜寻方向和搜寻起点，也许您直观想到，可以拿它来解决页眉中的大标题（本书为**标题 2**）所引发的问题。这个问题曾在 4.2.1 节讨论过，起因于我们难以控制“当某页没有出现大标题（**标题 2**）”时的 StyleRef

表现。依我的想法，在某一章中，如果某页有大标题，那么页眉区就取该页的第一个出现的大标题；如果某页没有大标题，那么页眉区就沿用前一页页眉所用的大标题。这是很合理的安排，但像（例如）本书第 6,11,12 章一开始并不立即有大标题，如果这时候还沿用前一个大标题就不对了：



此时即使使用 VI 选项，也不可能如愿以偿，因为不论从头往尾找，或是从尾往头找，都会找到一个标的物填进来，然而我希望的却是“空”。

怎么办呢？从前当我还没开始使用“自动化章节（大纲）编号”功能时，碰到这个问题，我会在前一章最尾端放一个“大节”对应样式，并刻意令其文字内容为空。这个空白大节标题便被下一章的页眉区抓去，达成我的目的。

但是现在我开始使用“自动化章节（大纲）编号”，如果还沿用上述手法，那个空白大节会出现编号，这就破坏了画面，也破坏了目录。

只好再祭出肮脏招数：于制版过程中修改。但如果是直接打印机输出，我真的不知道怎么办了⑧

10. 宏、VBA、Word 对象	253
10.1. 所谓宏 (macro)	253
10.1.1. 制作宏.....	254
10.1.2. 播放宏.....	259
10.1.3. 宏与文档及模板的关系.....	260
10.1.4. 将宏制成按钮.....	261
10.1.5. 制作一个测试环境.....	264
10.2. 所谓 VBA	265
10.2.1. VBA 语法简介	265
10.2.2. 自学 VBA Programming 的好办法	273
10.3. Word 对象 (Objects)	275
10.3.1. 面向对象 (Object Oriented)	275
10.3.2. Word 对象模型 (Object Model)	279
10.3.3. 重要而常用的 Word 对象	281
10.3.4. 自学 Word Programming 的好办法	305
10.4. 实例一：程序代码自动插入行号	305
10.5. 实例二：两岸术语转换.....	311
10.6. 实例三：列出文档使用的所有字体	320

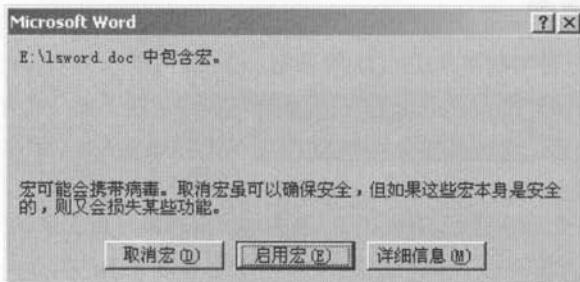
10

宏、VBA、Word 对象

本章探讨 Word 的程序操控性，并某种程度地揭示 Word 的设计架构。

Word 文档对 Word 软件本身的程序化操作，系透过宏（macros）达成。宏是一种以 VBA 语言写就的程序模块，完成后可被关联至某个工具栏按钮，方便用户使用。单纯只会写 VBA 语言，不可能操控 Word，还必须了解 Word 对象模型（**Object Model**），才能够知道 Word 开放哪些对象，以及可用它们做些什么事情。

由于宏是一种程序，有些居心叵测的人以此性质制作出所谓的宏病毒。为防堵宏病毒，Word 只要侦测出文档中有宏将被载入，便发出以下警告：



这样的讯息往往吓坏用户，导致宏恐慌症，将宏全体判了死刑。劣币驱逐了良币！不过，自己写的宏（一如本章教给您的技术）那就放心用吧。

宏可被储存（贴附）于文档、文档模板或 Normal.dot 身上（详 5.2.1 节和 10.1.3 节）。惟有储存于文档和文档模板身上的宏，才会在即将被载入时引发以上对话框。

10.1. 所谓宏 (macro)

虽然 Word 有很好的功能帮我们处理繁重的编撰或排版任务，也许我们仍不时

需要在文档中重复琐屑无趣的动作，例如一再选择!**程序代码**，为每一行程序代码加行号¹，或一再以某些固定对应的词汇对文档内容进行转换²。如果能够将这些重复的劳动记录下来，集成“单一命令”，就可以节省许多人力、时间和精神。

上述所谓的“单一命令”，就是宏（macros），也就是“一连串 Word 指令和操作”的集合。下面是宏的若干典型用途：

- 加速常规的（规律性的）编辑动作和格式化动作
- 结合多个指令（命令）
- 使对话框中的选项更易使用
- 将一连串复杂的作业自动化

Word 可将一系列 Word 指令以 VBA³语言录制成宏。稍后我会谈到宏的录制与播放细节。

宏的生命随着模板或文档而走。您可以在任何模板或文档中储存宏。详见 5.2 节和 10.1.3 节。

10.1.1. 制作宏

Word 提供三种宏制作方式：(1) 宏录制，(2) 宏创建，(3) Visual Basic 编辑器。是的，您可以将“对 Word 实施的某一段操作过程”录制下来，成为一个宏，无需撰写任何程序代码。也可以开启一个空白宏，在其中添加 VBA 程序代码。

点击【工具→宏(M)】可得图 10-1。其中的宏录制器可帮您录制一个宏，Visual Basic 编辑器可为您开启一个录制妥当的宏，让您修改其中的指令，并加入某些无法录制的 VBA 指令，使宏更富弹性，功能更强大。

¹ 带有行号的程序代码，对阅读或解说或做心得笔记，都很有帮助。

² 例如，海峡两岸在各个领域，尤其是计算机领域，存在数不清的术语差异。

³ Visual Basic for Applications，一种与 Visual Basic 同宗的编程语言。

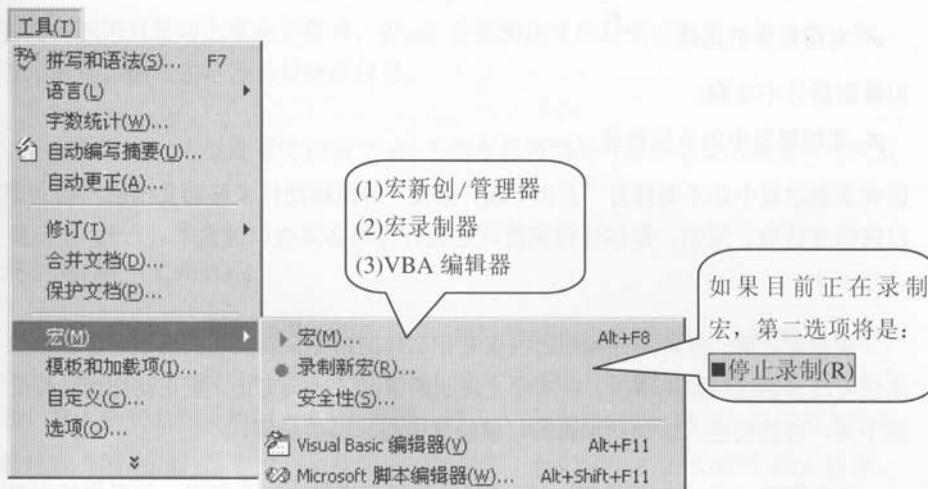


图10-1 【工具→宏(M)】

宏的录制 (Record)

点击图 10-1 的●录制新宏(R), 便启动宏录制器, 开始录制往后的一连串 Word 操作。此时的鼠标光标会变成飞鼠形状, 而图 10-1 的●录制新宏(R)会变成■停止录制(R)等待被点击以结束录制。录制过程可暂停, 而后于停止处继续录制。

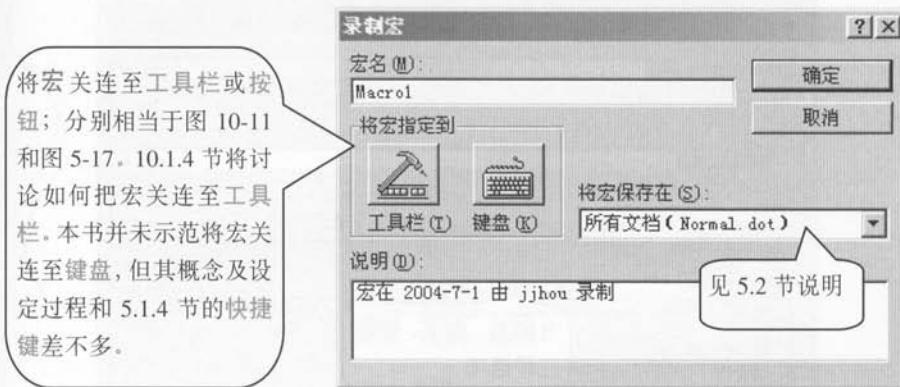


图10-2 宏录制器

录制过程中, 可使用鼠标点击:

- ✓ 菜单上的命令

✓ 对话框中的选项

但录制器并不录制：

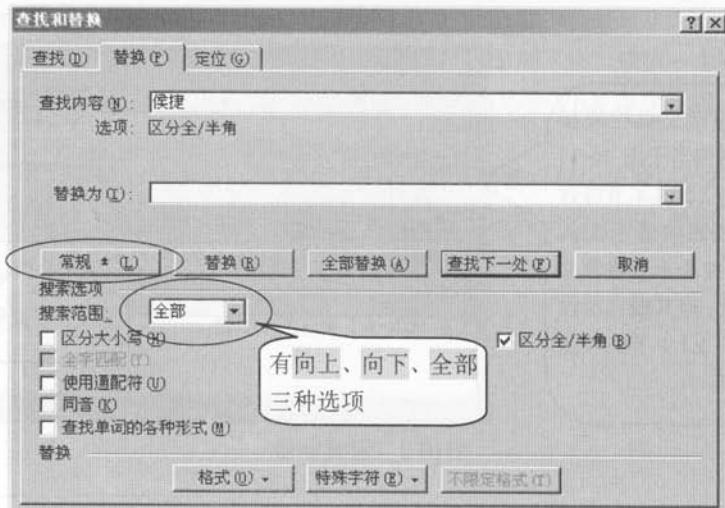
✗ 文档视窗中的鼠标轨迹

因此录制过程中您不能借着“点击”或“拖曳”等鼠标动作来移动文档插入点，或以鼠标来选取、复制、搬移任何东西。这些行为都必须改以键盘来进行⁴，才会被录制下来。

录制宏之前，应该妥善规划宏内该有什么样的指令，以及它们的执行步骤。如果您发现录制过程发生错误，在未停止录制的情况下回头更正，更正动作也会被录制下来。当然您也可以稍后编辑宏，移除其中的非必要步骤。

录制宏前，请试着预期 Word 可能显示的任何讯息。举个例子，如果您的宏内含【文件→关闭(C)】命令，而您的文档中有一些尚未储存的变更，Word 会要求您先储存文档，也就是会要求输入文件名。然而通常我们希望宏愈自动化愈好，非必要不与用户交互。因此如果想要避免上述讯息，录制时请在进行【文件→关闭(C)】前先进行【文件→保存(S)】。

又例如，如果宏内含【编辑→查找(F)】或【编辑→替换(E)】命令，请让它的搜寻范围遍及全部（如果这的确是您所希望的）：



⁴ 这实在是有一点儿小小的为难。谁还（会）用键盘来操作视窗程序呢？

因为，如果只是向上或向下搜寻，Word 会在到达文档起始或结尾时暂时中止宏，显示信息，询问您是否希望继续搜寻。

总而言之，请尽量避免宏和 Word 之间有任何问答（除非您要的就是一个交互式宏）。

宏的新建 (Create)

点击图 10-1 的 **▶ 宏(M)** 可得图 10-3。如果先前存在有任何宏，会被列于中央方块上，其中第一个宏名称并且被列于图上的文字方块内。现在，请清除文字方块内容，填入新宏名称，然后点击 **创建(C)**，进入 Visual Basic 编辑器，其中的文字编辑视窗（图 10-4）呈现出一个极简单的宏骨干，我们可以在上面撰写 VBA 程序。

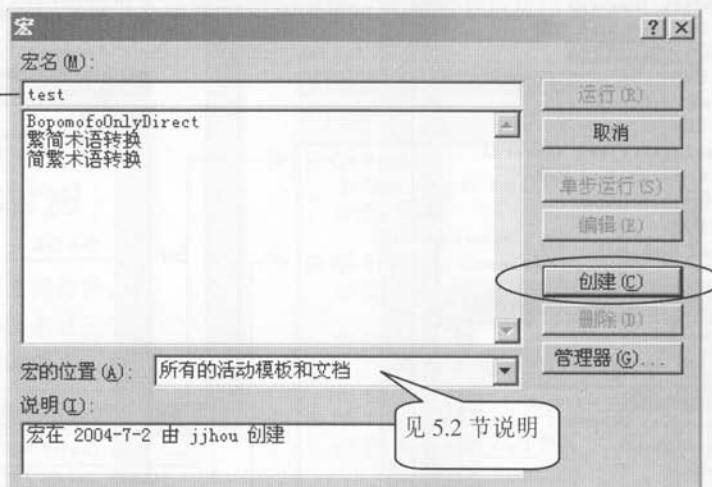


图 10-3 新建一个宏

```
Sub test()
    ' test Macro
    ' 宏在 2004-7-2 由 jjhou 创建
End Sub
```

图 10-4 VBA 程序编辑视窗

Visual Basic 编辑器

点击图 10-1 的 Visual Basic 编辑器(V)，便可在 VBA 编辑器（图 10-5）中输入 VBA 程序代码，建立宏。

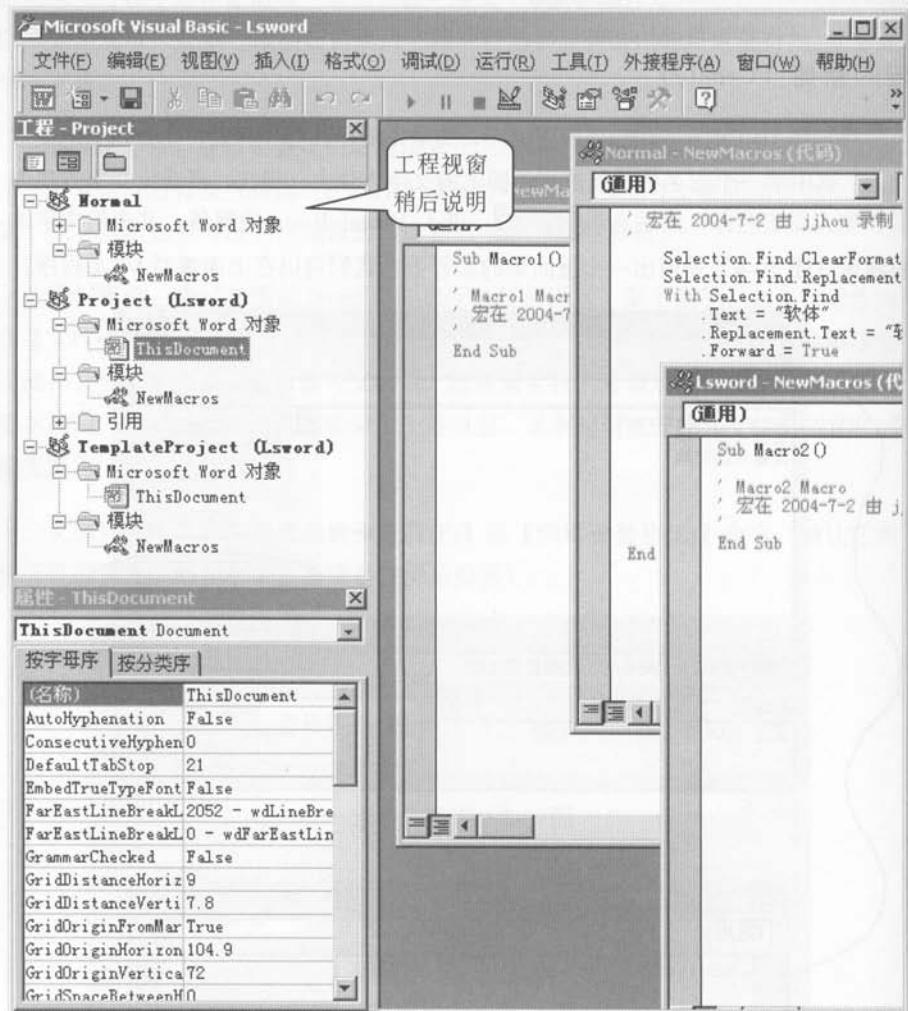


图 10-5 VBA 编辑器

宏、模块、工程、文档、模板

VBA 编辑器把宏(macros)、模块(modules)、工程.projects)、文档(documents)、模板(templates)之间的关系规划如下：

- 宏可独立命名。多个宏可储存于一个模块之内。
- 模块可独立命名。多个模块可储存于一个工程(project)内。
- 工程(project)可以是文档，也可以是模板。
- 文档和模板的关系如 5.2 节所述。

图 10-6 是 VBA 编辑器的工程管理视窗。随着图最左所示的选择不同(储存于文档或文档模板或共同模板 Normal.dot)，宏将被放进图 10-6 的三个可能地点。

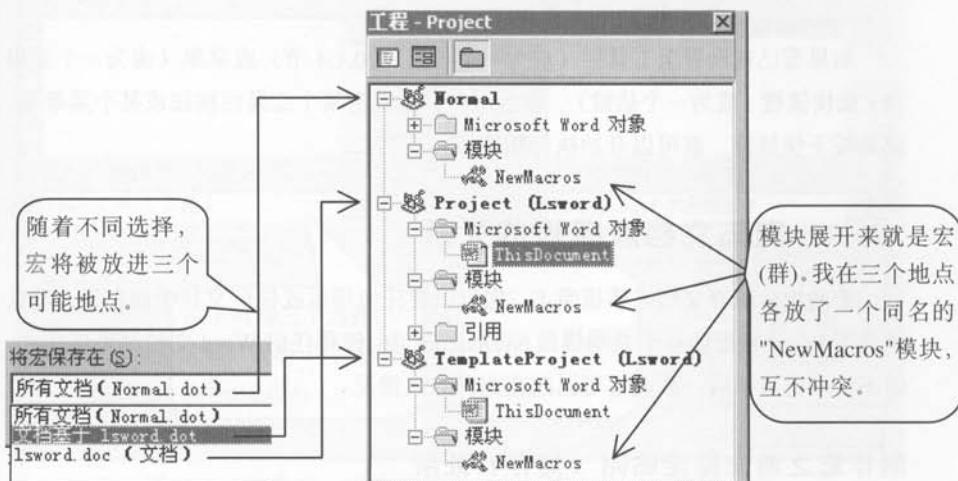


图 10-6 宏可被置于三个地方 (共同模板、文档模板、文档)

10.1.2. 播放宏

最基本 (但不方便) 的宏播放方式是：点击【工具→宏(M)→宏(M)】，再从图 10-7 对话框中点击您希望执行的宏。

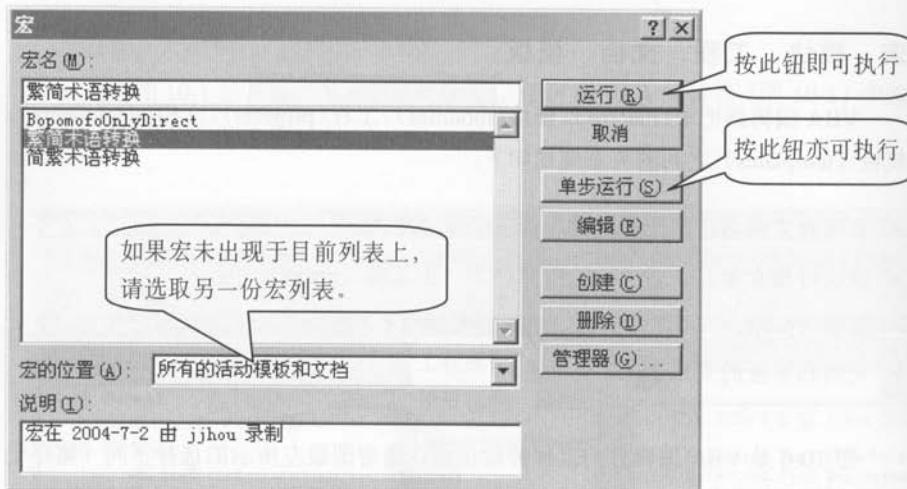


图10-7 执行某个宏

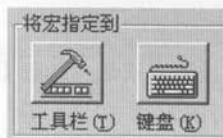
如果宏已被设置至工具栏（成为一个按钮，10.1.4 节）或菜单（成为一个菜单项）或快捷键（成为一个热键），那么用户只需点击某个工具栏按钮或某个菜单项，或是按下快捷键，就可以开始执行相关宏。

10.1.3. 宏与文档及模板的关系

宏的生命随着文档或模板而走。您可以在任何模板或任何文档中储存宏。默认情况下 Word 将宏储存于共同模板 Normal.dot 内，使得任何 Word 文档都可使用之。但不论事前或事后，我们都可以改变这种默认情况。

制作宏之前先指定贴附（储存）处所

宏如果不是录制而成，就是新建产生。这两种情况都可以事前指定宏的储存场所。当您准备开始录制一个宏时，图 10-2 中央有以下两个按钮：



点击左边按钮，表示要将宏依附于某个工具栏按钮身上，会出现图 10-11。图 10-11 允许用户指定宏的栖身场所 (.dot)。

如果点击的是右边按钮，会出现图 5-17。图中亦可指定宏的栖身场所 (.dot)。

宏制作之后改变贴附处所（管理器）

如果储存于公用模板 (normal.dot) 内的某个宏，其实仅仅适用于某特定类型的文档，您可以将此宏复制到该文档所属模板内，并删除 Normal.dot 中的该宏。欲复制、删除、重新命名宏，请使用管理器。点击【工具→宏→宏(M)】，再点击管理器(G)便可获得图 10-8。图中左右两清单分别列出所选之宏储存场所中的所有宏，再互以中央区的三个按钮`复制`、`删除`、`重命名`进行管理。5.2.5 节对于管理器也有一些说明。

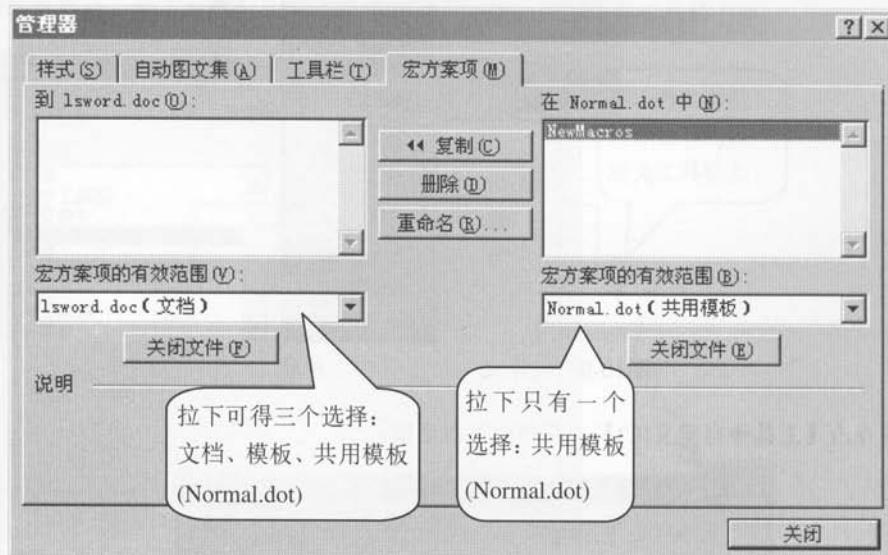


图10-8 管理器。可管理样式、自动图文集、工具栏、宏

10.1.4. 将宏制成按钮

如果经常使用某个宏，为求方便，可将它指定为一个工具栏按钮、菜单项或快捷键。这么一来就可以直接执行该宏而无需开启【工具→宏→宏(M)】对话框。下面是步骤：

1. 制作一个自定义工具栏。
2. 将宏制作成自定义工具栏中的一个按钮。虽然宏可贴附（储存）于文档 (.doc) 或模板 (.dot)，但只有贴附（储存）于模板的宏才能设置自定义工具栏按钮。

假设我已做好两个宏：“繁简术语转换”和“简繁术语转换”（详见 10.4.3 节），以下示范如何将它们制作成工具栏按钮。首先点击【视图→工具栏→自定义(C)】：



图10-9 自定义一个工具栏，步骤 1

或点击【工具→自定义(C)】，获得以下对话框：

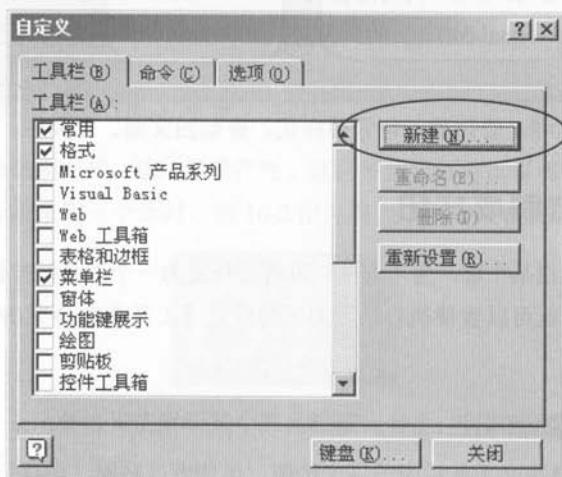
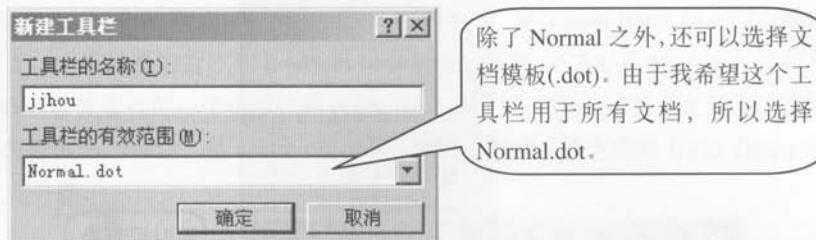


图10-10 自定义一个工具栏，步骤 2

点击图 10-10 右上角的【新建(N)】，获得以下对话框：



填入工具栏名称后，点击【确定】钮，于是屏幕上出现一个空白工具栏 ，位置可任意拖放。接下来将制作好的宏放到这个 "jjhou" 自定义工具栏上：

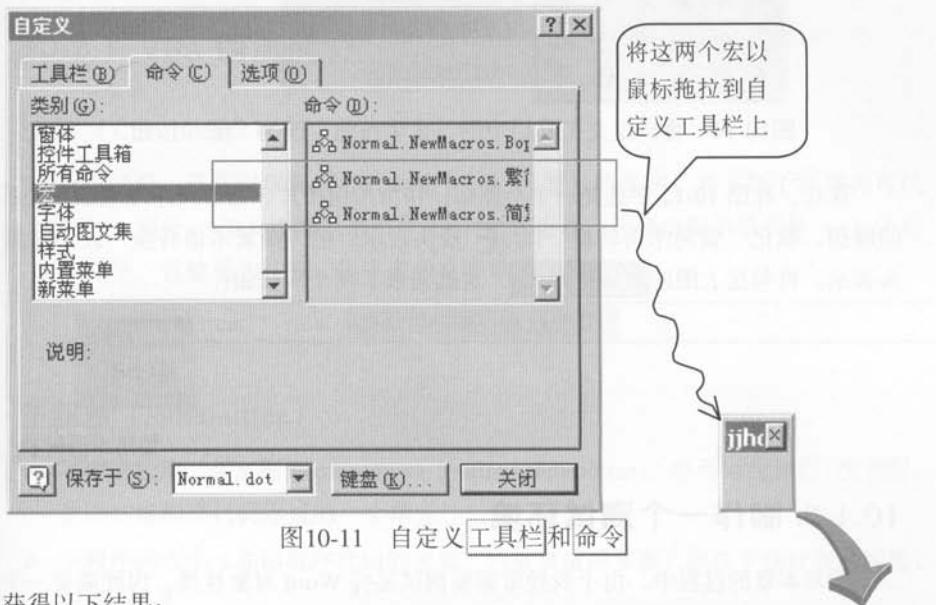
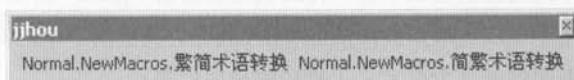


图10-11 自定义工具栏和命令

获得以下结果：



"jjhou" 工具栏上出现了两个按钮，但它们都是文字画面，没有明显的按钮形式。现在我打算将它们改为图片按钮。首先在任一个按钮上按鼠标右键，出现【右键菜单】如图 10-9 右半部，点击其中的【自定义(C)】，重新获得【自定义】对话框如图 10-11。然后再次以鼠标右键点击 "jjhou" 工具栏上的按钮，获得【右键菜单】如下图（和刚才的第一次【右键菜单】不同）：

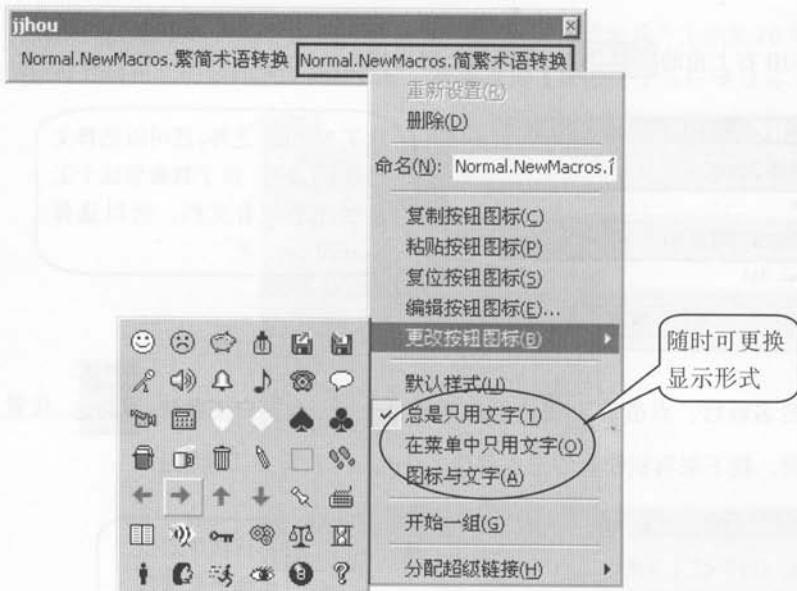
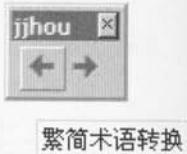


图 10-12 将自定义工具栏中的“文字按钮”改为“图片按钮”

现在，在图 10-12 中选择一个小图标，作为按钮图片。基于两岸地理位置关系的联想，我把“繁简术语转换”以 \leftarrow 箭头表示，把“简繁术语转换”以 \rightarrow 箭头表示。再勾选上图的默认样式(U)，这就完成了两个宏按钮：



10.1.5. 制作一个测试环境

撰写本章的过程中，由于我经常需要测试某些 Word 对象性质，因此需要一个单纯易用的测试平台。为此我在上述的“jjhou”自定工具栏中添加了一个按钮，适用于 10.3.3 节那些简短的 VBA 程序。我也为本章的其他宏（空壳）都先设好按钮。下面就是这个自定义工具栏的最后结果：



10.2. 所谓 VBA

VBA 的全名是 Visual Basic for Application，是 Visual Basic 的一门血缘极近的旁支。由于血缘极近，两者相貌酷似也就可以想象。VBA 是 Microsoft 公司用于其 Office 软件套件的一个语言，可供用户撰写宏，对 Office 进行二次开发。这种二次开发的能力和弹性，是 Microsoft Office 远胜于其他（缺乏宏能力的）Office 软件的一大关键。

这一节为您简介 VBA。我尽量选择本章末尾三个实例用到的指令或 Word 组件加以说明。本书并非为编程人员（programmers）而写，作为本书读者的您很可能从来没写过程序。我会尽可能让您觉得，写程序是件很有趣的事。

10.2.1. VBA 语法简介

注释（Comments）

- 所谓注释，是用以解释程序代码意义（给人看）的文字，并不属于程序内可执行的一部分。VBA 注释以单引号（'）形式出现，只要出现于单引号（'）之后的文字，皆被视为注释。以下灰色文字就是注释：

```
Square = num * num ' 将数值自乘，就是平方值
```

子程序（SubRoutine）

- 所谓子程序，英文称为 procedures, routines, subroutines，也可译为例程/次例程，是由一堆程序代码形成的一个单元。
- 子程序的大小（亦即程序代码的多寡、所做事情的多寡）取决于设计者的经验、习惯……⁵
- 子程序可以无限次调用其他子程序，也可以被其他子程序无限次调用；调用者和被调用者（在程序中）的出现次序无所谓。

⁵ 通常我们把“完整做一件事”的整组独立动作设计为一个子程序。但是“完整一件事”的规模有大有小，并无放之四海皆准的规则。也许有人认为“找出文件中所有蓝色字”是件完整的事，有人认为“找出文件中所有蓝色字并全部改为红色，而且改用粗体、字号加一倍”才是一件完整的事。所以，没有定论！程序设计就某种角度而言是一种艺术◎（涉及“设计”便是一种艺术◎）

- 子程序之内不可含另一个子程序。
- 宏由子程序构成。事实上宏只不过是“子程序的另一个称谓”而已。
- 当您录制一个宏，就是在您所选择的文档 (.doc) 或模板 (.dot) 的模块 (module) 之内产生一个子程序，像下面这样，其中的 “Sub” 代表 “Subroutine”：

```
Sub MacroName()
    .....macro code.....
End Sub
```

- 子程序名称多长都无所谓，基本原则是愈能够反映子程序的行为愈好，例如 ConvertTaiwanTermsToMainlainTerms 一看就知道是“转换台湾术语为大陆术语”。名称形式有许多种，我喜欢的形式是：各单词之间以首字大写来区分。有人喜欢以下划线区分，也不错。首字母通常大写（此点和变量名称不同）。
- End Sub 指令并不要求写出子程序名称，因为它必须和先前的子程序声明成双成对，否则就是错误。
- 调用子程序的方式是：在某个子程序内直接写出另一个子程序的名称，像这样：

```
Sub ProcedureA()
    ProcedureB      '调用子程序 B
    ProcedureC      '调用子程序 C
End Sub
```

```
Sub ProcedureB()
    ...
End Sub
```

```
Sub ProcedureC()
    ...
End Sub
```

- 子程序可以携带参数，像这样：

```
Sub ProcedureA(param1 As String, param2 As Integer)
    '这里可以拿参数 param1 和 param2 来使用
End Sub
```

- 参数的传递方式有 pass by value (ByVal) 和 pass by reference (ByRef) 两种，前者传入参数的副本（因此即使被修改亦不会影响原件），后者传入参数的本尊（因此一旦被修改会影响原件）。ByRef 是默认行为。

```
Sub ProcedureA(ByRef param1 As String, ByVal param2 As Integer)
    '这里可以拿参数 param1 和 param2 来使用。
    'param1 如果被改动，会影响调用端。
```

```
'param2 如果被改动，不会影响调用端。
End Sub
```

- 子程序没有返回值。

函数 (Function)

函数的作用和子程序 (procedures) 基本相同。以下是不同之处。

- 宏不带参数。所以宏乃是以子程序完成，不是以函数完成。
- 函数可返回数据，也可以不返回。如果要返回，将函数名称视为一个变量并赋予其值，就代表返回：

```
Function Square(num As Long) As Long 6
    Square = num * num;   '这一行将数值返回（给调用端）
End Function

'调用方式
Dim X As Long
X = Square(5)
```

变量 (Variable)

- 程序中用以承载数字或文字的储存空间。
- 既曰变量，就是内容可变（亦即可被写入新值）。
- 变量名称多长都无所谓，其命名原则与子程序的命名相同，但通常首字母小写。
- 下面声明两个变量：一是名为 TextLine 的 String (字符串) 变量，一是名为 CharPos 的 Integer (整数) 变量。

```
Dim textLine As String
Dim charPos As Integer
```

数据型别 (Data Type)

- 任何数据都应该有其型别。例如稍早我们看到的“字符串”和“整数”。

⁶ 此例相当于以下的 C/C++ 函数：

```
long Square(long& num) {
    return num*num;
}
```

➤ 如果您没有为变量指定型别, VBA 会设为默认型别 Variants。我极不鼓励如此, 这会养成很不好的编程习惯, 很容易造成不易除错的程序错误。

➤ VBA 提供的数据型别有:

表10-1 VBA 数据型别 (Data Types)

型别	简介	数值范围 or 补充 or 实例
Numeric 数值	Integer	16-bit 整数 -32,768 ~ +32,767
	Long	32-bit 整数 -2,147,483,648 ~ + 2,147,483,647
	Single	单精度浮点数 1.401298E-45 ~ 3.402823E+38
	Double	双精度浮点数 4.94065645841247E-324 ~ 1.79769313486232E+308
	Currency 货币	货币数额专用 不该使用 Double, 因为一分钱也遗失不得。
	Byte	8-bit 整数 0 ~ 255
	Decimal	12-bytes 整数 适用于复杂科学或统计运算
String (字符串)	长度非固定之字符串	最长可达 20 亿个字符
String * n	长度固定为 n 之字符串	例: Dim str As String * 20
Date 日期	日期专用, 以#字符括起。 VBA 将以你的系统的时间格式 (由你设置)	例: deadLine = #03/25/04# 或 deadLine = #14:20:30 PM#
Boolean 布尔值	真假值, True 或 False。 可使用 Not, Or, And, Xor 四个操作来处理它们。	例: Dim workOk As Boolean Dim sad As Boolean workOk = True sad = Not workOk
Object 对象	用以存放对象 例: Dim objGeneric As Object ;任意型别之对象 Dim objSelection As Selection ;特定型别之对象 Set objSelection = Application.Selection Set objGeneric = objSelection (上面用到的 Selection, Application 都是 Word 对象, 详 10.3 节)	对象可为任意型别 or 特定型别
Variants	可接纳任何型别。如果声明变量时未曾指定型别, 就用此一默认型别。赋值时会自动进行必要的型别转换。可透过 VarType 函数取得内含物的真正型别 (表 10-2)。	例: Dim length Dim width 除非必要, 尽量不用 Variant, 因为如此的弹性必然带来额外开销 (overhead)

型别		简介	数值范围 or 补充 or 实例
Array 阵列	一维	相同型别的一系列数据；每笔数据有个索引。索引默认由 0 开始，但可使用 Option Base n 改变为 n，亦可在声明时直接指定上限和下限。	例：Dim num(10) As Integer num(0) = 5 '第一元素 num(10) = 25 '最后元素 Dim val(1 To 7) As Integer val(1) = 5 '第一元素 val(7) = 25 '最后元素
	多维	同上，但多维。	例：Dim grid(3,5) As Integer grid(0,0) = 5 '第一元素 grid(3,5) = 25 '最后元素
	动态	元素个数不定。将来任何时候可设置或调整大小。	例：Dim addr() As String ReDim Preserve addr(50)
Collections ⁷ 群集	某些数据并非像 array 那样单纯地以位置来组织，而是以元素所带的某个键（key）来组织。这便是 Collections。		例：Dim empl As New Collection empl.Add "jzhou", "1234" MsgBox empl.Item("1234") empl("1234") = "Jason" empl.Remove "1234"
User-Defined 用户自定义	您也可以打造自己的型别。其中所含数据可任意组合。赋值动作 (=) 可处理整笔数据。		例：Type Student ID As long Name As String Phone As String * 7 End Type Dim students(30) As Student students(0).ID = 681001 students(0).Name = "Jason" students(0).Phone= "1234567"

表10-2 VarType 函数的返回值及其意义

常数	数值
vbEmpty	0
vbNull	1

⁷ Word 之中存在许多对象 (objects)，也存在许多群集 (collections)，例如 Documents, Tempaltes 等等。它们并不属于这里所说的 Collection 型别，而是各自属于各自特定的型别（名称就是 Documents, Templates...），但具备这里所说的（群集相关）特征。由于它们的型别各不相同，所以都应该有不同的对待。但它们具备某些相同点，例如都有 Count property。

常数	数值
vbInteger	2
vbLong	3
vbSingle	4
vbDouble	5
vbCurrency	6
vbDate	7
vbString	8
vbObject	9
vbError	10
vbBoolean	11
vbVariant	12
vbDataObject	13
vbDecimal	14
vbByte	17
vbArray	8192

流程控制 (Flow Control)

- 通常，程序要不是重复特定次数的某些动作（这样的重复动作称为循环，loop），要不就是判断某事（物）的真假，决定采取什么样的应对措施。
- VBA 提供的条件判断有：

```
If <Boolean Expression> Then
    ...
    'Actions
End If
```

```
If <Boolean Expression> Then
    ...
    'Actions
Else
    ...
    'Actions
End If
```

```
If <Boolean Expression1> Then
    ...
    'Actions
ElseIf <Boolean Expression2> Then
    ...
    'Actions
End If
```

- VBA 提供的循环动作有 For 和 Do 两种（以下灰色部分可有可无）：

```
For <Counter> = <LowerBound> To <UpperBound> [Step <increment>]
    ...
    'Actions
Next <Counter>
```

例：

```
For i=10 To 1 Step -5
    ...
    'Actions
Next i
```

```
For Each CurrentDocument In Documents
    MsgBox CurrentDocument.Name      ;将文档名称输出至消息框
Next CurrentDocument
```

```
Do While <Boolean Expression>
    ...
    'Actions
Loop
```

```
Do Until <Boolean Expression>
    ...
    'Actions
Loop
```

```
Do
    ...
    'Actions
Loop While <Boolean Expression>
```

```
Do
    ...
    'Actions
Loop Until <Boolean Expression>
```

举个例子，下面的意思是，当 NOT EOF(1) 成立（为真）时，做某些动作：

→ Do While Not EOF(1) '当 Not EOF(1) 成立（亦即当 EOF(1) 不成立）
 ... '做某些动作
 Loop '回到 Do 指令，继续“判断…执行…”的过程

- VBA 强制跳离循环的动作有 Exit For 和 Exit Do 两种。插入于上述任何一种 For 或 Do 循环，即可跳离循环。

开启文件 (Open File)

- 如果需要输入的数据过多，或者数据内容并非固定，那就不该写死在程序中，应该写在文件内由程序读取。这也增加了程序的弹性。
- 10.4.3 节有个“两岸术语转换”宏实例，由于待转术语可能与时俱增，并且可能

因为两岸用字习惯的改变而有所更动，所以由用户维护一个转换文字表（文件）并由程序读取之，是最好的解法。

- 以下指令开启一个用以输入的文件（路径必须写明），令为编号 1：

```
Open "c:\windows\desktop\convert.txt" For Input As #1
```

读取文件内容 (Read File)

- 以下指令读取先前被开启、用于输入、编号为 1 的文件。读取方式是一次一行（读到换行符为止），并将结果放进 TextLine。读到的东西都视为文字（即使读到 3 或 2.5 这类文字），因此 TextLine 必须是个 String 变量。

```
Line Input #1, TextLine
```

关闭文件 (Close File)

- 文件必须有开有闭，否则可能无法保有最新（最后的）数据。这是因为电脑系统都有“缓冲”（buffering）概念，会把数据存放在内存中，适当时时候才真正写入文件。“缓冲区满载”是数据被系统真正写入文件的适当时机，“关闭文件”则是另一个时机。

- 以下指令将先前被开启为编号 1 的文件关闭：

```
Close #1
```

With、MsgBox()、CStr()、Chr()

这里介绍四个很被频繁使用的 VBA 语句和函数。范例请见 10.3.2 节末尾。

- With 语句：VBA 带有面向对象（Object Oriented）味道⁸，其程序由众多对象组合运转而成。为方便在程序中使用对象而不必写出太冗长的对象名称，遂有 With 语句。
- MsgBox 函数：可显示一个消息框，是最简单的信息输出工具（至屏幕）。
- CStr 函数：可将任意型别的数据转换为字符串，搭配 MsgBox 输出。
- Chr 函数：可生成并返回一个 ASCII 码字符，搭配 MsgBox 输出。

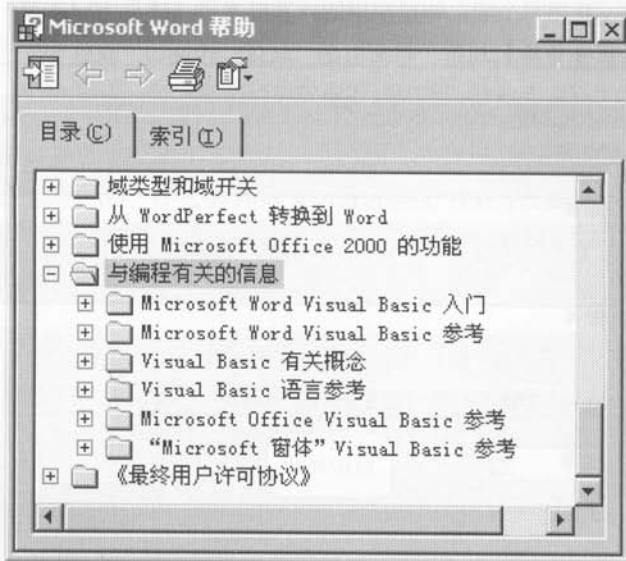
⁸ 虽然 VBA 并不完全是个面向对象编程语言⑨

10.2.2. 自学 VBA Programming 的好办法

VBA 语法对于稍具程序编写经验的人并不困难，但对于完全不懂编程的人就可能颇有难度。宏少不得要使用 Word 对象，而 Word 对象花色繁多，没有静下心来花点功夫加以组织，难有所得。先把所有 VBA 语法和所有 Word 对象都看过一遍再卷起袖子开始干，当然很好，可很多人时间不允许，很多人的勇气和毅力也需要大锻炼才行。

学习的最大动力是需求，我们不妨从需求着手。不论您希望宏为您做些什么，如果先自己操作一遍（或一部分），并打开宏录制器将过程录制起来，然后观察其中的 VBA 程序代码，就可以看到您最需要的 VBA 语法和 Word 组件。

有了这些线索，再加上一份参考手册，那就万事不求人了。点击【帮助 ➔ Microsoft Word 帮助(H)】，打开 Word 的《*On-line Help*》，点击其中的 Microsoft Word Visual Basic 参考：



便可获得一份堪称详细的 Visual Basic 语言参考手册：



举个例子，我想写个宏，转换两岸的计算机术语（详见 10.4.3 节），因此我先录制一个【编辑→替换】动作，将所有的“软体”转换为“软件”：

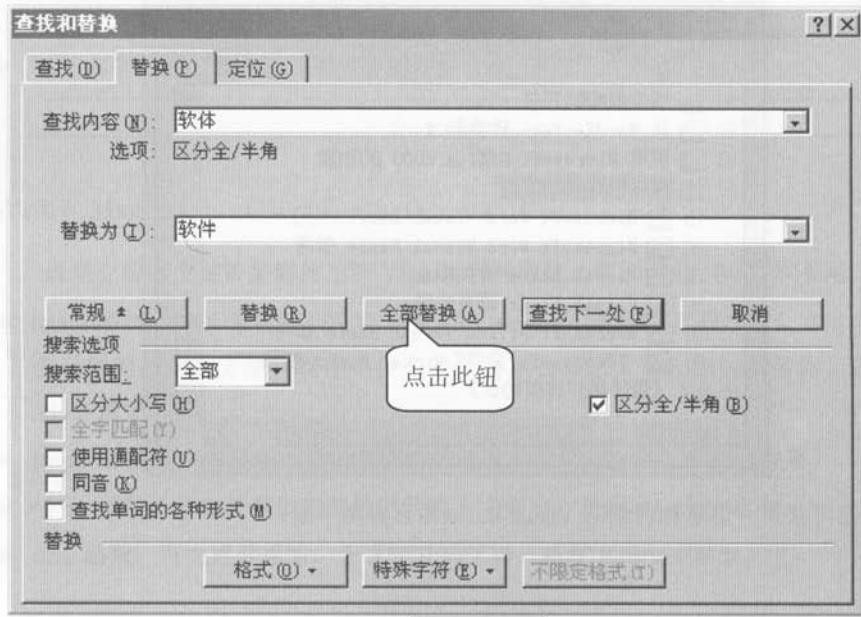


图10-13 查找和替换对话框

获得这样的宏内容：

```

#001 Sub 繁简术语转换()
#002 '
#003 ' 繁简术语转换 Macro
#004 ' 宏在 2004-7-2 由 jjhou 录制
#005 '
#006     Selection.Find.ClearFormatting
#007     Selection.Find.Replacement.ClearFormatting
#008     With Selection.Find
#009         .Text = "软体"
#010         .Replacement.Text = "软件"
#011         .Forward = True
#012         .Wrap = wdFindContinue
#013         .Format = False
#014         .MatchCase = False
#015         .MatchWholeWord = False
#016         .MatchWildcards = False
#017         .MatchSoundsLike = False
#018         .MatchAllWordForms = False
#019         .MatchByte = True
#020     End With
#021     Selection.Find.Execute Replace:=wdReplaceAll
#022 End Sub

```

VBA10-1 宏，将文档中的所有“软体”替换为“软件”

如此一来我就有了线索，可到前述的 Microsoft Word Visual Basic 联机帮助文档中检索什么是 Selection，什么是 Find, Replacement, Text, Forward, Format ……

10.3. Word 对象 (Objects)

VBA 语法困难度不高，是比较容易解决的技术问题。相形之下林林总总的 Word 对象才是“以程序操控 Word”的最大关键。作为一个 Word programmer，您不但必须具备面向对象概念，手上还必须有一本详实的 Word 对象手册⁹。

10.3.1. 面向对象 (Object Oriented)

面向对象是近十数年来编程语言 (programming language) 的主流，也是编程技术的主流。其基本概念是：整个程序全部(或尽可能大部分)由对象组成，类 (classes) 负责定义对象 (objects)，其中定义了属性 (properties) 和动作 (methods)。

⁹ 我推荐《Word 2000 VBA Programmer's Reference》。详见本书附录 A。

在 VBA 中，对象（object）代表的是一个可被描述以及被处置的某种“东西”。对象提供对外接口（interfaces）。对象具有 **property**（属性），用来描述其自我性质（attribute）。对象也有 **method**（方法/函数），用来执行某些动作。对象的完整定义（包括 properties 和 methods）置于所谓的类（class）内。我们可以根据一个类产生无数对象，每个对象都被视为该类的一个实体（instance）。

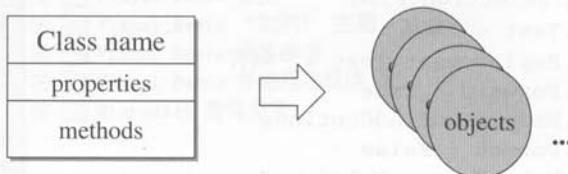


图10-14 类（classes）与对象（objects）的关系



本书以“类”称呼 class，以“对象”称呼 object。但本书将保留 property 和 method 不译。property 若译为“属性”会和 attribute 冲突（我以“属性”表示 attribute），method 若译为“方法”会和一般用语混淆，缺乏术语突出性。其实我也希望保留 class 和 object 不译，但太多英文术语有时候会造成版面上的（视觉）破坏。



对 C/C++/Java/C# 程序员而言，property 就像是 get/set 访问函数（accessor）；虽然是个函数，却可以以变量的形式来使用它。



请注意，以下以 *Lucida Console* 字体表现的，都是类名称：名称末尾若有 's'，表示是个群集（collection）。ABC class 实现出来的对象就称为 ABC 对象。以 *Arial* 字体表现的都是 properties。以 *Times New Roman* 斜体字表现的都是 methods。

关于对象、类、properties、methods，让我举个和本书最直接相关的例子。如果您要编写一个排版软件（请把 Word 的使用经验搬出来吧），可以这样想象：

（※ 以下所说其实便是 Word 的实际设计）

- 需要一个 Application 对象用来代表这个排版软件。排版的主体是文档，但还是有一些属性“独立于文档之外”，例如 ActivePrinter（当前打印机），ActiveDocument（当前文档），ActiveWindow（当前视窗），FontNames（所有可用字体）、Height（主视窗高度）、Width（主视窗宽度）……当然还少不得需

要一个用以维护排版主体（文档）的结构：Documents。Application 对象应该具备以下动作能力：*Help*（启动帮助系统）、*NewWindow*（为当前文档开一个新视窗）、*PrintOut*（打印）、*Quit*（结束）……

- **Documents** 是个群集 (collections，详见表 10-1)，用来串起一份份文档 (Word 不是可以同时开启好几份文档吗②)。它应该具备 *Add*、*Open*、*Close*、*Save* 等动作，使得以添加、开启、关闭、储存元素 (每个元素都是文档，**Document**)。
- **Document** 是排版 (或文档编写) 的主体。想一想，文档内有些什么呢 (进行至此您从本书学习了些什么页面元素呢)？应该有整体内容 (**Content**)、段落 (**Paragraphs**)、文句 (**Sentences**)、单词 (**Words**)、字符 (**Characters**)、脚注 (**Footnotes**)、尾注 (**Endnotes**)、批注 (**Comments**)、域 (**Fields**)、超链接 (**Hyperlinks**)、索引 (**Indexes**)、节 (**Sections**)、绘图形状 (**Shapes**)、表格 (**Tables**)、章节目录 (**TablesOfContents**)、图目录 (**TablesOfFigures**)，以及各种自定引文目录 (**TablesOfAuthorities**)。这些统统都应该是 **Document** 对象的 properties；以上除了 Content 属于 Range 对象，其他都应该是各自属于他们自己的同名类。
- 除了上述与文档内容有关的组成，**Document** 对象还应该包含相应的视窗 (**Windows/Windows**)、文件名 (**Name/String**)、文件路径 (**Path/String**)、密码 (**Password/String**)、是否只读 (**ReadOnly/Boolean**)、是否被修改 (**Saved/Boolean**)、所有样式 (**Styles/Styles**) 等 properties。
- **Document** 对象也应该包括以下动作 (methods)：语法检查 (*CheckGrammer*)、拼写检查 (*CheckSpelling*)、关闭 (*Close*)、计算统计数据 (*ComputeStatistics*)、追踪超链接 (*FollowHyperlink*)、打印 (*PrintOut*)、重复键入 (*Redo*)、复原键入 (*Undo*)、储存文件 (*Save*)、另存为 (*SaveAs*)、圈选 (*Select*)……
- **Paragraph** (段落) 对象应该有左缩进 (*LeftIndent*)、右缩进 (*RightIndent*)、行距 (*LineSpacing*)、大纲级别 (*OutlineLevel*)、样式 (*Style*) 等性质 (properties)。也应该有缩进 (*Indent*)、单倍行距 (*Space1*)、1.5 倍行距 (*Space15*)、2 倍行距 (*Space2*) 等动作 (methods)。
- **Sentences** (文句) 对象应该拥有一系列 Range 对象，用来表现文档 (**Document**) 或某范围 (**Range**) 或某圈选区 (**Selection**) 内的所有文句。倒是不必存在 **Sentence** 对象。
- **Words** (单词) 对象应该拥有一系列 Range 对象，用来表现文档 (**Document**) 或某范围 (**Range**) 或某圈选区 (**Selection**) 内的所有单词。倒是不必存在

Word 对象。

- **characters** (字符) 对象应该拥有一系列 Range 对象, 用来表现组成其父对象 (parent object) 的每一个字符。每个 Range 对象表现一个字符。倒是不必存在 Character 对象。
- **Sections** (节) 对象应该是个群集, 内含一个个 Section 对象。每个 Section 对象应该有边界 (Borders)、页眉 (Headers)、页脚 (Footers)、页面设置 (PageSetup) 等性质 (properties)。
- **styles** (样式) 对象应该是个群集, 内含一个个 Style 对象 (包括 Word 内建和用户自定)。Style 对象应该有以下性质 (properties): 基准样式 (BaseStyle/Variant)、是否内建 (BuiltIn/Boolean)、字体 (Font)、描述文字 (Description/String)、是否被使用 (Inuse/Boolean)、列表级别 (ListLevelNumber/Long)、后续段落样式 (NextParagraphStyle/Variant)、样式类型 (Type) ……
- **Selection** (圈选区) 对象所代表的是文档中某个 (可视的) 被圈选区, 或 Word 之中的目前插入点 (文字输入点)。此对象类似 Range 对象, 但一份文档可关联许多 Range 对象, 而 Word 却只该有一个 active Selection 对象。这个对象拥有“用以代表各种页面元素及各种设置”的性质 (properties), 包括 Paragraphs、Sentences、Words、Characters、Text (圈选区内的文字)、HeaderFooter、Hyperlink、Footnotes、Endnotes、Fields、Bookmarks、Comments、Cells (表格单元格)、PageSetup、Font、Style……
- Selection (圈选区) 对象中还应该有一个特别的 property: Find。它返回一个 Find 对象, 有能力在圈选区中寻找特定文字。
- **Find** (寻找) 对象用来表现 Word **查找和替换**对话框 (中的**查找**属性页), 两者有完全相同的功能。此对象可被 Range 对象和 Selection 对象处理。在 Selection 对象中使用 Find 对象时, 一旦找到符合条件的文字, Selection 对象的圈选范围会有所改变。Find 对象有些什么 properties 和 methods 呢? 想想【编辑→查找】+**查找和替换**+**查找**提供了什么, 对应过来就是了②, 其中应该有 Text、Font 用以描述待寻文字, 另一个 property 是 Replacement。
- **Replacement** 对象代表 Word **查找和替换**对话框中的**替换**属性页, 两者有完全相同的功能。父对象永远是 Find 对象。Replacement 对象应该有 Text、Font 两个 properties 用以描述替换文字。

有了以上描述，程序设计经验丰富的人可能已经在心中勾勒出一幅 Word 程序架构图。（当然啦，冥想比较简单，动嘴巴最快，实务则是非常非常复杂！）即使缺乏程序设计经验，您也可以从以上描述对应本书先前所谈的各种页面元素和版式设置，而有一个更清楚的梗概和更深刻的印象。

10.3.2. Word 对象模型 (Object Model)

VBA 并不是 Word 专属语言，而是一个一般性编程语言，适用于任何内建 VBA 的应用程序。VBA 对应用程序的动作主要是透过程序显露出来的许多对象。

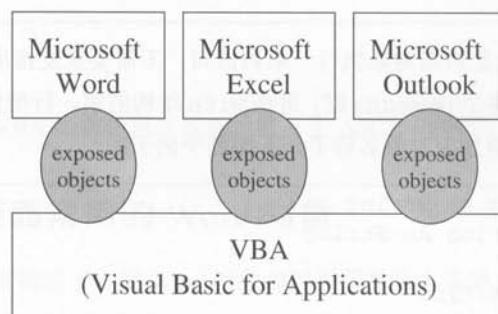


图10-15 VBA 和应用程序 (applications) 的关系

这些被应用程序显露出来的对象，以及从它们身上又展开出去的其他任何对象，组成所谓的应用程序对象模型 (Application's Object Model)。有了应用程序对象模型，VBA 才能够和该应用程序一起运转活动。

Word 拥有一个十分庞大而复杂的对象模型 (**Word Object Model**)。其中众多对象所代表的东西，从整个 Word 程序，到表格内的每一单元格、文档内的每一段落 (paragraphs)、段落内的每一文句 (sentences)、文句内的每一单词 (words)、单词内的每一字符 (characters) ……，巨细靡遗无所不含。这些对象，依据其描述的对象在文档内的上下层次，构成了一个庞大的阶层体系 (hierarchy)。稍后我将挑选一些重要的、易理解的 Word 对象加以介绍并给出实例。

由于 Word 对象层层叠叠，所以我先谈谈如何方便地在程序中使用它们，以及如何在 Word 环境下显示所获信息。

With 语句 (statement)

VBA 语法规规定以句点 ‘.’ 来访问对象的 properties，例如：

```
Dim capsLock As Boolean
Dim caption As String
...
caption = Application.Caption
capsLock = Application.CapsLock
```

Word 对象的 properties 往往又是 Word 对象，而且很可能一连数层。一连串写下去容易出错，也带给程序员比较大的负担。

With 语句可让您对某对象执行一系列语句，不需要重复指出对象名称。例如要改变一个对象的多个 properties 时，可在 with 结构的第一行指出对象名称，其后的赋值动作就不必再加上对象名称了。下面是个例子：

```
Dim findString As String
Dim replaceString As String
...
With Selection.Find
    .Text = findString
    .Replacement.Text = replaceString
    .Replacement.Font.Color = wdColorBlue
    .Forward = True
    .Wrap = wdFindContinue
End With
```

相当于（不使用 With 语句）：

```
Dim findString As String
Dim replaceString As String
...
Selection.Find.Text = findString
Selection.Find.Replacement.Text = replaceString
Selection.Find.Replacement.Font.Color = wdColorBlue
Selection.Find.Forward = True
Selection.Find.Wrap = wdFindContinue
```

MsgBox 函数

MsgBox 是 VBA 提供的一个函数，用来显示信息（只能是字符串）于屏幕上。特别适合用于快速而单纯地“显示变量值”或“表示到达宏的某处”。用法如下：

```
MsgBox Application.ActivePrinter   'String
MsgBox Selection.Text           'String
```

```

MsgBox Selection.Font.Name      'String
MsgBox Selection.Font.Size     'Single
MsgBox "test Help"            'String

```

CStr 函数

CStr 是一个 VBA 函数，可将任意型别的数据转换为字符串：

```

Dim msg As String
msg = "目前所有 Characters: "
+ CStr (ActiveDocument.Characters.Count)
+ Chr (10)
MsgBox msg

```

Chr 函数

Chr 是一个 VBA 函数，可生成并返回一个 ASCII 码字符。实例如上。

10.3.3. 重要而常用的 Word 对象

这一节我特别挑选了一些 Word 对象加以说明并给出实例。我的挑选原则是：

- 本章稍后数个宏范例中用到
- 对理解 Word 程序结构有帮助

这些 Word 对象在 10.3.1 节已有相当程度的描述(当时假设排版软件应该如何如何，其实便是以 Word 为蓝图)，阅读时请交叉参考该处。本节以阶层体系图再次描述它们的部分 properties 和 methods，并给出访问实例。

以下所列各个 Word 对象，如果对象名称即其类名称，那么就只显示对象名称，像这样： Style，否则对象名称显示于前，类名称显示于后，像这样： Path/String。

如果对象(or 类)名称是复数形式(后有 's')，表示是个 collection(请参考表 10-1)，那么我会将它和其元素对象(or 类)的名称叠放在一起显示，像这样：

Documents
Document

以下各个 Word 对象的测试程序，适合以 10.1.5 节的测试环境来进行。这些测试程序几乎每一行都以 MsgBox 输出信息，有时会让用户不胜其扰。更好的做法是把所有文字信息集中(或分段)以 MsgBox 展示。但字符串的处理并非此处重心，程序代码太多徒然混乱焦点。因此我把文字信息的合并工作留给读者。

图 10-16 显示 Word 对象模型之中与“Word 文档组织”最有关的一些对象，并以它们在文档中的彼此从属关系来描绘它们。

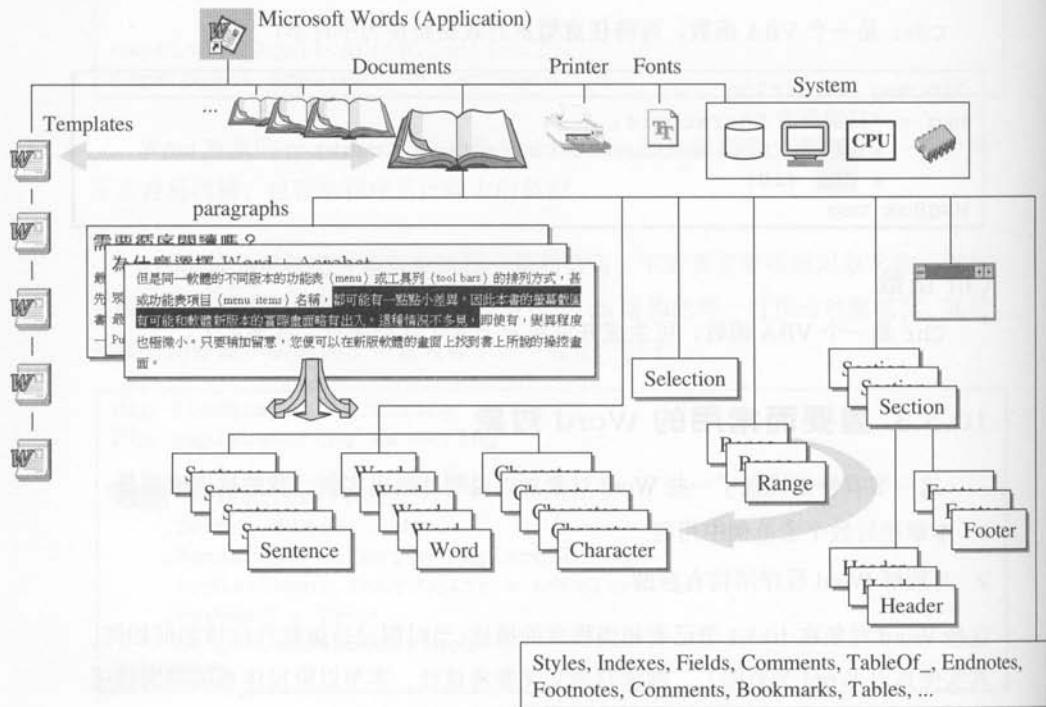


图10-16 Word 对象的逻辑组织

面对图 10-16，您可以将图中对象的上下关系视为“上位者拥有下位者”。因此您可以说 **Application** 对象拥有 **Templates**、**Documents**、**ActivePrinter**、**Fonts** 和 **System** 对象，而 **Documents** 对象拥有 **Paragraphs**、**Sentences**、**Words**、**Characters**、**Sections** 等对象。特别请注意的是，并非：

- **Document** 对象拥有 **Paragraph** 对象，后者拥有 **Sentences** 对象，后者拥有 **words** 对象、后者拥有 **Characters** 对象。

而是：

- **Document** 对象拥有 **Paragraph**、**Sentences**、**Words**、**Characters** 对象。

Application

Application 对象位居 Word 对象阶层体系最上层，表现的是 Word 程序本身。其 properties 和 methods 独立于任何文档之外。这个对象一共拥有 91 个 properties 和 62 个 methods，图 10-17 列出局部。通常这个对象被用来访问 Word 对象模型中的其他所有对象，或透过它取得 Word 默认信息（如 **UserName** 或 **UserInitials** 等等），或是透过它了解 Word 环境设置，例如打印机、视窗大小等等。

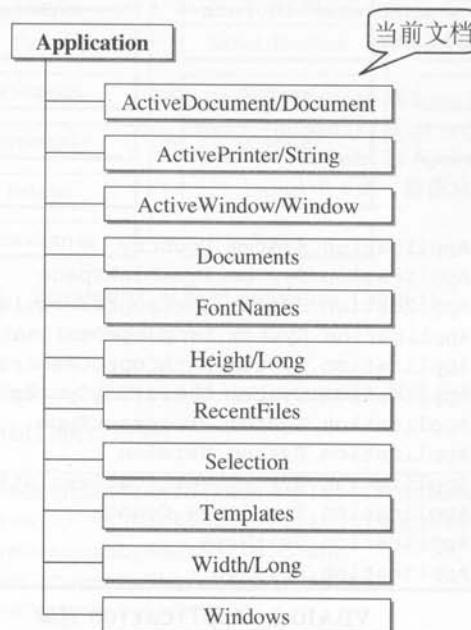


图10-17 Application 对象的 properties (局部列出)

下面以消息框 (MsgBox) 显示从 Application 对象获得的信息。程序中的注释便是我的电脑当时的执行结果：

#001 MsgBox Application.ActivePrinter	'HP LaserJet 4p - LPT1:
#002 MsgBox Application.Height	'582
#003 MsgBox Application.Width	'774
#004 MsgBox Application.Build	'8.0.4614
#005 MsgBox Application.Caption	'Microsoft Word - lsword
#006 MsgBox Application.DefaultSaveFormat	'空字符串表示Word documents
#007 MsgBox Application.DisplayRecentFiles	'True
#008 MsgBox Application.Documents.Count	'3 (目前开启的文档数)

```

#009 MsgBox Application.FontNames.Count      '99 (目前安装可用的字体数)
#010 MsgBox Application.Left                '-3
#011 MsgBox Application.MacroContainer.FullName
#012             'C:\Program Files\Microsoft Office\Templates\Normal.dot
#013 MsgBox Application.NormalTemplate.FullName
#014             'C:\Program Files\Microsoft Office\Templates\Normal.dot
#015 MsgBox Application.RecentFiles.Count    '9
#016
#017 Dim mnRecentFile As RecentFile
#018 For Each mnRecentFile In RecentFiles
#019     Dim str As String
#020     str = mnRecentFile.Path + "\" + mnRecentFile.Name
#021     MsgBox str
#022 Next
#023     'E:\lsword\doc\b5\lsword.doc
#024     'E:\personal\common\robert.doc
#025     'E:\temp\t1.doc
#026     '(依此类推, 共 9 个)
#027
#028 MsgBox Application.System.Country        '886
#029 MsgBox Application.System.FreeDiskSpace   '2147155968
#030 MsgBox Application.System.HorizontalResolution '1024
#031 MsgBox Application.System.LanguageDesignation 'Chinese (Taiwan)
#032 MsgBox Application.System.MathCoprocessorInstalled 'True
#033 MsgBox Application.System.OperatingSystem    '中文 Windows
#034 MsgBox Application.System.ProcessorType       'Pentium
#035 MsgBox Application.System.Version           '4.10
#036 MsgBox Application.System.VerticalResolution '768
#037 MsgBox Application.Templates.Count         '2
#038 MsgBox Application.UserName               'jjhou
#039 MsgBox Application.Version                '8.0a

```

VBA10-2 Application 对象

Documents/Document

Documents 是一个群集 (collection)，内含目前被开启的所有文档。每一份文档相应于一个 **Document** 对象。图 10-18 列出 **Document** 对象的一部分性质 (properties)；实际有 122 个 properties 和 62 个 methods。

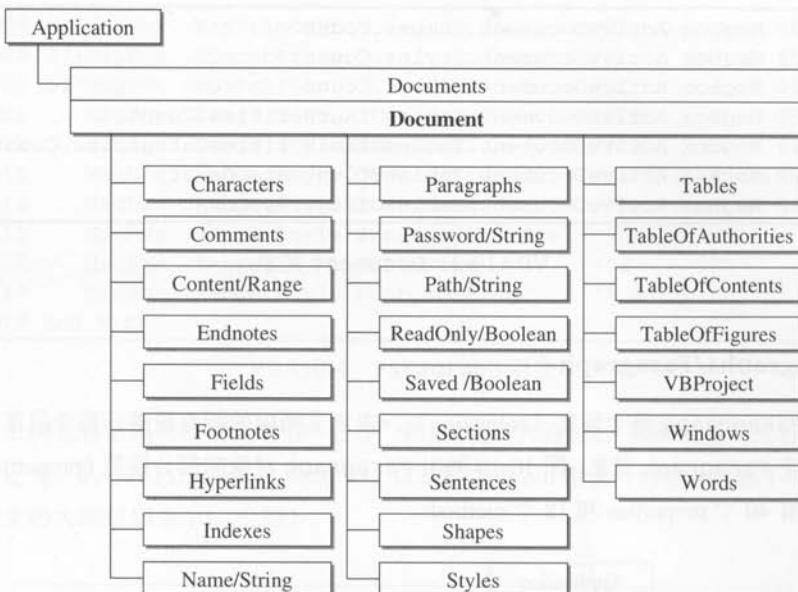


图10-18 Document 对象的 properties (局部列出)

下面以消息框 (MsgBox) 显示从 ActiveDocument 对象获得的信息。程序中的注释便是我的电脑当时的执行结果：

```

#001 MsgBox ActiveDocument.AttachedTemplate.FullName
#002      'C:\Program Files\Microsoft Office\Templates\lsword.dot
#003 MsgBox ActiveDocument.Bookmarks.Count          '2
#004 MsgBox ActiveDocument.Characters.Count        '131352
#005 MsgBox ActiveDocument.CodeName                'ThisDocument
#006 MsgBox ActiveDocument.Comments.Count          '1
#007 MsgBox ActiveDocument.Endnotes.Count          '1
#008 MsgBox ActiveDocument.Fields.Count           '1082
#009 MsgBox ActiveDocument.Footnotes.Count         '48
#010 MsgBox ActiveDocument.FullName               'E:\lsword\doc\b5\lsword.doc
#011 MsgBox ActiveDocument.HasPassword            'False
#012 MsgBox ActiveDocument.Hyperlinks.Count       '3
#013 MsgBox ActiveDocument.Indexes.Count          '0
#014 MsgBox ActiveDocument.ListParagraphs.Count   '243
#015 MsgBox ActiveDocument.ListTemplates.Count    '71
#016 MsgBox ActiveDocument.Paragraphs.Count       '4571
#017 MsgBox ActiveDocument.Password              'type error
#018 MsgBox ActiveDocument.ReadOnly              'False
#019 MsgBox ActiveDocument.Saved                 'False
#020 MsgBox ActiveDocument.Sections.Count         '20
#021 MsgBox ActiveDocument.Sentences.Count        '5399
  
```

#022 MsgBox ActiveDocument.Shapes.Count	'420
#023 MsgBox ActiveDocument.Styles.Count	'167
#024 MsgBox ActiveDocument.Tables.Count	'24
#025 MsgBox ActiveDocument.TablesOfAuthorities.Count	'0
#026 MsgBox ActiveDocument.TablesOfAuthoritiesCategories.Count	'16
#027 MsgBox ActiveDocument.TablesOfContents.Count	'1
#028 MsgBox ActiveDocument.TablesOfFigures.Count	'3

VBA10-3 Document 对象

Paragraphs/Paragraph

Paragraphs 是个群集 (collection)，内含文档内的所有段落。每个段落相应于一个 **Paragraph** 对象。图 10-19 列出 **Paragraph** 对象的部分性质 (properties)。实际有 40 个 properties 和 18 个 methods。

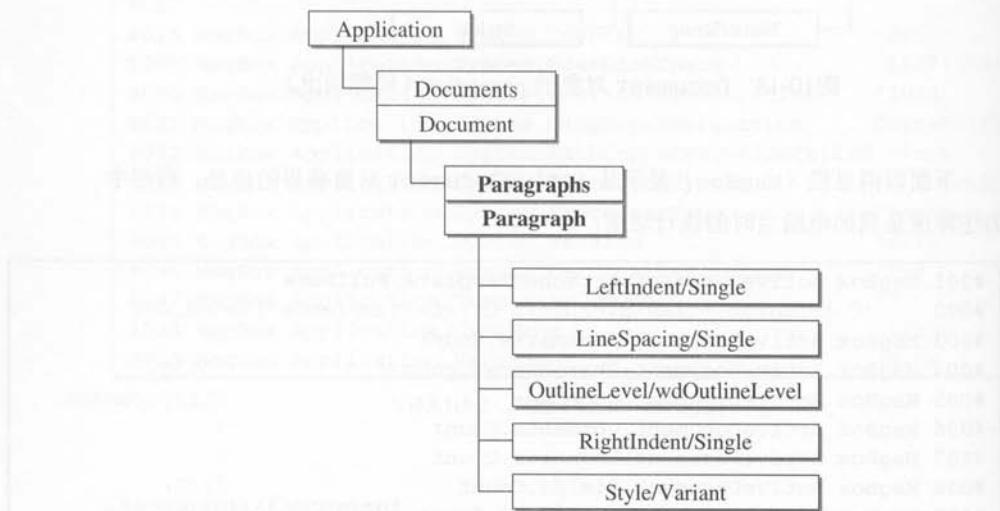


图 10-19 Paragraph 对象的 properties (局部列出)

下面以消息框 (MsgBox) 显示从 **Selection** 对象获得的信息。程序中的注释便是我的电脑当时的执行结果：

#001 '圈选 (或将鼠标移至) 本书的“自序”标题	
#002 MsgBox Selection.Paragraphs.Count	'1
#003 Set prg = Selection.Paragraphs.First	
#004 With prg	
#005 MsgBox .LeftIndent	'0
#006 MsgBox .LineSpacing	'16
#007 MsgBox .OutlineLevel	'1

```

#008    MsgBox .RightIndent           '0
#009    MsgBox .SpaceBefore          '5
#010    MsgBox .SpaceAfter           '30
#011    MsgBox .Range.Text           '自序
#012    MsgBox .Range.Style.NameLocal '标题 1
#013    MsgBox .Range.Style.Description '!章名 alt-m+缩进:...
#014    MsgBox .Range.Style.Font.Name   'Footlight MT Light
#015    MsgBox .Range.Style.Font.NameFarEast '方正粗圆简体
#016    MsgBox .Range.Style.Font.Size     '24
#017    MsgBox .Range.Style.Font.Spacing  '0.4
#018 End With

```

VBA10-4 Paragraph 对象

上例是将鼠标光标移至本书“自序”标题所获得的结果。如果将鼠标光标移到一般正文，其 OutlineLevel 将是 10，这是因为 Word 默认（内建）标题 1~9，所以内文的大纲级别为 10，合理！

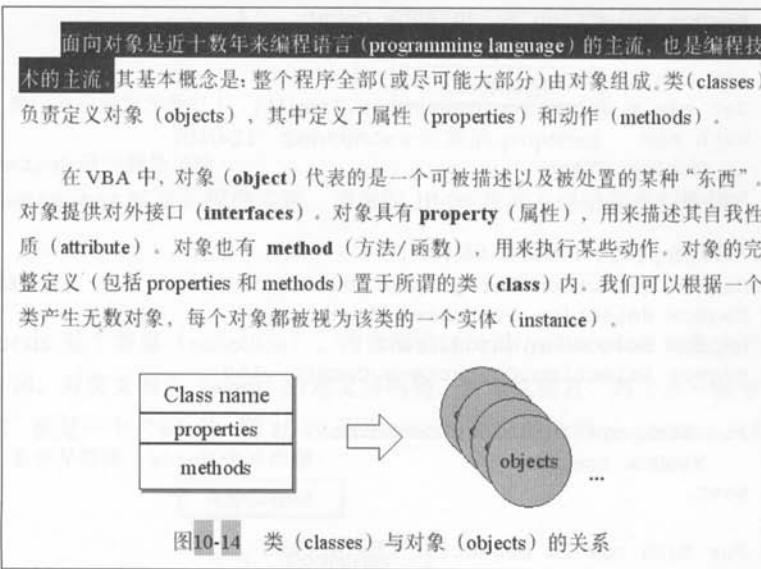


图10-14 类 (classes) 与对象 (objects) 的关系

图10-20 解释 paragraphs, sentences, words, characters

再举一个例子。图 10-20 和 VBA10-5 可让我们窥知 Word 如何对待段落内的文字。该图内总共有 4 个 paragraphs 和 11 个 sentences：

- 4 个 paragraphs 相当明显：二段文字、一张图片（被视为一个 paragraph）、一个题注（也被视为一个 paragraph）。

- 11 个 sentences 分别是：二段文字中以句号 ('。' 或 '。') 为分割符所拆解出来的 3+6 个 sentences，加上图片（被视为内含一个 sentence）和题注（也被视为内含一个 sentence）。

图中圈选起来的（高亮反白）区域有 1 个 sentence、9 个 words、50 个 characters：

- 1 个 sentence，给无疑问。

- 9 个 words 分别是：

面向对象是近十数年来编程语言	(programming	lanugage) 的主流，	也是编程技术的主流。
----------------	--------------	----------	--------	------------

- 50 个 characters 分别是：每个中文字算一个，每个英文字符（字母）算一个，每个空格也算一个。所以是：'面'、'向'、'对'、'象'、'是'、'近'……'('、'p'、'r'、'o'……

```

#001 '圈选图 10-20 整个范围
#002 MsgBox Selection.Paragraphs.Count      '4
#003 MsgBox Selection.Sentences.Count        '11
#004 Dim rge As Range
#005 Set rge = Selection.Sentences.Item(1)    '只要尝试指定索引值 1~11
#006 With rge
#007   MsgBox .Text                          '便可看到所有 Sentences
#008 End With
#009
#010 '圈选图 10-20 中的高亮 (反白) 区
#011 MsgBox Selection.Paragraphs.Count      '1
#012 MsgBox Selection.Sentences.Count        '1
#013 MsgBox Selection.Words.Count           '9
#014 MsgBox Selection.Characters.Count      '50
#015
#016 For Each rge In Selection.Words
#017   MsgBox rge.Text                      '输出 9 个 Words, 如稍早所言。
#018 Next
#019
#020 For Each rge In Selection.Characters
#021   MsgBox rge.Text                      '输出 50 个 Characters, 如稍早所言
#022 Next

```

VBA10-5 剖析 paragraphs、sentences、words 和 characters

Sentences

Sentences 是个群集 (collection)，内含圈选区（以 **Selection** 对象表示）内的所有以句点 (‘。’ 或 ‘.’) 分隔开来的文字。图 10-21 列出 **Sentences** 对象的全部 properties，只有 2 个！

必须注意的是，“文句” (**sentence**) 是由 **Range** 对象来表现，而不是由 **Sentence** 对象来表现。Word 对象模型中并不存在所谓的 **Sentence** 对象。

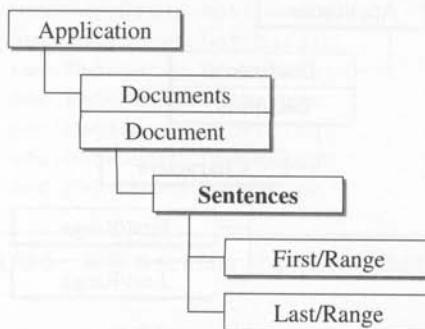


图10-21 Sentences 对象的 properties

关于 **Sentences** 对象的程序实例，请看图 10-20 及其下说明和范例程序。

Words

words 是个群集 (collection)，内含圈选区（以 **Selection** 对象表示）内的所有单词。对英文而言 “word” 的意义很明确。对中文而言“两个非一般中文字所夹击的”就是一个 “word”。图 10-22 列出 **words** 对象的全部 properties，只有 2 个！

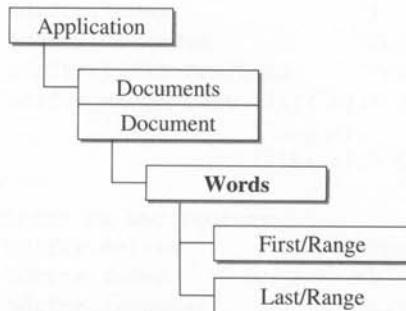


图10-22 words 对象的 properties

关于 `Words` 对象的程序实例，请看图 10-20 及其下说明和范例程序。

Characters

`Characters` 是个群集 (collection)，内含圈选区 (以 `Selection` 表示) 内的所有字符。对英文而言 "character" 就是 "字母或标点符号"，对中文而言则是 "一个中文字或符号"。图 10-23 列出 `Characters` 对象的全部 properties，只有 2 个！

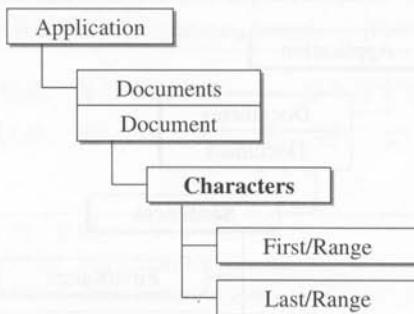


图10-23 `Characters` 对象的 properties

关于 `Characters` 对象的程序实例，请看图 10-20 及其下说明和范例程序。

Sections/Section

Sections 是个群集 (collection) , 内含整份文档中的每一个节 (section) , 每个节相应于一个 **Section** 对象。图 10-24 列出 **Section** 对象的部分 properties. 实际有 7 个 properties 和 0 个 methods. 下面是范例程序代码。

```
#001 Dim sec As Section
#002 Set sec = ActiveDocument.Sections.First
#003 With sec
#004     MsgBox sec.PageSetup.BottomMargin      '39.7
#005     MsgBox sec.PageSetup.LeftMargin        '92.15
#006     MsgBox sec.PageSetup.RightMargin       '92.15
#007     MsgBox sec.PageSetup.TopMargin         '65.2
#008     MsgBox sec.PageSetup.PaperSize        '41
#009     MsgBox sec.PageSetup.PageHeight       '652.05
#010     MsgBox sec.PageSetup.PageSize          '524.5
#011 End With
```

第一个 Section

VBA10-6 观察 Section 对象和 PageSetup 对象

```
#001 Dim bdr As border
#002 Dim ftr As HeaderFooter
#003 Dim hdr As HeaderFooter
#004 Set sec = ActiveDocument.Sections.Item(12)
#005                                         'Sections.Item(12) 亦可
#006 MsgBox sec.Borders.Count           '4, 不论哪个 Section 都一样。
#007 MsgBox sec.Footers.Count          '3, 不论哪个 Section 都一样。
#008 MsgBox sec.Headers.Count          '3, 不论哪个 Section 都一样。
#009
#010 For Each hdrftr In sec.Headers
#011     MsgBox hdrftr.Exists           'True  False  True
#012     MsgBox hdrftr.Index            '1      2      3
#013     MsgBox hdrftr.IsHeader        'True  True   True
#014     MsgBox hdrftr.LinkToPrevious  'True  True   True
#015     MsgBox hdrftr.Range.Text     '217(\n)10.3Word 对象 (Objects)
#016                                         '(blank)
#017                                         '218(\n)10 宏、VBA、Word 对象
#018 Next
#019 For Each hdrftr In sec.Footers
#020     MsgBox hdrftr.Exists           'True  False  True
#021     MsgBox hdrftr.Index            '1      2      3
#022     MsgBox hdrftr.IsHeader        'False False  False
#023     MsgBox hdrftr.LinkToPrevious  'True  True   True
#024     MsgBox hdrftr.Range.Text     'The Art of Word - Writing and Typesetting Large Scale Documents
```

随便找个 Section 瞧瞧

```
#026 '(blank)
#027 'The Art of Word - Writing and Typesetting Large Scale Documents
#028 Next
```

VBA10-7 剖析 Section 对象（执行结果写于注释内）

下面是对 VBA10-7 执行结果的分析整理：

In Section.Headers

Exists	Index	IsHeader	LinkTo Previous	Range.Text	说明
True	1	True	True	217 10.3Word 对象 (Objects)	右页页眉的外侧页码 右页页眉的内侧大标题名
False	2	True	True	(blank)	(blank)
True	3	True	True	218 10 宏、VBA、Word 对象	左页页眉的外侧页码 左页页眉的内侧章名

In Section.Footers

Exists	Index	IsHeader	LinkTo Previous	Range.Text	说明
True	1	False	True	The Art of Word - Write and Typeset Large Scale Documents	右页页脚的右侧文字
False	2	False	True	(blank)	(blank)
True	3	False	True	The Art of Word - Write and Typeset Large Scale Documents	左页页脚的右侧文字

对于这些 Headers 和 Footers 栏位，目前我还有一些地方不解，尚未找到答案：

- 每个 Section 对象有三个 Header 对象和三个 Footer 对象，为什么？
- 三个 Header 对象和三个 Footer 对象中的第二个做什么用？其 Exists property 是 False，表示它其实不存在。为什么 Word 需要设计这个东西？或者说，为什么 Word 要让 Header 对象和 Footer 对象的编号跳号进行？
- 不论选择观察哪个 Section 对象（上例观察的是 Sections.Item(12)），显示的结果（上表中的 Range.Text）却总是输入点所在页（前后）的页眉和页脚内容。

- 随着选择之 **Section** 对象不同（例如选择观察 **Sections.Item(9)**），上述两个表格呈现的内容有时候是奇数页（右页）先，有时候是偶数页（左页）先。

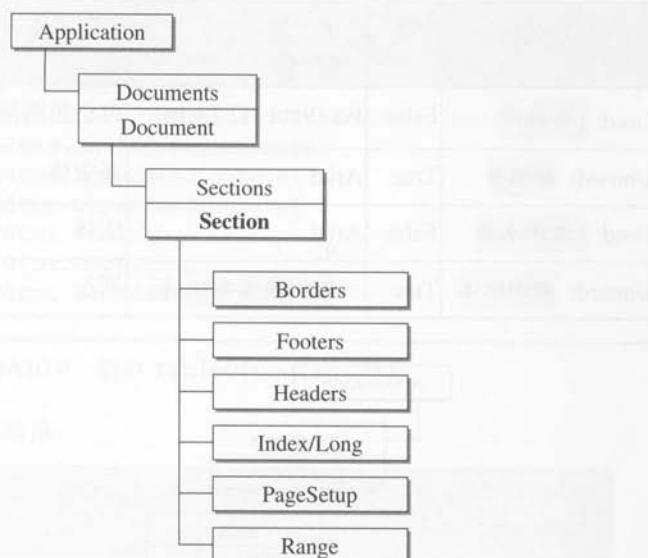


图10-24 **Section** 对象的 properties (局部列出)

Styles/Style

Styles 是个群集 (collection)，内含整份文档所有的样式 (Styles)，每个样式相应于一个 **Style** 对象。图 10-25 列出 **Style** 对象的部分 properties。实际有 18 个 properties 和 2 个 methods。下面是范例程序代码。

```

#001 '请注意：以下会产生许多 MsgBox，让您按到手软。建议自行处理（合并）字符串
#002 Dim stl As style
#003 For Each stl In ActiveDocument.Styles
#004   If stl.InUse Then
#005     MsgBox "Used: " + stl.NameLocal
#006   Else
#007     MsgBox "Unused: " + stl.NameLocal
#008   End If
#009   MsgBox stl.Builtin
#010   MsgBox "Font: " + stl.Font.Name          '英文字体
#011   MsgBox "Font: " + stl.Font.NameFarEast   '中文字体
#012   MsgBox stl.Type
#013 Next
  
```

VBA10-8 列出文档的所有样式

此例将目前文档“使用中”和“未使用”的样式（styles）全部列出。下面是部分执行结果：

样式名称 & 使用状态	内建	英文字体	中文字体	类型	说明
Used: !中标 ¹⁰	False	Footlight MT Light	方正粗圆简体	1	类型 1 代表段落样式
Unused: 标题 9	True	Arial	新宋体	1	
Used: !文内术语	False	Arial	宋体	2	类型 2 代表字符样式
Unused: 强调粗体	True	(默认段落字体+粗)	同左	2	

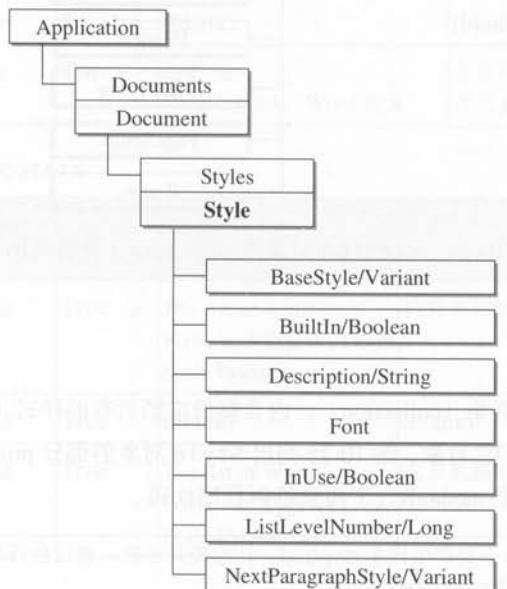


图10-25 Style 对象的 properties（局部列出）

TablesOfContents/ TableOfContents

VBA10-3 中曾经出现 TablesOfContents 对象。这是个群集（collection），内含整份文档所有的内容目录（TableOfContents），每一份内容目录相当于一个

¹⁰ 我总是为“自定样式”之名称加上前置符号 ‘!’，并在名称最后加上快捷键助忆符号（例如 alt-z）。此处之显示暂略后者。

TableofContens 对象。图 10-26 列出 TableofContens 对象的全部 properties 共 12 个，另有 3 个 methods。下面是范例程序代码。

```
#001 MsgBox ActiveDocument.TablesOfContents.Count '1
#002 Dim tblCnt As TableOfContents
#003 For Each tblCnt In ActiveDocument.TablesOfContents
#004     MsgBox tblCnt.IncludePageNumbers
#005     MsgBox tblCnt.LowerHeadingLevel
#006     MsgBox tblCnt.UpperHeadingLevel
#007     MsgBox tblCnt.TableID
#008     MsgBox tblCnt.UseFields
#009     MsgBox tblCnt.UseHeadingStyles
#010 Next
```

VBA10-9 剖析 TableofContens 对象

下面是 VBA10-9 的执行结果：

含页码?	HeadingLevel 下限	HeadingLevel 上限	TableID	使用域?	使用 HeadingStyle?
True	5	1	(blank)	False	True

Application

Documents

Document

TablesOfContents

TableOfContents

HeadingStyles

TabLeader / wdTabLeader

HidePageNumbersInWeb / Boolean

TableID / String

IncludePageNumbers / Boolean

UpperHeadingLevel / Long

LowerHeadingLevel / Long

UseFields / Boolean

Range

UseHeadingStyle / Boolean

RightAlignPageNumbers / Boolean

UseHyperlinks / Boolean

图10-26 TableofContens 对象的 properties

TablesOfFigures/ TableOfFigures

VBA10-3 中曾经出现 **TablesOfFigures** 对象。这是个群集 (collection)，内含整份文档所有的图表目录 (TableOfFigures)，每一份图表目录相应于一个 **TableOfFigures** 对象。图 10-27 列出 **TableOfFigures** 对象的所有 properties 共 14 个，另有 3 个 methods。下面是范例程序代码。

```
#001 MsgBox ActiveDocument.TablesOfFigures.Count '3
#002 Dim tblFig As TableOfFigures
#003 For Each tblFig In ActiveDocument.TablesOfFigures
#004     MsgBox tblFig.Caption
#005     MsgBox tblFig.IncludeLabel
#006     MsgBox tblFig.IncludePageNumbers
#007     MsgBox tblFig.LowerHeadingLevel
#008     MsgBox tblFig.UpperHeadingLevel
#009     MsgBox tblFig.TableID
#010     MsgBox tblFig.UseFields
#011     MsgBox tblFig.UseHeadingStyles
#012     MsgBox tblFig.Range.Text
#013 Next
```

VBA10-10 剖析 **TableOfFigures** 对象

下面是 VBA10-10 的执行结果：

题注 Caption	含标签 Label?	含页码 ?	Heading Level 下限	Heading Level 上限	TableID	使用域 (Fields)?	使用 Heading Style?	Range.Text
表	True	True	9	1	(blank)	False	False	(整个表目录)
图	True	True	9	1	(blank)	False	False	(整个图目录)
VBA	True	True	9	1	(blank)	False	False	(整个 VBA 程序目录)

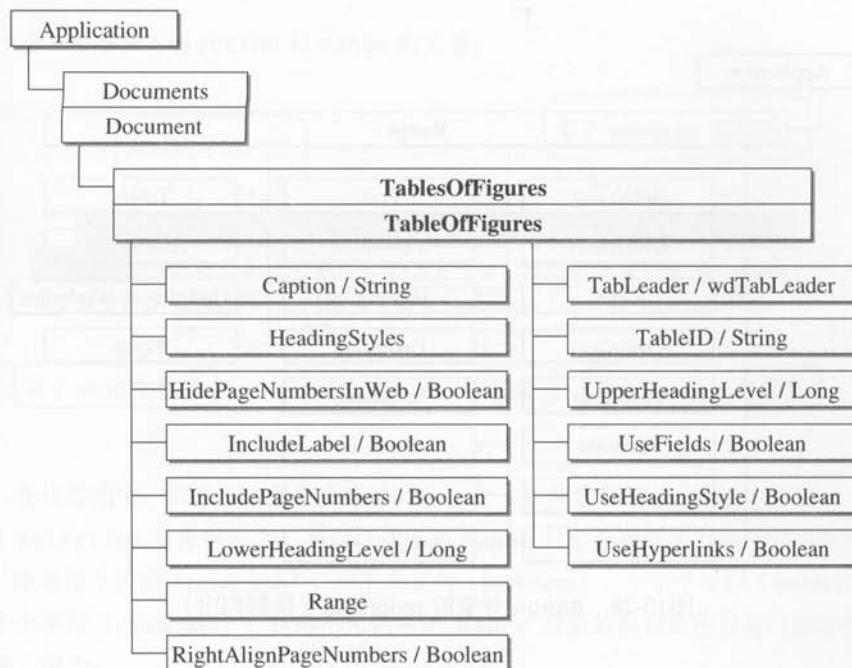


图10-27 TableOfFigures 对象的 properties

Range

Range 对象用来定义文档内特定的一个点或一个区域，是最常被使用的一种 Word 对象。由于此物用来描述文档内的区域，因此不难想象一些“描述文档某部分”的对象(例如 **Footnote**, **Endnote**, **Bookmark**, **Cell**, **Comment**, **Row**, **Section**, **Index**, **Hyperlink**, **Selection**, **Table**……) 都有可能是 **Range** 的“父对象”。

图 10-28 列出 **Range** 的部分 properties，实际有 73 个 properties 和 66 个 methods。

先前的 VBA10-4、VBA10-5、VBA10-7 已展示 **Range** 对象的使用实例。

Selection

Selection 对象用以表现文档中的“被圈选区”。如果没有任何区域被圈选，那么就代表目前插入点。**Selection** 的内容相当类似 **Range**，但一份文档可拥有多个 **Range** 对象，整个 Word 却只会有一个 active **Selection** 对象。图 10-29 列出 **Selection** 的部分 properties，实际有 54 个 properties 和 93 个 methods。

先前的 VBA10-4、VBA10-5 已展示 **Selection** 对象的使用实例。

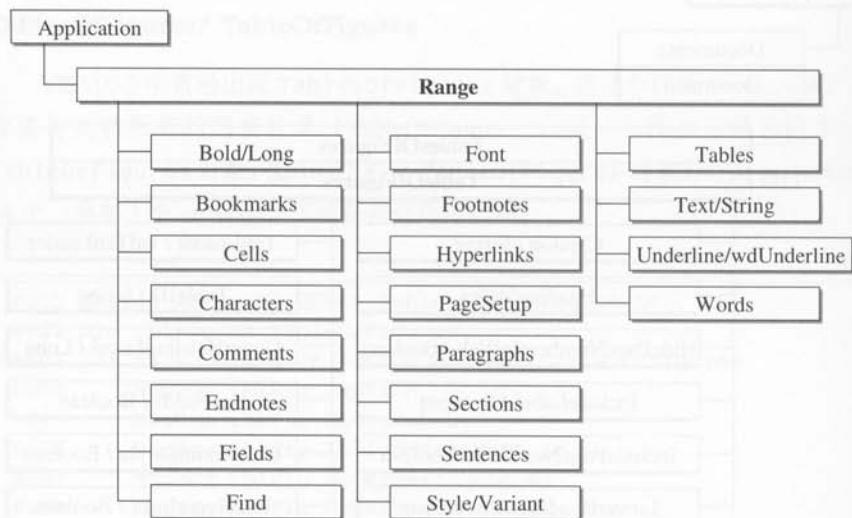


图10-28 Range 对象的 properties (局部列出)

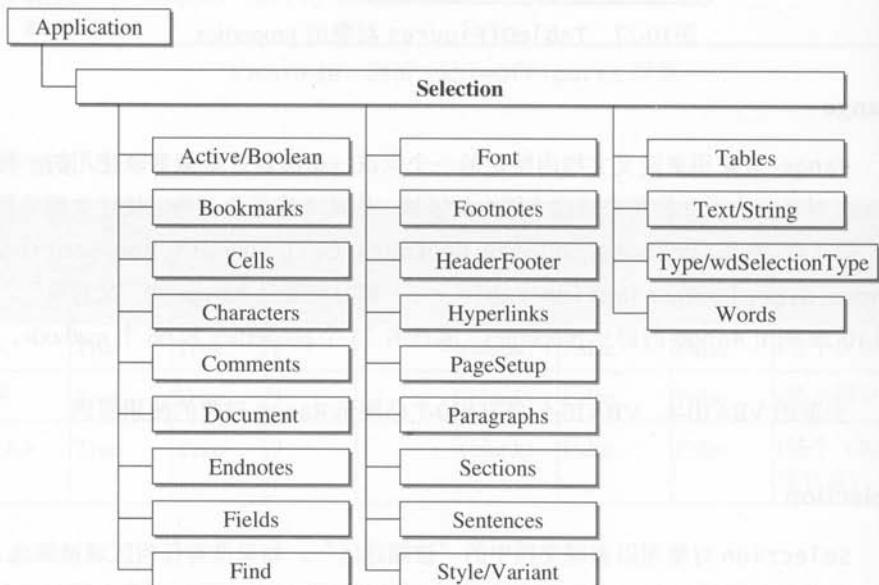
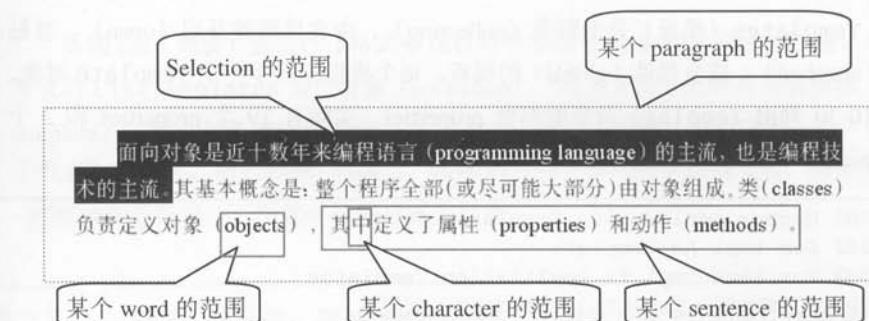


图10-29 Selection 对象的 properties (局部列出)

下图进一步解释 Selection 和 Range 的关系:



在这张图中, 高亮区 (黑底白字区) 由一个 (亦是整个 Word 系统中的惟一一个) Selection 对象描述之。另有许许多多 Range 对象分别描述上图内的不同事物, 像是整个段落 (paragraph)、一个个文句 (sentences)、一个个单词 (words)、一个个字符 (characters)。这些形形色色的 Range 对象的涵盖范围很有可能互相重叠, 因为:

- 段落内的文句 (sentence), 其范围当然被涵盖在段落内
- 文句内的单词 (word), 其范围当然被涵盖在文句内
- 单词内的字符 (character), 其范围当然被涵盖在单词内

Templates/Template

Templates (模板) 是个群集 (collection)，内含目前被开启 (open)、被贴附 (attached)、或全局性 (global) 的模板。每个模板相应于一个 **Template** 对象。图 10-30 列出 **Template** 对象的部分 properties，实际有 19 个 properties 和 2 个 methods。下面是范例程序代码：

```
#001 MsgBox Application.Templates.Count      '2
#002 Dim tmpl As Template
#003 For Each tmpl In Application.Templates
#004     MsgBox tmpl.FullName
#005     MsgBox tmpl.Name
#006     MsgBox tmpl.Path
#007     MsgBox tmpl.Saved
#008     MsgBox tmpl.Type
#009 Next
```

VBA10-11 剖析 **Template** 对象

下表是执行结果：

FullName	Name	Path ¹¹	Saved	Type
OfficePath\Templates\lsword.dot	lsword.dot	OfficePath\Templates	True	2
OfficePath\Templates\normal.dot	normal.dot	OfficePath\Templates	True	0

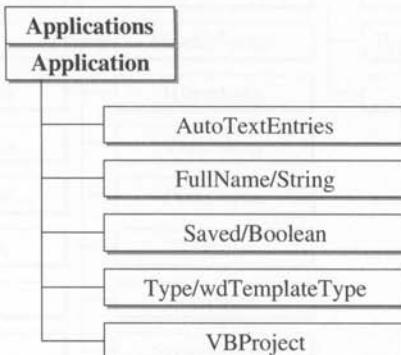


图10-30 **Template** 对象的 properties (局部列出)

¹¹ 为求精简表述，这里以 OfficePath 代表 "C:\Program Files\Microsoft Office"。

ListTemplates/ListTemplate

所谓 list (列表) 是指以【格式→项目符号和编号(N)】制作而成的段落。这里所说的 ListTemplates 是个群集 (collection)，内含文档中的所有列表模板 (list template)。每个列表模板相应于一个 ListTemplate 对象，也相应于一个“适用于列表”的样式，例如 **列表~列表 5、列表项目符号~列表项目符号 5、列表编号~列表编号 5** 等等。下图就是这些样式的展示：

这是一个
Wingdings
字体符号

“列表”的样式，例如 **列表~列表 5、列表项目符号~列表项目符号 5、
列表编号~列表编号 5** 等等。下图就是这些样式的展示：

- 本句使用 **列表项目符号** 样式
- 本句使用 **列表项目符号 2** 样式
- 本句使用 **列表项目符号 3** 样式
- 本句使用 **列表项目符号 4** 样式
- 本句使用 **列表项目符号 5** 样式

1. 本句使用 **列表编号** 样式
 1. 本句使用 **列表编号 2** 样式
 1. 本句使用 **列表编号 3** 样式
 1. 本句使用 **列表编号 4** 样式
 1. 本句使用 **列表编号 5** 样式

本句使用 **列表** 样式

本句使用 **列表 2** 样式

本句使用 **列表 3** 样式

本句使用 **列表 4** 样式

本句使用 **列表 5** 样式

这些样式主要是定义出不同的编号或符号格式，以及不同的缩进距离

↑ 使用各种“适用于列表”的样式后，所得的屏幕截图

图 10-31 列出 ListTemplate 的全部 4 个 properties。它没有 methods。下面是范例程序代码：

```
#001 MsgBox ActiveDocument.ListTemplates.Count '86
#002 Dim listTmpl As ListTemplate
#003 Dim i As Integer
#004
#005 For i = 1 To 10 Step 1
#006   Set listTmpl = ActiveDocument.ListTemplates(i)
#007   MsgBox listTmpl.ListLevels.Count
#008   MsgBox listTmpl.ListLevels(1).Font.Name
#009   MsgBox listTmpl.ListLevels(1).LinkedStyle
#010   MsgBox listTmpl.ListLevels(1).NumberFormat
#011   MsgBox listTmpl.ListLevels(1).NumberPosition
#012   MsgBox listTmpl.Name
#013   MsgBox listTmpl.OutlineNumbered
#014 Next i
```

VBA10-12 剖析 ListTemplate 对象

下表是执行结果，显示某 10 个列表模板的格式设置。请就其样式名称、编号（符号）格式、编号（符号）位置，和上一页的清单图比对，获得更多的认识。

Count	Font. Name	LinkStyle	Number Format	Number Position	Name	Outline Numbered
1	(blank)	列表编号 5	%1.	96	(blank)	False
1	(blank)	列表编号 4	%1.	72	(blank)	False
1	(blank)	列表编号 3	%1.	48	(blank)	False
1	(blank)	列表编号 2	%1.	24	(blank)	False
1	Wingdings	列表项目符号 5	(blank)	96	(blank)	False
1	Wingdings	列表项目符号 4	(blank)	72	(blank)	False
1	Wingdings	列表项目符号 3	(blank)	48	(blank)	False
1	Wingdings	列表项目符号 2	(blank)	24	(blank)	False
1	(blank)	列表编号	%1.	0	(blank)	False
1	Wingdings	列表项目符号	(blank)	0	(blank)	False

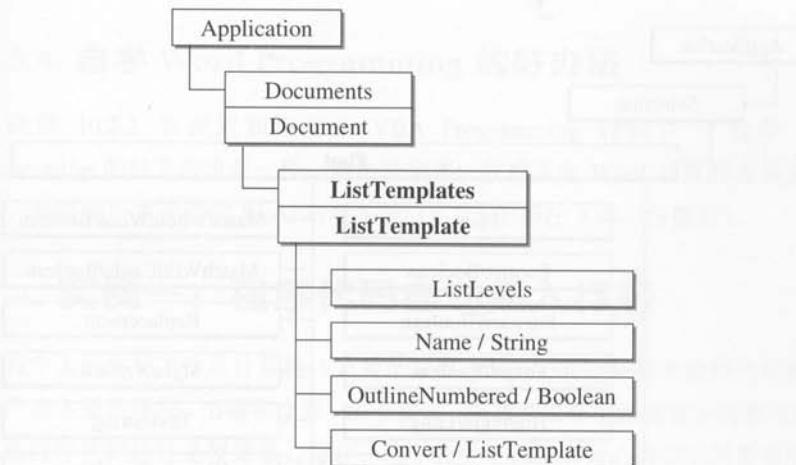


图 10-31 ListTemplate 对象的 properties

Find

Find 对象代表着 Word 的【查找和替换】对话框（中的【查找】属性页）；两者有完全相同的功能。Find 对象可被 Range 对象和 Selection 对象处理。在 Selection 对象中使用 Find 对象时，一旦找到符合条件的文字，Selection 对象的选取范围将会有所改变。图 10-32 列出 Find 对象的部分 properties。此物实际有 27 个 properties 和 4 个 methods。稍后我将连同 Replacement 对象一起完成范例程序。

Replacement

Replacement 对象代表 Word 的【查找和替换】对话框中的【替换】属性页；两者有完全相同的功能。其“父对象”永远是 Find 对象。图 10-33 列出 Replacement 对象的部分 properties。此物实际有 9 个 properties 和 1 个 methods。

下面这个例子找出文档之中所有的“发轫”，替换为“起源”：

```

#001 With Selection.Find
#002     .Text = "发轫"
#003     .Replacement.Text = "起源"
#004     .Forward = True
#005     .Wrap = wdFindContinue
#006 End With
#007 Selection.Find.Execute Replace:=wdReplaceAll
  
```

VBA10-13 示范 Find 对象和 Replacement 对象

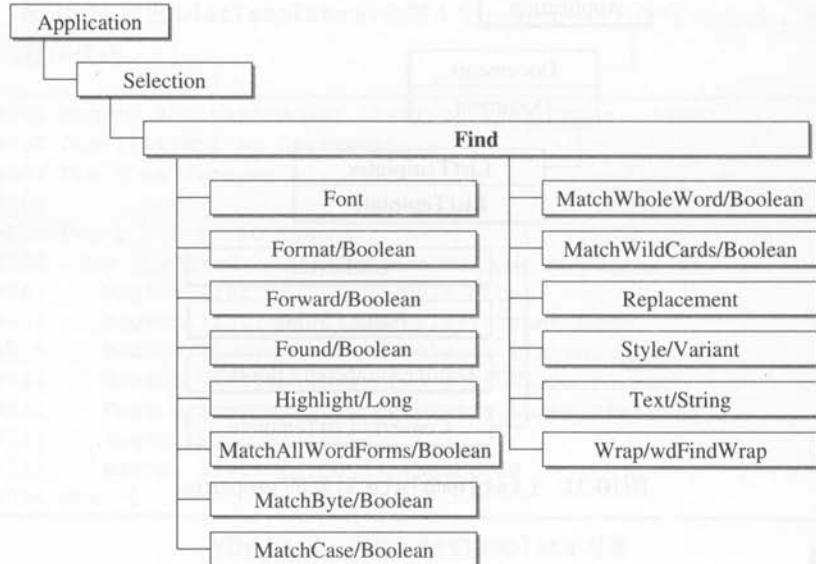


图10-32 Find 对象的 properties (局部列出)

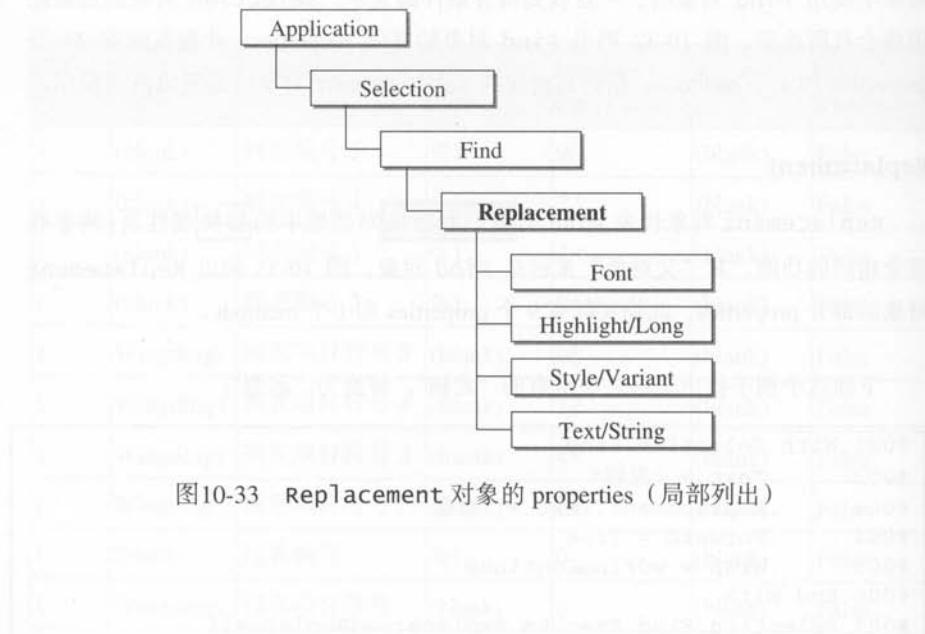


图10-33 Replacement 对象的 properties (局部列出)

10.3.4. 自学 Word Programming 的好办法

就像 10.2.2 节谈到如何自学 VBA Programming 时所言，自学 Word Programming 的好方法也是一样：藉由录制宏，获得某些 Word 对象的多多少少的概念，再搭配一本完善的 Word 对象手册（p.275 的脚注 9 有一份推荐）。

10.4. 实例一：程序代码自动插入行号

我个人的主要工作是计算机（尤其是编程，programming）技术的研究和教育，我的产出主要是课程、书籍和文章。由于讲义、书籍和文章中经常会出现程序代码，而这些程序代码往往是授课重心所在，为了让学生和读者做心得笔记时易有所本，也为了上课或与读者讨论时可以轻松指出“第 x 页上半段程序代码第 x 行”这样的明确目标，所以不论各段程序代码数量多寡，我总是希望它们都能够带上行号（每一段都从 1 起编）。

10.4.1. 全自动加行号

如果为书中程序代码设计一个专用的**!程序代码**样式，是不是我就可以写个宏，搜寻整份文档，找出所有该样式，然后将每一段代码加上从 1 起编的行号呢？¹²

这个想法很好。首先要知道一点，每一行程序代码都以换行符结尾，因此每一行程序代码都是“以**!程序代码**为样式”的段落。欲在整份文档中找到一大块一大块“以**!程序代码**为样式”的文字内容，可以这么做：

- 令变量 StartParag 从 1 开始，走过文档内的所有段落。
- ■ 如果 StartParag 遭遇**!程序代码**样式，就令 EndParag 从下一个段落开始继续往下，直到遭遇非**!程序代码**样式的段落为止。
- 现在，StartParag 和 EndParag 便圈起了一整块**!程序代码**。此范围内的每一个段落将是一行程序代码。现在，为每一行程序代码加上行号（详后）。
- 令 StartParag 从目前的 EndParag 开始继续往下走。

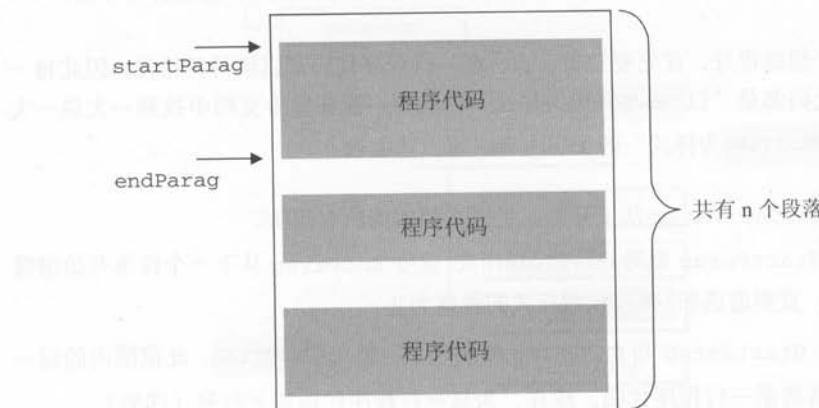
这个流程可以图 10-34 表示。程序代码如 VBA10-14。

¹² Word 内建有“自动编列行号”功能，见 3.4.4 节。但那并不是我要的！

```

#001 Dim startParag    '区块中的第一个段落（代表第一行程序代码）
#002 Dim endParag      '区块中的最后一个段落（代表最后一行程序代码）
#003   '注：采前闭后开区间 [ ]，所以 endParag 其实是代表最后一行的下一段落
#004 Dim nLineNum       '行号（数值）
#005 Dim sLineNum As String '行号（文字）
#006 Dim i
#007
#008 For startParag = 1 To ActiveDocument.Paragraphs.Count
#009     If ActiveDocument.Paragraphs(startParag).style = " !程序代码" Then
#010         For endParag = (startParag + 1) To ActiveDocument.Paragraphs.Count
#011             If ActiveDocument.Paragraphs(endParag).style <> " !程序代码" Then
#012                 Exit For
#013             End If
#014         Next endParag
#015
#016     '面对找出来的这个区块
#017     For i = startParag To endParag - 1      '每一段落（每行程序代码）
#018         nLineNum = (i - startParag + 1)        '计算行号（从 1 开始）
#019         ...
#020     Next i
#021     startParag = endParag      '准备寻找下一个 " !程序代码" 区块
#022 End If
#023 Next startParag

```

VBA10-14 找出文档内的每一区块!**程序代码**图10-34 找出文档内的每一整块!**程序代码**

这个办法好则好矣，但有些地方颇为棘手：

- 大型文档的段落数量可能高达数千个（本书就有 6000 个以上的段落），需要相

当长的处理时间。为此最好另设计一些“进度表”之类的画面给心焦的用户看，但那需要另费一番功夫，而且偏离主题。

- 操作过一遍这样的动作之后，如果又有新的！**程序代码**加进来，不能再执行一次这个宏，否则会发生二次编号。虽然程序中多做一些处理便可解决这个问题，但显然我们的负担又将加重。
- 程序代码区段如果只有 2~3 行，实在也没必要为它们加行号！
- 有时候我会在文档中加一些！**程序代码**空行，用以调整页面布局，因此若以正常的**内文**作为空行，前后空格也忒大了些。这固然不是好习惯——我其实应该设计一个专门样式作为调整页面布局时的空行之用——但毕竟已经发生了。那么，上述宏做法会把我的那些其实并非程序代码的“！**程序代码**空行”加上行号。

10.4.2. 半自动加行号

其实我并不需要如此全自动的“加行号”功能。只要能够让我圈选一块范围，再由 Word 在那个范围加行号，就可以了。下面是新构想的宏：

```
#001 Dim parag As Paragraph
#002 Dim nLineNum
#003 Dim sLineNum As String
#004 Dim i
#005
#006 nLineNum = 0
#007 For Each parag In Selection.Paragraphs      '对每行代码(每个段落)
#008     nLineNum = nLineNum + 1                  '行号
#009     parag.Range.Select                      '圈选目前这一行
#010
#011     sLineNum = str(nLineNum)                 '行号转为文字
#012     sLineNum = LTrim(sLineNum)                '去除字符串开头和结尾的空格
#013     For i = 1 To (3 - Len(sLineNum))        '补足三位数(前加'0')
#014         sLineNum = "0" + sLineNum
#015     Next i
#016     sLineNum = "#" & sLineNum                '前加符号 '#'
#017     sLineNum = sLineNum + " "
#018     Selection.InsertBefore sLineNum          '将行号安插于文字之前
#019 Next
```

VBA10-15 加行号，错误示范

但是这个宏只为“被圈选的第一个段落（第一行程序代码）”加行号，就停摆了。怎么回事？

问题出在其中使用了 `Range.Select`，这个动作会改变 `Selection` 对象的圈选范围，而由于循环是倚赖圈选范围来进行的：

```
For Each parag In Selection.Paragraphs    '对于每一行代码（每个段落）
```

所以再没有第二行需要处理，也就停摆了。

如果把循环改成这样：

```
#001 Dim rge As Range          '存放从 Selection 取出的 Range
#002
#003 For nLineNum = 1 To Selection.Paragraphs.Count '对每个段落
#004     ... '备妥行号（文字）sLineNum
#005     Set rge = Selection.Paragraphs(nLineNum).Range '取出个段落
#006     rge.Text = sLineNum + rge.Text                  '加上行号
#007 Next nLineNum
```

还是只有第一个段落会加上行号，原因一样：`Selection` 对象在 `rge.Text` 获得新值时被改变了。

如果一开始先把 `Selection` 对象记录下来，就不怕稍后被改变。下面是新写法：

```
#001 Dim rge As Range          '存放一个个 Range
#002 Dim selRge As Range       '存放最初的 Selection Range
#003
#004 Set selRge = Selection.Range '首先记录 Selection
#005 For nLineNum = 1 To selRge.Paragraphs.Count '对每个段落
#006     ... '备妥行号（文字）sLineNum
#007     Set rge = selRge.Paragraphs(nLineNum).Range '取出段落范围
#008     rge.Text = sLineNum + rge.Text                '为文字加上行号
#009 Next nLineNum
```

老问题解决了。现在所有被圈选的段落都会被加上行号。但问题又来了：由于我是以“重新设置文字”的方式来加行号，因此其段落格式（paragraph format）会采用下一段落的样式。也就是说如果我临时起意为程序代码添加一些格式花样（字符颜色、边框等等。不在样式设置内，而是视状况加上），它们在宏执行过后还存不存在，就取决于下一个段落。影响最大的当然是最后一行程序代码了，因为其下一段落的样式不再是！**程序代码**而是**正文**。

这绝非我想要的结果。最后我找到另一种办法，以 `Range` 对象的 `InsertBefore` 取代稍早的“重新设置文字”，终于彻底解决了问题：

```

#001 Sub 程序代码加行号()
#002 Dim nLineNum          '行号(数值)
#003 Dim sLineNum As String '行号(文字)
#004 Dim selRge As Range   '用以记录整块程序代码(多行)
#005 Dim i                 '计数器
#006
#007 Set selRge = Selection.Range      '首先记录 Selection
#008 For nLineNum = 1 To selRge.Paragraphs.Count '对于每个段落
#009     sLineNum = str(nLineNum)           '行号转为文字
#010     sLineNum = LTrim(sLineNum)         '去除开头和结尾的空格
#011     For i = 1 To (3 - Len(sLineNum)) '补足三位数(前加'0')
#012         sLineNum = "0" + sLineNum
#013     Next i
#014     sLineNum = "#" & sLineNum        '前加符号 '#'
#015     sLineNum = sLineNum + " "        '后加空格
#016
#017     selRge.Paragraphs(nLineNum).Range.InsertBefore (sLineNum)
#018     '将行号插入于前
#019 Next nLineNum
#020 End Sub

```

VBA10-16 加行号，正确示范

10.4.3. 改变注释颜色

如果能够将程序代码中的注释颜色替换为某种颜色，将会相当有助于阅读。由于身为作者的我事先知道本书得采黑蓝双色印刷，因此将书中程序代码的注释替换为蓝色，是一个很实际的用途。

首先我必须在每一行程序代码中找出注释起始位置，再将其后的所有文字（全被视为注释）更换颜色。寻找注释起始位置非常简单，因为 VBA 程序很固定地以单引号 '' 作为注释起始记号。只要找到它，以它为基准，将该行文字切为两部分，前半部就是程序代码，后半部就是注释。10.5.3 节介绍了 VBA 提供的字符串处理函数，这里将用到其中的 InStr() 和 Mid()，见表 10-3。

由于这一节是在原设计之后新添的构想，所以以不更动先前程序为原则。因此加工前的每一行程序都已经有了行号。首先记录每一行程序代码（在此为一个段落）的 Range（注：以下程序片段的行号等同于稍后完整宏 VBA10-17 的行号）：

```

#008 Dim lineProgramRange As Range      '单行程序代码范围
...
#029 Set lineProgramRange = selRge.Paragraphs(nLineNum).Range

```

然后找出注释起始记号 '' 的位置：

```
#030     TextLine = lineProgramRange.Text      '取得整行文字(代码)
#031     CharPos = InStr(1, TextLine, Chr(39)) '寻找注释起始点
```

如果没有找到 ''，表示此行程序代码无注释，也就罢了。如果找到，则便获得其所在位置（由此行代码开头起算）。这时候便可运用 Range 对象的 SetRange method 设置一个新范围，涵盖整个注释区。要知道，每一个 Range 对象都有 Start 和 End 两个 properties，分别表示第一个字符和最后一个字符的位置（由文档开头起算）。只要设置这两个 properties，就是重新定义 Range 对象的涵盖范围（如图 10-35）：

```
#035     RgnStart = lineProgramRange.Start
#036     RgnEnd = lineProgramRange.End
#037     lineProgramRange.SetRange Start:=RgnStart + CharPos,
#038                           End:=RgnEnd
```

其中 RgnStart 和 RgnEnd 都是我声明的 Long 变量。



图 10-35 重新设置一个 Range 对象的涵盖范围

接下来圈选上述这个“涵盖整个注释区”的 Range 对象，设其字体颜色为蓝：

```
#039     lineProgramRange.Select
#040     Selection.Font.ColorIndex = wdBlue           '令注释为蓝色
```

下面是新版宏，灰底部分是新增加的变量和程序代码：

```
#001 Sub 程序代码加行号()
#002 Dim nLineNum                      '行号(数值)
#003 Dim sLineNum As String            '行号(文字)
#004 Dim selRge As Range              '用以记录整区块程序代码(多行)
#005 Dim i                            '计数器
#006
#007 '以下变量用于“将注释改为蓝色”
#008 Dim lineProgramRange As Range    '单行程序代码范围
```

```

#009 Dim commentRange As Range          '注释范围
#010 Dim TextLine As String            '每一行程序代码（内容）
#011 Dim ProgComment As String         '每一行程序代码内的注释文字
#012 Dim RgnStart As Long             'Start of a Range
#013 Dim RgnEnd As Long              'End of a Range

#014
#015 Set selRge = Selection.Range      '首先记录 Selection
#016 For nLineNum = 1 To selRge.Paragraphs.Count   '针对每个段落
#017     sLineNum = str(nLineNum)           '行号转为文字
#018     sLineNum = LTrim(sLineNum)        '去除字符串开头和结尾的空格
#019     For i = 1 To (3 - Len(sLineNum)) '补足三位数（前加‘0’）
#020         sLineNum = "0" + sLineNum
#021     Next i
#022     sLineNum = "#" & sLineNum        '前加符号‘#’
#023     sLineNum = sLineNum + " "        '后加空格
#024
#025     selRge.Paragraphs(nLineNum).Range.InsertBefore (sLineNum)
#026                                         '将行号插入于前
#027
#028     '将注释改为蓝色
#029     Set lineProgramRange = selRge.Paragraphs(nLineNum).Range
#030     TextLine = lineProgramRange.Text      '取得整行文字（代码）
#031     CharPos = InStr(1, TextLine, Chr(39)) '寻找注释起始点
#032
#033     If CharPos <> 0 Then
#034         ProgComment = Mid(TextLine, CharPos + 1) '切割出注释文字
#035         RgnStart = lineProgramRange.Start
#036         RgnEnd = lineProgramRange.End
#037         lineProgramRange.SetRange Start:=RgnStart + CharPos,
#038                                         End:=RgnEnd
#039         lineProgramRange.Select
#040         Selection.Font.ColorIndex = wdBlue          '令注释为蓝色
#041     End If
#042 Next nLineNum
#043 End Sub

```

VBA10-17 加行号，并将注释设为蓝色

10.5. 实例二：两岸术语转换

我的很大工作是写译计算机书籍，这些书籍从 2000 年开始，同步在台湾和大陆出版繁体版和简体版。大家或许不知道，经过 50 年分治，两岸计算机术语早已自成体系，分道扬镳②。差异之大，绝对让初次接触彼岸计算机书籍的读者寸步难行。

这是事实，没什么好说的了。作为一个计算机书籍的写译者，又在两岸出版，

我能够做的，就是在繁体版和简体版之间转换术语。不，不是转换内码（BIG5 和 GB），那种简单事情交给机器就行了，转换术语才是需要脑袋瓜的工作。

过去我的工作方式是，就着逐渐累积的经验所整理出来的一份两岸术语对照表，逐一针对其中所列术语，对整份文档进行【编辑→替换(R)】动作。这份对照表内有 500 个以上的语汇，于是我要手工进行 500 次【编辑→替换(R)】动作。以前我的 Word 排版功力不够，一本书分成三个文件（采用罗马编页的文前、采用阿拉伯编页的正文、以及采用双栏格式的索引），这又大大增加了【编辑→替换(R)】动作的次数。

如果能够一个按钮做掉所有这些工作，那可真是阿弥陀佛。

Word 2000+有繁简转换功能，并有两岸用语转换，的确可以一个按钮做掉所有转换工作。但两岸用语并不是一个萝卜一个坑那么简单。举个例子，大陆有一本《程序员》杂志，虽然“程序”在台湾称为“程式”，“程序员”在台湾称为“程序员”或“程式师”，但总不好把人家一本杂志径自改名为《程式师》吧？Word 2000 就会这么硬干！又例如繁体的“程序员”被转换后成为简体的“程序员”，再转为繁体后竟成了“程式师”。此外，两岸用语中的范本↔模板，文件↔文档，档案↔文件，物件↔对象……，有些被 Word 转了，有些没转！这不能怪 Word，毕竟它只站在辅助立场，其转换字典无法与时俱进走在最前端，可以理解。

我的构想是：反求诸己，利用原先备妥的中英繁简术语对照表¹³，设法以 VBA 程序读出其中内容并区分出英文术语、台湾术语、大陆术语，再录制一个宏，观察 Word 的【编辑→替换】动用哪些 Word 对象，然后照着葫芦画瓢。术语转换完毕后，如果需要转换内码（BIG5→GB），再使用 Word 2000+的繁简（或简繁）转换，并将术语转换功能关闭：点击【工具→语言→中文简繁转换】，在其对话框（图 10-36）中取消勾选转换时包括词汇。

¹³ 我把它开放于侯捷网站 <http://www.jjhou.com/terms.htm>

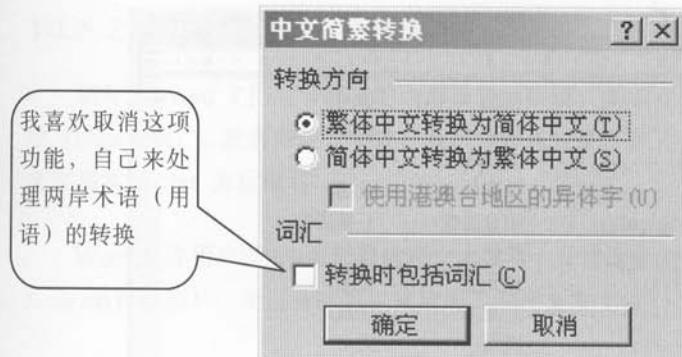


图 10-36 简繁中文互转

10.5.1. 文本文件格式设计

这样构想下的中英繁简术语对照表（文件），内容长相如图 10-37。

图 10-37 刻意运用 Word 的 **显示/隐藏** 按钮 展现其语汇分隔符号：所有分隔符号都是空格（space），没有制表符（tab）。这样的设计是为了简化将来宏对字符串的处理，因为 VBA 提供的字符串处理函数只有表 10-3 所列数个，以仅有的这些函数来处理如图 10-38 的各种“spaces、tabs 夹杂”的情况，将会相当麻烦。毕竟这只是个私人用的宏，并无商品化企图，而术语对照表又是由宏使用者（我）来维护，无乱造之虞，因此我选择了一条捷径，对文件格式做出较为严格的规定（也就是只允许 space 而不允许 tab 作为分隔符号），避开太大的弹性所带来的令人厌烦的工作。



图10-37 中英繁简术语对照表 文件格式一

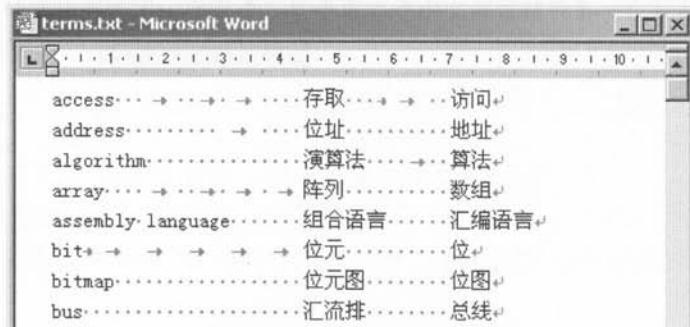


图10-38 中英繁简术语对照表 文件格式二

10.5.2. 读取文本文件内容

相对于 Word 文档有着繁复的内部设置，所谓“文本文件”意指只有单纯的文字（plain text），没有颜色、字体、大小、特殊效果、样式……额外设置。这样的文件通常以 .txt 为后缀名。

Word 允许用户以 VBA 读取这种文本文件。其读取方式是顺序读取（除非另以 Seek() 函数控制），所以有时候也被称为“顺序文本文件”。

Word 允许用户同时开启 4 个文本文件。开启后的文本文件可以阅读、增加内容、修改、删除内容。开启模式就读写来区分，有 Read、Write、ReadWrite 三种，就其他角度而言，有 Append、Binary、Input、Output、Random 五种。

开启一个文本文件的做法是：

```
Open filename For open-mode #file-code
```

例如：

```
Open "d:\jjweb\terms.txt" For Input As #1
```

文本文件的关闭方式是：

```
Close #file-code
```

例如：

```
Close #1
```

读取已开启的(顺序)文本文件的方式有：`Input()`、`Input #`、`Line Input #` 等方法。本例用到的是：

➤ `Line Input #`：读取一行（以换行字符为据）。例如：

```
Dim TextLine As String  
  
Open "c:\text.txt" For Input As #1  
Line Input #1, TextLine
```

便是从 #1 文件读取一行文字，放到 TextLine 中。

文本文件的结束符号是：

```
EOF(#file-code)
```

例如：

```
Open "c:\text.txt" For Input As #1      '开启文件
Do While Not EOF(1)                  '文件尚未到尾端
...
Loop                                '继续
Close #1                            '关闭文件
```

10.5.3. 字符串处理

VBA 的字符串处理函数并不丰富，总共也就表 10-3 里这么小猫两三只，连不以字符串处理著称的 C/C++ 标准程序库也比它强。

表10-3 VBA 字符串处理函数一览

作用	关键字 函数名	说明
字符串比较	StrComp	N/A
转换字符串	StrConv	可转换为大/小写、单/双位元组、Unicode……
大小写转换	Format, Lcase, Ucase	N/A
建立“重复字符” 字符串	Space, String	N/A
计算字符串长度	Len	返回 long
设置字符串格式	Format	星期、月日、时间等特殊格式
重排字符串	LSet, RSet	N/A
除了 InStr 返回 Variant(Long) 其余皆返回 Variant(String)	InStr	返回某字符串在另一字符串中的最先出现位置。如果没有找到，返回 0 InStr([start,]string1, string2 [compare])
	Left	返回某字符串由左算起之 n 个字符所构成的子串 Left(string, length)
	LTrim	消除前头（左）空白 LTrim (string)

作用	关键字 函数名	说明
	Mid	返回某字符串中指定区域内的字符所构成的子串 <code>Mid(string, start [,length])</code>
	Right	返回某字符串由右算起之 n 个字符所构成的子串 <code>Right(string, length)</code>
	RTrim	消除后头（右）空白 <code>RTrim (string)</code>
	Trim	消除前后（左右）空白 <code>Trim(string)</code>
设置字符串 比较规则	Option Compare	N/A
运用 ASCII 与 ASCII 值	Asc, Chr	Chr(n) 返回 ASCII 码值为 n 的字符

先前我已提过转换术语表文件格式的考虑（图 10-37）。因此我们知道，每一行文字有三个语汇，第一个是英文术语，第二个是台湾术语，第三个是大陆术语。每个语汇间隔不定数量的空格（spaces），其间不可有任何 tabs¹⁴。取出这三个语汇的做法是：

- 将整行去除头尾空白
- 寻找整行文字中的 space 首次出现处，以其左侧为英文术语，右侧为中文术语。
- 去除中、英文术语的头尾空白
- 寻找中文术语中的 space 首次出现处，以其左侧为台湾术语，右侧为大陆术语。
- 去除台湾术语、大陆术语的头尾空白
- 将三个术语储存到相应的 `String` array 内备用。如果是繁转简（代号 1），就以台湾术语为 `FindString`，大陆术语为 `ReplaceString`。如果是简转繁（代号 2），则以大陆术语为 `FindString`，台湾术语为 `ReplaceString`。

¹⁴ 嫌“按 Space 比按 Tab 慢太多”的急性子朋友，可进入控制面板，利用  将按键重复速度调到最快◎

VBA10-18 是相关程序片段。

```

#001 Sub 繁简术语转换()
#002     Convert (1)
#003 End Sub
#004
#005 Sub 简繁术语转换()
#006     Convert (2)
#007 End Sub
#008
#009 Sub Convert(kind As Integer)
#010 ' 这个子程序给 繁简术语转换() 或 简繁术语转换() 调用
#011
#012 Dim FindStrings(2000) As String          '搜寻字符串 (将被替换)
#013 Dim ReplaceStrings(2000) As String        '替换字符串
#014
#015 Dim ETerm As String                      '英文术语
#016 Dim CTerm As String                      '中文术语 (含台湾术语和大陆术语两个语汇)
#017 Dim TaiwanTerm As String                 '台湾术语
#018 Dim MainlandTerm As String                '大陆术语
#019 Dim TextLine As String                   '从文件读内的整行文字
#020
#021 Dim SpaceChar                           'space
#022 Dim CharPos As Variant                  '搜寻所得的字符位置
#023 Dim i                                    '数组索引
#024
#025 Open "c:\windows\desktop\terms.txt" For Input As #1
#026 i = 0
#027 SpaceChar = Chr(32)
#028
#029 Do While Not EOF(1)
#030     Line Input #1, TextLine
#031     TextLine = Trim(TextLine)             '去除前后 space
#032
#033 CharPos = InStr(1, TextLine, SpaceChar) '寻找整行第一个 space
#034 ETerm = Mid(TextLine, 1, CharPos - 1)
#035 CTerm = Mid(TextLine, CharPos + 1)
#036
#037 ETerm = Trim(ETerm)                     '去除英文术语前后 space
#038 CTerm = Trim(CTerm)                     '去除中文术语前后 space
#039
#040 CharPos = InStr(1, CTerm, SpaceChar)   '寻找中文术语第一个 space
#041 TaiwanTerm = Mid(CTerm, 1, CharPos - 1)
#042 MainlandTerm = Mid(CTerm, CharPos + 1)
#043
#044 TaiwanTerm = Trim(TaiwanTerm)           '去除台湾术语前后 space
#045 MainlandTerm = Trim(MainlandTerm)        '去除大陆术语前后 space

```

```

#046
#047 If kind = 1 Then           '繁转简
#048     FindStrings(i) = TaiwanTerm
#049     ReplaceStrings(i) = MainlandTerm   '搜寻台湾术语
#050 Else                         '转换为大陆术语
#051     FindStrings(i) = MainlandTerm
#052     ReplaceStrings(i) = TaiwanTerm    '简转繁
#053 End If                       '搜寻大陆术语
#054
#055 ...          '这里进行文字替换动作（见 10.5.4 节）
#056
#057     i = i + 1
#058 Loop
#059 Close #1
#060
#061 End Sub

```

VBA10-18 将读入的一整行文字切分为三个语汇

10.5.4. 以查找和替换对话框为师

从术语转换表（文件）中正确取出所有中英繁简术语后，接下来的活儿就非常简单了：只要将一对一对的 FindStrings 和 ReplaceStrings 喂给 **查找和替换** 对话框即可。图 10-13 和 VBA10-1 便是录制了“将文档中的“软体”全部转为“软件””这样一个宏，作为我们的施工蓝图。只需将 VBA10-1 的函数内容复制到 VBA10-18 尾端的 “...” 处，再将其中两行（如下）：

```

.Text = "软体"
.Replacement.Text = "软件"

```

改为以下即可：

```

.Text = FindStrings(i)
.Replacement.Text = ReplaceStrings(i)

```

这样替换出来的结果，其文字格式完全依旧，原本是斜体的、粗体的、带下划线的、带底纹的、带框的、带特殊颜色的……，统统保留原样，只有文字内容被转换。（还记得吗，我们曾经在上一个例子遇到“格式跑掉了”的湍流）

10.5.5. 需要脑袋瓜的地方

上述转换并非十全十美，至少有两个缺点：

- 不经意的错误转换。例如台湾称 process 为“行程”，大陆称之为“进程”，那

么文档中的“执行程式”会被转换为“执进程式”。又例如台湾称 memory 为“记忆体”，大陆称之为“内存”，于是大陆版文档中的“……之内存在着……”会变成台湾版的“……之记忆体在着……”。

- 与术语转换表中的次序相依。例如：文件（台湾） \Leftrightarrow 文档（大陆），档案（台湾） \Leftrightarrow 文件（大陆）。那么如果先把所有的“档案”转换为“文件”，再把所有的“文件”转换为“文档”，就错了。如果先把所有的“文件”转换为“文档”，再把所有的“档案”转换为“文件”，就对了。

10.6. 实例三：列出文档使用的所有字体

如果您的文档有印刷用途（例如书籍），就需要制版。过去采用照像制版，管它字体有什么变化，眼见为凭统统拍摄下来就是了。如今虽然品质进步为直接从输出机输出网片，工作上的准备却退步到“输出机必须装有文档所用的一切字体”¹⁵。

字体的准备工作该是谁的责任呢？我搞不清楚。大概是输出中心吧，不都说作者最大嘛②，而且这里面可能也有字体版权的问题需要厘清，大概出版社或输出中心才能够处理这些庶务。然而，虽说输出中心应该会有库存或装机的各式各样令人观止的字体，如果文档提供者能够列出一个清单，让人家有所依循有所准备，想必有助于降低未来可能的往返沟通（这种沟通愈少愈好）。

Word 有所谓的 `Font` 对象，其上有 `Name` 和 `NameFarEast` 两个 properties，前者用以表现内置字体（多为英文字体）名称，后者用以表现远东字体名称。现在我们遇到两个问题：

- 如何获得整份文档所使用的每一个 `Font` 对象？
- 获得一大堆字体名称后，如何只保留不重复者？

10.6.1. 关于 Collection

有一种数据结构名为 `Map`，又名 `Dictionary`，作用就像字典或电话簿一样，让人得以根据某个键值（Key）快速查询对应之实值（Value）。Word 也提供这样的

¹⁵ 如果输出机上没有安装文件所使用的字体，输出时会选择默认字体。

东西，并把它称为 Collection¹⁶。

Collection 的一个很大特色是，既然有所谓的 Key，那便该是独一无二的，不允许重复。我们似乎可以利用这个性质，将字体名称（字符串）视为 Key 丢进 Collection 中，就保证绝不重复了。

于是我查了查资料，找出 Collection 的声明方式和用法：

```
Dim fonts As New Collection
```

此行的意思是声明一个 fonts Collection，并产生一份实体。接下来我便可以使用 Collection 提供的三个 methods 和一个 property：

- *Add method*: 可加入任何东西（字符串、数值、对象……）：

```
<Collection Name>.Add <Item>, [<Key>], [<Before>], [<After>]
```

其中的元素（item）不需隶属同一种资料型别（此与 array 不同），但通常我们会放进相同型别的元素。Key 必须是个独一无二的字符串。如果没有指定它，日后您将无法直接（根据 Key）取回元素，只能根据位置索引（1,2,3…）来取回元素，就像 array 那样（但是别忘了，元素的位置会随着新元素的插入而变化）。

Before 和 After 都用来精确控制被插入元素在 Collection 中的位置。通常我们不在乎这些，这里就不说了。详请参考 VBA 手册。

- *Remove method*: 移除某个元素。其中的 Index 可以是 Key，也可以是位置索引（1,2,3…）。

```
<Collection Name>.Remove <Index>
```

- *Item method*: 取回某个元素，或设置某个原已存在的元素的值。其中的 Index 可以是 Key，也可以是位置索引（1,2,3…）。

```
<Collection Name>.Item <Index>
```

- *Count property*: 返回目前的元素个数。

```
<Collection Name>.Count
```

¹⁶ 我习惯把 Collection 译为“群集”。很多人把这个字眼视为一般（任何种类）的容器泛称，C++社群就是如此。至于 Word 这里所说“具备 Key 和 Value”的数据结构，C++社群称之为 Map。

因此我便试着这么做：

```
#001 Dim fonts As New Collection      '字体集
#002
#003 For i = 1 To 10          '取前 10 个 Style 对象
#004     fonts.Add ActiveDocument.Styles.Item(i).Font.Name
#005     fonts.Add ActiveDocument.Styles.Item(i).Font.NameFarEast
#006 Next i
#007
#008 Dim msg As String
#009 Dim obj As Variant
#010 For Each obj In fonts          '把所有字体名称集合起来
#011     msg = msg + obj + Chr(10)    '每一个名称后加换行符
#012 Next
#013 MsgBox msg                     '输出消息框
```

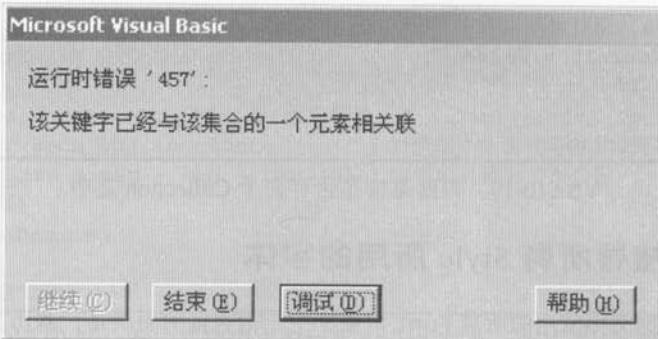
以某个文档为操作对象，获得结果如下。似乎 Collection 并没有把“Key 必须独一无二”放在心上⑧。



慢，刚才我根本没有指定 Key 嘛！于是修改 Add 动作，以字体名称作为 Key：

```
fonts.Add ActiveDocument.Styles.Item(i).Font.Name, _
ActiveDocument.Styles.Item(i).Font.Name
```

现在 VBA 果然发现重复的 Key 了，可它竟然悍然结束了程序：



这真令人抓狂。如果它只是默默地拒绝，让程序继续运行，多好！¹⁷

至此，现成的功能是应付不来了，我必须自行设计解法。但若因此而大张旗鼓似乎划不来，我采用最直接最暴力的办法：每有新物欲加入 Collection，就把 Collection 遍历一遍，看看欲加人物是否已存在于其中¹⁸。下面就是“判断某物是否存在于 Collection 之中”的程序：

```
#001 Function NotInIt(str As String, _
#002                      ByRef coll As Collection) As Boolean
#003 ' 接受一个 String 和一个 Collection。
#004 ' 如果该 String 在该 Collection 内，返回 False
#005 ' 如果该 String 不在该 Collection 内，返回 True
#006
#007 Dim notIn As Boolean
#008 notIn = True
#009
#010 '顺序检验 - 最笨的手法！
#011 Dim obj As Variant
#012 For Each obj In coll
#013     If obj = str Then
#014         notIn = False
#015     Exit For
#016 End If
#017 Next
#018 If notIn Then
#020     NotInIt = True
```

¹⁷ 人家 C++ Map 就是这么做的

¹⁸ 就此观之，已经没有非使用 Collection 不可的理由。array 一定可以胜任。

```
#021 Else
#022     NotInIt = False
#023 End If
#024
#025 End Function
```

VBA10-19 判断某物是否存在于 Collection 之中

10.6.2. 搜寻所有 Style 所用的字体

想要得到文档使用的所有 `Font` 对象，一个很直接的想法是，遍历文档中的所有 `Style` 对象，取其 `Font` 对象，然后取其 `Name`（内置字体，多为英文字体）和 `NameFarEast`（远东字体，吾人环境下为中文字体），分别加入 `efonts` 和 `cfonts`：

```
#001 Sub 列出所有字体()
#002 ' 遍历所有 styles
#003
#004 Dim efonts As New Collection      ' 英文字体集
#005 Dim cfonts As New Collection      ' 中文字体集
#006 Dim stl As style
#007 Dim useCount As Integer
#008
#009 useCount = 0
#010 For i = 1 To ActiveDocument.Styles.Count      ' collection 从 1 开始
#011     Set stl = ActiveDocument.Styles.Item(i)    ' 系统中的每一个样式
#012     If stl.inuse Then
#013         useCount = useCount + 1
#014         If NotInIt(stl.Font.Name, efonts) Then   ' 字体名称首次出现
#015             efonts.Add stl.Font.Name            ' 加入它
#016         End If
#017         If NotInIt(stl.Font.NameFarEast, cfonts) Then ' 首先出现
#018             cfonts.Add stl.Font.NameFarEast        ' 加入它
#019         End If
#020     End If
#021 Next i
#022
#023 ' 以下显示所有信息
#024 Dim msg As String
#025 Dim obj As Variant
#026
#027 msg = "目前所有 Styles: " + CStr(ActiveDocument.Styles.Count) _
#028             + Chr(10)
#029 msg = msg + "目前被使用的 Styles: " + CStr(useCount) _
#030             + Chr(10) + Chr(10)
#031
#032 msg = msg + "内置字体: " + CStr.efonts.Count) + Chr(10)
#033 For Each obj In efonts                 ' 把所有内置字体名称合并起来
```

```
#034     msg = msg + obj + Chr(10)      '每个名称后加换行符
#035 Next
#036
#037 msg = msg + Chr(10) + "远东字体: " + CStr(cffonts.Count) + Chr(10)
#038 For Each obj In cffonts           '继续加所有远东字体名称
#039     msg = msg + obj + Chr(10)      '每个名称后加换行符
#040 Next
#041 MsgBox msg                      '显示
#042
#043 End Sub
```

VBA10-20 遍历文档内含的所有 Style 对象，取其 Font 对象

获得的结果是：



图10-39 遍历所有 Style 对象，获得的所有字体名称

10.6.3. 搜索所有 Character 所用的字体

但是，仔细想一想，遍历所有 Style 对象取其所用之 Font 对象，以此窥知文档使用了哪些字体，这种做法有个盲点。并不是文档中使用的所有字体都必须先在样式（style）上设置。不，不是每个人都有那么好的规律性，也不是每件事都必须

一板一眼。偶而我们会在少量地方直接以【格式→字体(F)】修改字体。通常只有面对次数频繁的修改，我们才会特别为它做出一个专属样式。

字体作用在字符身上，因此如果取文档内含的所有 **Character** 对象，再取后者的 **Font** 对象，应该就万无一失了。第 290 页告诉我们，Word 其实并未提供 **Character** 对象，只提供一个 **Characters** 对象，内含一大堆 **Range** 对象，每个 **Range** 指出一个字符(Character)范围，并带有一个 **Font** 对象指出该字符所用字体。

以下便使用这样的性质，取出所有字符的字体。如此一来文档中的字体再无漏网之虞！由于每个字符不是英文就是中文（在我们目前的环境下），其字体名称不是存在于 **Name**（内置字体）就是存在于 **NameFarEast**（远东字体）中，不像 **Style** 对象那样同时设有内建和远东两种字体，因此我们不该分两个 Collections 储存，应该把它们统统放在一起：

```
#001 Sub 列出所有字体()
#002 '
#003 ' 遍历所有 Characters
#004
#005 Dim fonts As New Collection      '字体集
#006 Dim rgn As Range
#007
#008 For Each rgn In ActiveDocument.Characters
#009     If NotInit(rgn.Font.Name, fonts) Then      '字体名称首次出现
#010         fonts.Add rgn.Font.Name                '加入它
#011     End If
#012     If NotInit(rgn.Font.NameFarEast, fonts) Then '首次出现
#013         fonts.Add rgn.Font.NameFarEast          '加入它
#014     End If
#015 Next
#016
#017 '以下显示所有信息
#018 Dim msg As String
#019 Dim obj As Variant
#020 msg = "目前所有 Characters: " _
#021     + CStr(ActiveDocument.Characters.Count) + Chr(10)
#022 msg = msg + "目前所用字体: " + CStr(fonts.Count) _
#023     + Chr(10) + Chr(10)
#024 For Each obj In fonts            '把所有字体名称合并起来
#025     msg = msg + obj + Chr(10)    '每个名称后加换行符
#026 Next
#027 MsgBox msg                      '显示
#028
#029 End Sub
```

VBA10-21 遍历文档内含的所有 `Characters` 对象，取其 `Font` 对象

执行结果如下：

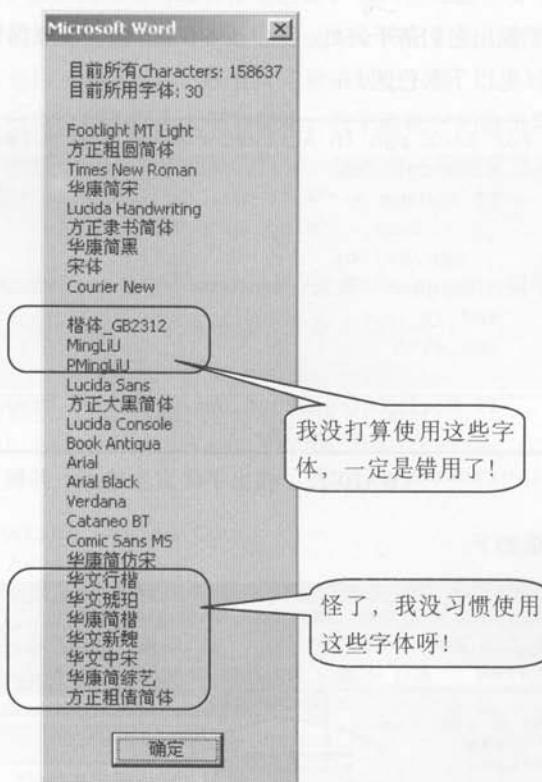


图10-40 遍历所有 `Character` 对象，获得的所有字体名称

上图是在 Windows98 + Word97 + 512MB RAM + Intel Centrino CPU (1200 MHz) + 30GB HardDisk 的环境下，花费 27.2 秒所得的结果。这个时间满尴尬的，等待颇长，离开又不能干啥。这只是本书写作当时的 158,637 个字符，如果以我曾经译过的最大部头书《C++ Primer》3/e 而言，高达 1,504,386 个字符，那可有得瞧了。冲杯咖啡去吧。

10.6.4. 后记：字体发生处

这一节是无心的收获。

我发现图 10-40 列出了一些我很陌生的字体，某些是我过去曾经用过的字体，

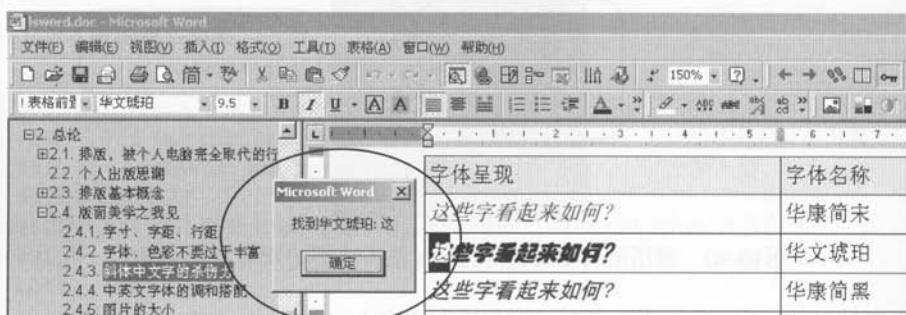
某些是我从来就不主张在正式文档中使用的字体。那么，是不是我在书中错用了它们？

为了找出它们落于何处，我修改 VBA10-21，在取得每一个 Character 之后检查其字体（见以下灰色区）：

```
#001 For Each rgn In ActiveDocument.Characters
#002     With rgn.Font
#003         If (.Name = "华文琥珀") Or (.Name = "华康简综艺") Or _
#004             (.Name = "MingLiU") Then
#005             rgn.Select
#006             MsgBox "找到" + .Name + ":" + CStr(rgn.Text)
#007     End If
#008     End With
#009
#010    If NotInIt(rgn.Font.Name, fonts) Then   '如果字体名称首次出现
#011        ... '以下同 VBA10-21
```

VBA10-22 找出字体发生地点，并跳至该处

执行结果如下：



至于这些字体是不是错用了，当然只有作者（或排版者）明白。

10.6.5. 列出所有样式的说明文字

表 1-2 曾经列出所有样式的说明文字。这些文字很繁琐，如果我采取“从每一个 **样式** 对话框中抄出说明栏中的文字”，那真是太暴力太不优雅也太累人了。其实可以运用宏完成这项烦人的工作。

解决这个问题的关键在于，我们知道 style 样式有一个 InUse 属性，可让我们判知此样式是否在文档中被使用。另有 NameLocal 和 Description 两个

properties，前者表现样式名称、后者表现样式的细目描述。这些正是我所需要的。

但是，在 VBA 程序中如此轻易取得细目描述之后，如何呈现这些文字使我得以轻松获得它们进而将它们放到我的书稿中呢？以 MsgBox 将它们输出到消息框并不切实际，因为消息框不允许 copy/paste，致使我“看得到吃不到”。一个好的做法是，将这些细目描述输出到文件去。为求更加简化，我干脆把它们输出到书稿文档的最尾端，反正这只是暂时信息，用毕就要清理掉。输出每一段细目描述之前，我先输出一个换行符号 Chr(10)。

下面是相关的 VBA 程序代码。从结果中我注意到，InUse property 似乎没有发挥应有的作用，所有样式不管三七二十一都被这个宏取出细目描述。

```
#001 '找出每一个 styles，将其 description 写到文档最后
#002 ActiveDocument.Paragraphs.Last.Range.InsertAfter (Chr(10))
#003                                     + "Styles Description: ")
#004
#005 MsgBox Application.Styles.Count
#006 Dim nowStyle As style
#007 For Each nowStyle In ActiveDocument.Styles
#008     If (nowStyle.inuse) Then
#009         ActiveDocument.Paragraphs.Last.Range.InsertAfter
#010             (Chr(10) + nowStyle.NameLocal)
#011         ActiveDocument.Paragraphs.Last.Range.InsertAfter
#012             (Chr(10) + nowStyle.Description)
#013     End If
#014 Next
```

VBA10-23 将所有使用样式 (inUse Style) 的细目写入文档最尾端

第11章 Word 电子文档
本章将介绍如何使用Word的电子文档功能，包括超链接、书签、批注、追踪修订、保护电子文档等。

11. Word 电子文档.....	331
11.1. 绪论.....	331
11.2. 超链接 (Hyperlink)	332
11.3. 书签 (Bookmark)	335
11.4. 批注 (Comment)	338
11.5. 追踪修订 (Track Changes)	339
11.6. 电子文档的保护.....	343
11.6.1. 批注保护.....	344
11.6.2. 修订保护.....	345
11.6.3. 开启保护.....	345
11.6.4. 防写保护/只读保护	346

11.6.4. 防写保护/只读保护

如果希望对Word文档进行保护，防止他人修改，可以在“文件”菜单中选择“另存为”，在“另存为”对话框中勾选“只读”或“写入权限受限制”。如果只是希望对Word文档进行保护，而不希望他人修改，可以在“文件”菜单中选择“另存为”，在“另存为”对话框中勾选“写入权限受限制”。

如果希望对Word文档进行保护，而不希望他人修改，可以在“文件”菜单中选择“另存为”，在“另存为”对话框中勾选“写入权限受限制”。

11

Word 电子文档

11.1. 绪论

作为文档，一般人直觉想到的是，将来会被打印到纸面上。但在电子时代这已经不是“绝对”（甚至不一定是“占大多数”）的行为了。是的，文档虽然可能成为书面文档，也非常有可能以电子（数字）方式保存或传布。

把 Word 文档开放到外部环境（公司或组织之外、作者或出版社之外、撰稿者或杂志社之外……），比较少见。首先因为 Word 文档流传出去几乎相当于程序源码流传出去，我们将无法确定我们的文档是否被擅改。其次，Word 文档并不是一个便携式文档，它在不同的软硬件环境上的呈现可能会有一点点小差异¹。正因为这两个因素，外流文档往往选择便携的、不易被改动的、具备压缩能力、有保护功能（密码）的 PDF 格式（第 12 章）。

然而，撇开外流的需要，研究单位、公司或组织内部的确很可能直接以 Word 文档交流。这其中涉及（1）文档可能是由团队共同完成、（2）文档在自己公司内部无“被恶意修改或破坏”之虞。事实显示，Word 电子文档的确在许多公司或组织内部有不小的流通量。再如指导教授和研究生共同撰写论文，彼此之间流通的也是可编修的 DOC 文档（如果采用 Word 平台的话）而不是 PDF 文档。

¹ 举个例子。如果我的合作出版社采用 DOC 文档进行输出，就让我比较头痛。往往对方输出的样稿，字距比我这边的原始情况小一点点（挤一点点）。原本孤字成行的情况可能经过微调之后消失了，整个段落少了一行。这就可能大幅影响整个版面，甚至影响索引的正确性。由于此情况中的字距是小一点点（幸好）而不是大一点点（那就糟了），版面只会缩短而不会膨胀，因此对付“分页点错乱”这个问题，只要我在文件中必要的每一页尾强制加上分页符号，基本上也就没有问题。但无论如何这个问题总使我必须于正式印刷前校对一次（甚至两次）蓝图，以确保所有断页和版面都按当初的设想乖乖摆放。这就是一种非常让人不能安心的 dirty work ☺。

电子文档的需求和用量日益庞大，因此 Word 提供的一些针对电子文档的功能也就很值得去认识并理解。当然，其中和超链接（hyperlink）有关的性质，对书面文档是完全无用的。

11.2. 超链接（Hyperlink）

超链接和大家可能已经很熟悉的 WWW 网页上的超链接功能完全相同，主要是为了让电子文档阅读者可以轻松地对内跳到文档某个定点，对外跳到某个网址（URL, Universal Resource Locator）。这对即时参照式的阅读有很大的协助作用。超链接只存在于电子文档中，不存在于书面文档。

当然啦，“跳过去很容易”很好，但也得要跳得回来才更好。在 HTML 网页中没有问题，因为浏览器有“回到上一页”的功能。Word 文档却没有这种方便，因为 Word 没有“回到上一页”的功能。

下面是制作超链接的实例。本书自序最后有一份署名如下：

侯捷 2003/12/01 于台湾新竹
jjhou@jihou.com
<http://www.jjhou.com> (繁体) <http://jjhou.csdn.net> (简体)

其中的一个电子邮箱和两个网址都是超链接。点击上述电子邮箱会出现图 11-1，点击上述网址则出现您电脑上装妥的网页浏览器（Internet Explorer 或 Navigator 之类）并链接至该网址。

欲做出前述的网址超链接，首先圈选标的文字，然后点击【插入→超级链接(I)】，出现图 11-2。如果您所圈选的标的文字本身就是网址形式（例如上图的 <http://www.jjhou.com>），会被 Word 自动填入图 11-2 上方的文字方块。如果标的文字不是网址，图中的文字方块便是空白，等待您输入一个正确型式的网址。

如欲做出上图的邮箱超链接，过程差不多。但图 11-2 上方文字方块的邮箱地址之前必须加上 "mailto:"，这是针对邮箱超链接必须填写的一个语汇。以本例而言，我应该填写 <mailto:jjhou@jihou.com>。

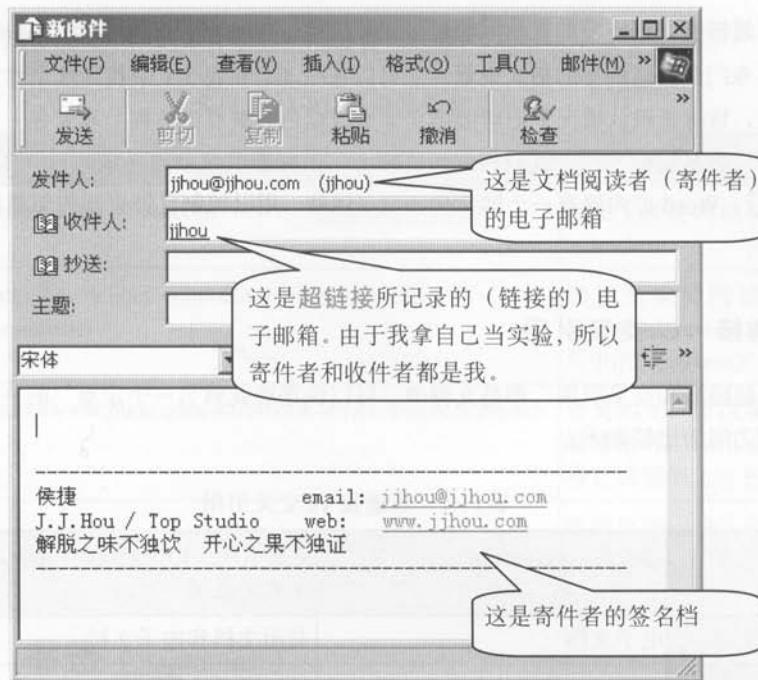


图11-1 超链接自动调用 email 软件，并填妥正确的收件者

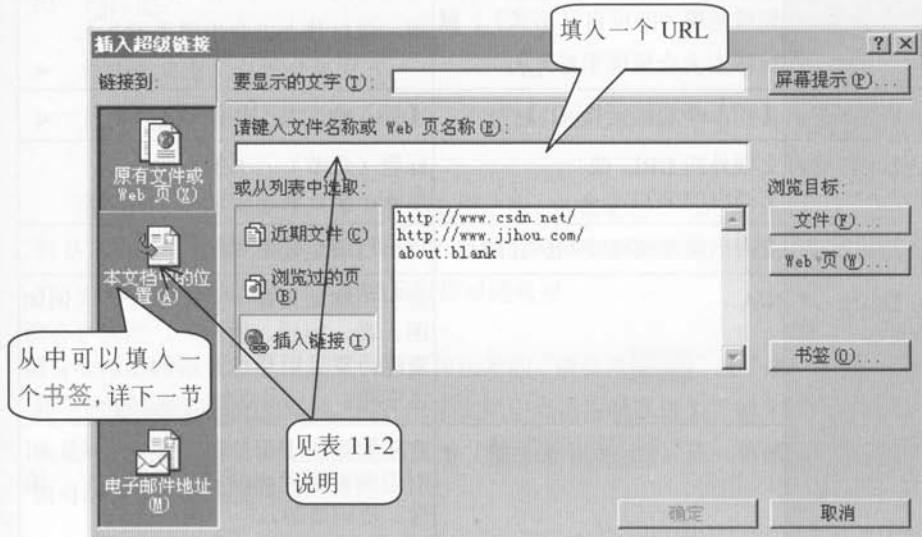


图11-2 插入超级链接 (hyperlink)

通常超链接的呈现应该和一般正文有所区别，才能让读者知道那是一个可以点

击的超链接。超链接的显示风貌可由用户决定：Word 内建的 **a 超级链接** 字符样式便是专门负责超链接的格式设置。您可以在图 5-1 和表 5-1 中找到这个样式并加以修改。该样式默认情况下把超链接文字变成蓝色（被点击后则呈现紫色），并加下划线。将来鼠标光标一旦被移到超链接上，鼠标形状便会由一般的箭头  变为手形 。Word 亦内建有一个**已访问的超级链接**，用以控制已被查阅的超链接的呈现样貌。

超链接 vs. 交叉引用

超链接和交叉引用，都是文档中“可以快速跳跃到另一个定点”的一种设施。它们之间的比较如下：

表11-1 超链接 vs. 交叉引用

	超链接 (Hyperlink)	交叉引用 (Cross Reference) 详见 5.12 节
适用于	电子文档	书面文档和电子文档
表现方式	直接以文档中的某页面元素（文字、图片……）为超链接载体。通常该页面元素默认以蓝色+下划线呈现（亦可自订格式）。鼠标移上去会呈现手形  。	以“请参考图 x-x”或“请参考第 n 页”中的图 x-x 和 n 作为载体。是个域，可选择以正常方式或灰底方式显现。鼠标移上去会出现手形  。
制作方式	【插入 ➔ 超级链接 (I)】	【插入 ➔ 交叉引用(R)】
链接/引用对象	文档外的 URL 或 文档内的书签	标题（章节）、书签、脚注、 尾注、图、表
操作与效果	点击后跳至被链接物所在地点	点击后跳至被引用物所在地点
特色 1	N/A	被引用物如果是个自动编号体（例如图、表、标题、题注），当编号有所变化，交叉引用这一端的显示亦会随之变化。
特色 2	N/A	交叉引用所引用的范围弹性极大，可引用到被引用物的标签（label）、编号、页码等等。

下表展示图 11-2 中的两个文字方块的设置和结果：

表11-2 URL 与书签的相互关系

内容设置（例）	意义及结果
URL: http://www.jjhou.com 书签:	链接至侯捷网站首页，index.htm
URL: http://www.jjhou.com/officebook-review.htm 书签: comment5	链接至侯捷网站中的 officebook-review.htm 网页中的“comment5”书签
URL: http://www.jjhou.com/officebook-review.htm#comment5 书签:	意义同上。下次编辑此超链接时，Word 会自动将它调整如上一栏
URL: 书签: comment5	链接至同一份文档中的“comment5”书签

11.3. 书签 (Bookmark)

Word 书签就像真实世界中的书签一样，是文档编写者可在文档任何地点加上的一种标记：

- 作为将来交叉引用时使用。
- 是文档阅读者可藉以快速定位的一个凭据。此一功用只存在于电子文档。

很显然，书签主要用来作为交叉引用之用，也就是用来“从 A 引用（快速跳跃）到 B”，A 是引用点而 B 是被引用点。我们需要在 B 处做一个书签，在 A 处做一个……唔……你说对了，一个交叉引用或超链接。

这么想来，前数章以交叉引用制作出来的“请参考图 x-x”或“请参考表 x-x”，其中的图 x-x 和表 x-x 都是“可在电子文档中点击后跳跃过去”的东西，它们都应该是一种引用载体，而跳过去的目标地（图 x-x 和表 x-x）是一种被引用物，也应该可以说是一种书签啰？

是的，图表题注也是一种书签。不论它们在台面上或台面下有没有相同的称谓，题注和书签拥有不少相同的性质。

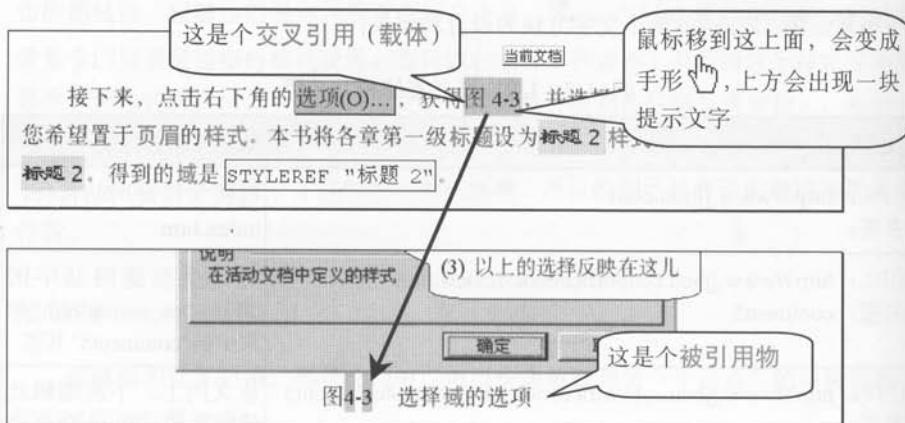


图11-3 图表题注其实就一种书签

现在我们来看看如何制造书签。首先将输入点移到您希望插入书签的地点，然后点击【插入→书签(K)】，获得画面如下。左图显示非隐藏（用户自定）书签，右图表示所有隐藏式书签：

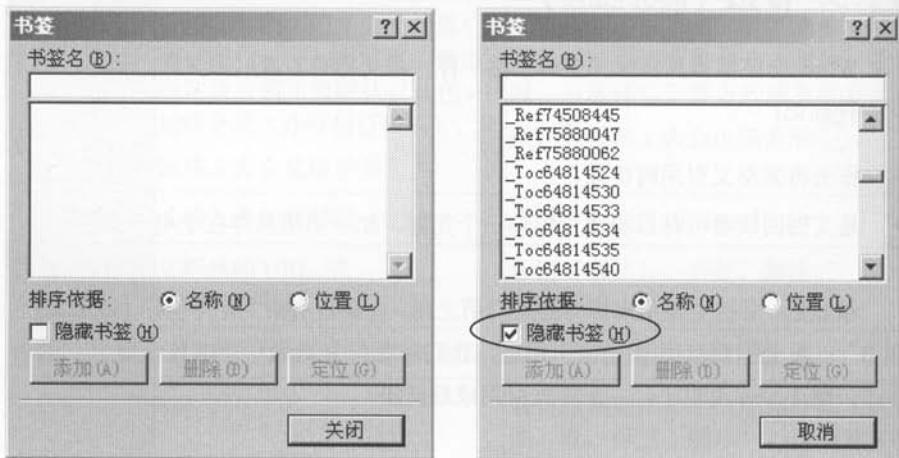


图11-4 书签对话框

在书签名(B):文本框中输入您想取的名称，于是添加(A)可按。点击添加(A)，一个新书签就成立了。



如果您的书签只是要在Word文档(或将来转制作的PDF文档)中使用，由于中文版Word处理中文相当稳定，因此以中文命名书签没问题。

但如果您有可能将 Word 文档转为 HTML 文档,由于 Word 书签会被转为网页 (Web Pages) 上的书签,而浏览器对于中文书签的处理比较令人忧心,因此最好采用英文来命名书签。

图 11-4 右图显现出许多隐藏式书签。它们都是些什么东西呢?我特别调节那个画面,让它展现 Ref 和 Toc,那正是题注和目录的域代码(见表 9-1)。也就是说,文档中的每一个题注和目录项(通常是章节标题)都是一种书签!

有了书签,任何地点只要插入超链接或交叉引用,便可快速参考(跳跃)到书签所在。见图 11-2 和图 5-44。

书签和题注(及章节编号)都是文档中“可被快速跳跃到达”的一种设施。它们之间的比较如下:

表 11-3 书签 vs. 题注和章节编号

	书签 (BookMark)	题注 (Caption) 和章节编号
适用于	电子文档	书面文档和电子文档
表现方式	屏幕上看不出来。只能在【插入→书签(K)】所得的书签对话框中看到所有书签名称,并以定位(G)查看每个书签的落脚处。	<ul style="list-style-type: none"> ■ 题注以“图 x-x”或“表 x-x”呈现。是个域,可选择以正常方式或灰底方式显现。 ■ 章节编号以“x 章”或“x.x 节”的形式呈现。
制作方式	【插入→书签(K)】	【插入→题注(C)】或多级别章节编号(5.11 节)
链接者/引用者	超链接	交叉引用
特色	N/A	题注和章节编号乃是由 Word 自动编号。一旦编号有变化,交叉引用端的显示亦随之变化。

11.4. 批注 (Comment)

批注的英文字眼是 comment², 意指在一份合作开发的文档中, 阅读者对他人所写的内容的看法和意见。这是一种互动式作业, 只存在于电子文档, 不存在于书面文档。

假设某份文档由 CHEN 和 HOU 两人共同负责。HOU 审阅 CHEN 的稿件时有一些意见想表达, 这就可以用上批注。首先圈选欲批注文句, 然后点击【插入→批注(M)】，画面变成图 11-5。主视窗分裂为上下两半。上半段是文档内容, 下半段是批注视窗, 视窗中会列出检阅者名称并自动对批注编号, 也就是图中的[jjhou1] 和[jjhou2], 其样貌由批注引用控制。批注文字的样貌由批注文字控制。被批注文句会被 Word 加上黄色底纹: 当前 (current) 批注是深黄色, 其他批注是浅黄色。这一颜色设置是无法修改的。请注意, HOU 这个人写批注时把日期时间也加了进去 (他是点击【插入→日期和时间(T)】加进去的)。这是很好的作业习惯, 显示此人受过良好的训练, 是一条训练有素的……嗯。

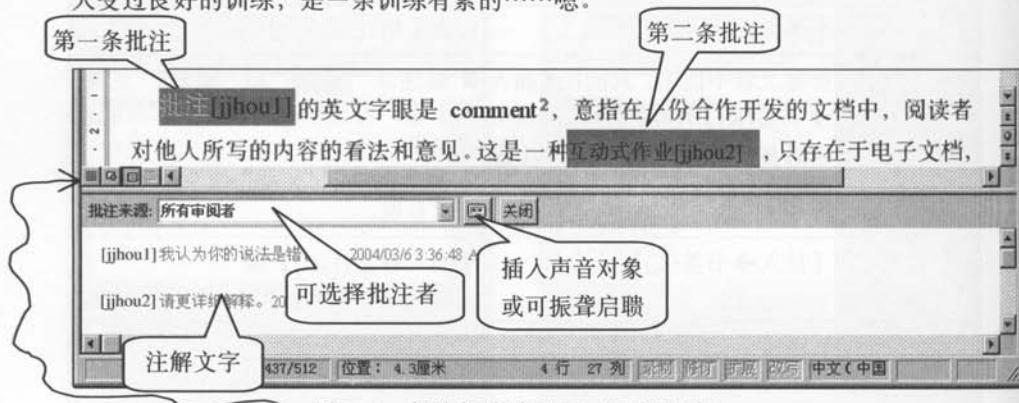


图11-5 批注视窗出现了（下侧视窗）

一旦批注完毕, 将分裂视窗的“分隔棒”拉到底 (也就消灭了分裂视窗), 或是点击批注视窗上方的关闭钮, 文档视窗即恢复原状, 如图 11-6。此时文档中的被批注文句除了留下浅黄底纹, 再无其他痕迹。如果用户将鼠标移到黄底纹上面, 会有一个文本框弹出批注 (并伴随一段发人深省的声音——如果曾有声音对象被插入

² 也许把 comment 译为“意见”比较更贴近原意。不都说 “no comment” 就是“没意见”、“不置可否”吗？

的话）。如果要修改批注文字，只需点击【右键菜单】的【编辑批注】即可令批注视窗重现江湖。

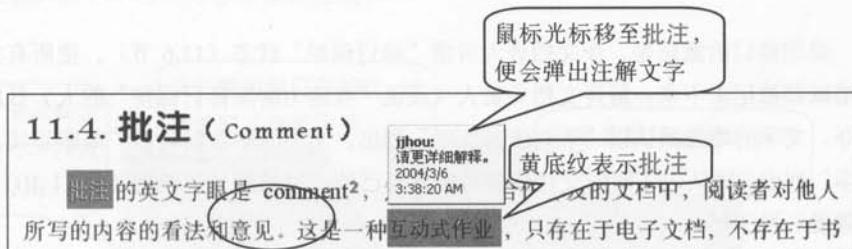


图11-6 带有批注的文句，以黄底纹展现

表 11-4 另有一些对于批注的说明。

11.5. 追踪修订 (Track Changes)

 读者来函：我是一个研究生，经常要处理一些长文档，排版自然是问题，可我还有排版以外的两个问题。

第一个问题是：当我写好文档，要请别人检阅时，总是会遇到多人协作的问题。目前我只能把原文复制一份给检阅者，他以不同色彩或字体标识出错误或意见。返还给我后，我再拿来和我的原文比对。这样一来会出现很多份副本，时间长了都不知道哪个版本最新；二来比对时很麻烦，有时甚至要开两个视窗（我总是抱怨我的屏幕太小）。我想知道Word是否支持多人协作功能呢？

第二个问题是：就像源代码版本管理一样（例如 CVS 或 VSS），Word 能不能提供一个“修订的历史记录”（不只是 Undo/Redo），在我需要的时候 Rollback 呢？这样我就可以不必再翻出众多备份，找到我想要的某句话再 copy/paste 过来了◎

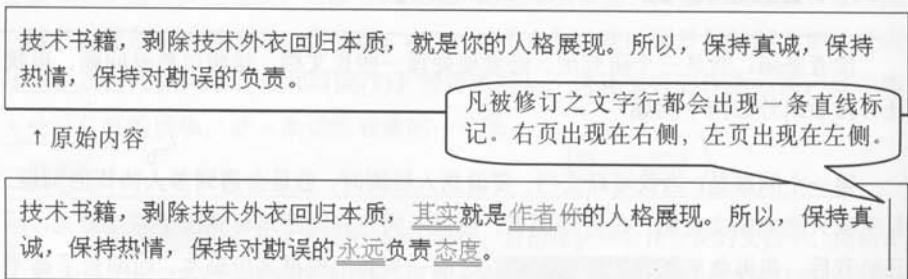
 侯捷简答：多人协作的情况愈来愈频繁地出现在各种文档的开发上。首先是学术论文，大部分时候至少是研究生+指导教授 2 人的协作。其次是公司各种书面报告、技术报告、白皮书、产品使用手册、作业规范……Word 以批注（comments）、追踪修订（Track Changes）、版本比较（Version Comparison）来支持多人协作，前者见上一节说明，后二者见本节说明。

以团队合作方式来发展文档，过程中需要频繁地沟通协调、同僚复审。在沟通

修改的过程中，除了上一节所说的批注（comments），还有一项利器可用，那就是追踪修订。

追踪修订的意思是，令文档进入所谓“修订保护”状态（11.6 节），使所有文字增减都被记录下来，留待文档负责人（或说“有能力解除修订保护”的人）日后定夺。文字的增加默认以“下划线及红字”列出，文字的删除默认以“单删除线及红字”列出。团队内不同的文档作者可设置自己的文字增删显现风格（图 11-10），使观者一目了然。

图 11-7 便是一张花花绿绿的追踪修订文档画面。其中有 4 段新增文字和 1 段删除文字。这些修订记录都是在“修订保护”（11.6 节）下产生的。将来解除保护（恢复正常编辑状态）后，便可逐一接受或拒绝这些修订（稍后详述）。



↑ 追踪修订。此前利用图 11-10 将文字插入样式设为“蓝色+双下划线”，文字删除样式设为“红色+单删除线”。

图11-7 追踪修订实例画面

点击【工具→修订→突出显示修订(H)】，可决定修订记录显现与否，如图 11-8。

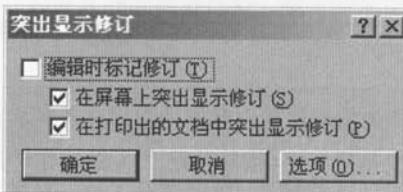


图11-8 追踪修订对话框

当文档进入修订保护状态，获得了追踪修订记录后（如图 11-7），一旦解除修订保护状态（需要密码，见 11.6 节），用户便可点击【工具→修订(T)→接受或拒绝修订(A)】，在图 11-9 中逐一确认各项修订。如果接受，修订记录就正式纳入文档，

修订标记（那些红蓝字啦、下划线啦、删除线啦）自动消失。如果拒绝，修订记录便被抹除，不对文档带来实质影响。

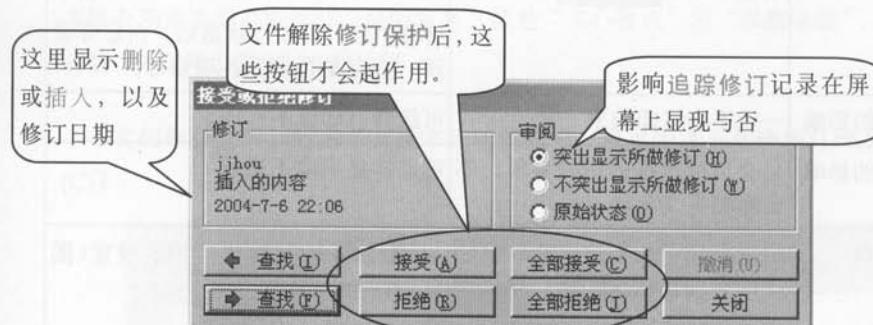


图11-9 接受或拒绝修订对话框

表 11-4 列出批注和追踪修订之间的比较。由于其中谈到两种文档保护状态：“批注保护”和“修订保护”，所以在阅读这个表时，或许您也可以跳阅 11.6 节“电子文档的保护”。

表 11-4 批注和追踪修订的比较

	批注 (Comments)	追踪修订 (Track Changes)
对文档的影响	只以“旁注”方式提供意见，并未修改内容。	修改（增加或删除）了内容。但还需经过确认才能被正式纳入。
插入方式	【插入→批注】	当文档处于“修订保护”（11.6 节）状态下，任何文字增减便会自动形成追踪修订效果，其他编修动作（例如格式变化）则不形成追踪记录。
插入时机	任何时候都可以插入批注。如果文档处于“批注保护”状态，用户就只能插入批注，无法做其他编修动作（此时许多菜单项都失效变成灰色）。	只有当文档处于“修订保护”状态，文字增减才会形成追踪修订效果。
删除方式	在批注（黄色底纹）身上点击【右键菜单】的【删除批注】。	在“修订保护”状态下，由修订者处理：如果对新增文字反悔，可以像平常一样拿掉；如果对删除文字反悔，却不能真正拿掉它。 解开“修订保护”状态后，由定夺者逐一决定是否接受各项修订。

	批注 (Comments)	追踪修订 (Track Comments)
编辑方式	在批注 (黄色底纹) 身上点击右键菜单的 编辑批注	在“修订保护”状态下, 由修订者自由编辑。 解开“修订保护”状态后, 由定夺者逐一决定是否接受各项修订。
对打印的影响	一定不会打印出来	可选择打印或不打印
对显示的影响	只会以提示框的形式呈现, 不会影响主文布局。	可选择显示或不显示
呈现样貌	批注视窗中的批注引用和批注文字分别由 批注引用 和 批注文字 两个样式加以控制	由【工具→选项】+ 修订 内的设置(图 11-10) 加以控制
补充说明	当文档处于“批注保护”状态下, 用户无法进行任何修订。	当文档处于“修订保护”状态, 用户还是可以插入批注。

点击【工具→选项(O)】+**修订**, 可看到图 11-10:

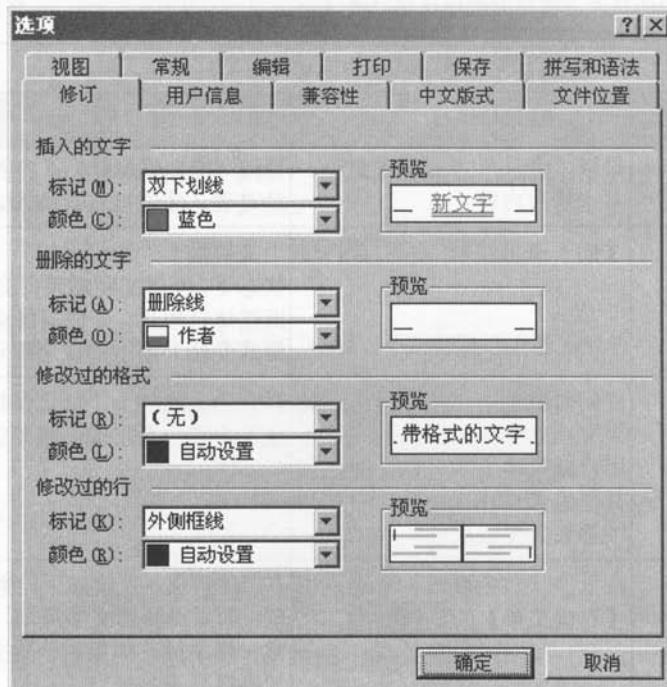


图11-10 设置个人的追踪修订记录显现风格

图 11-10 可为个别用户设置个人化的修订记录显现风格。针对插入的文字、删除的文字、修改过的格式、修改过的行，用户可分别设置修订记录的颜色和标记（下划线、双下划线、斜体、粗体、删除线）。Word 的其他格式工具栏上还有许多适合用来表现追踪修订记录的符号，像是“实心批点”和“双删除线”，可惜没有被 Word 用于图 11-10 供选择。

文档修订之后，将来若有版本比较的需求，可点击【工具→修订→比较文档(C)】：



图11-11 Word 文档版本比较

Word 的文档版本比较可以到达什么程度呢？劝您不必寄以厚望②。我试了几次之后还是归纳不出太多心得。在这一点上身为程序员的我比较能够体贴地为 Microsoft 着想，不做太多诘难。因为，恐怕连我们用户自己，都提不出来到底我们要的是什么样的文档比较、希望获得什么样的比较结果。

11.6. 电子文档的保护

Word 电子文档的保护共有五种：窗体保护、修订保护、批注保护、开启保护、防写（只读）保护（第 12 章的 PDF 也有后两种保护）。我个人对于后四种保护有实际经验，本节就来谈谈这四种保护。

11.6.1. 批注保护

11.4 节曾经提过，所谓批注是“以旁注方式对文档加上的意见”。如果您的文档传布出去之后只允许被加上批注，不允许其他任何修改动作，那么可将文档设为批注保护。只要点击【工具→保护文档(P)】+批注 即可，如图 11-12。图中的密码可设可不设。如果设了，将来解除保护时也需回答密码，如图 11-14。

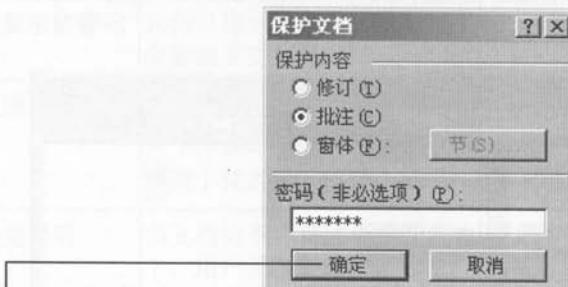


图11-12 文档保护：修订保护、批注保护、窗体保护

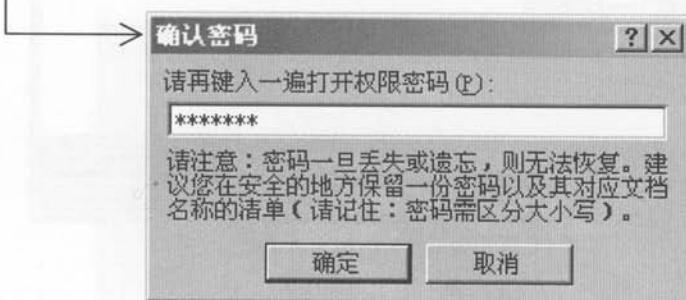


图11-13 文档保护之密码确认

一旦文档被设置为批注保护，所有可能造成文档内容改变的 Word 操作都将失效（disable）。如果用户还是去操作它们，Word 状态列（status bar）会出现如下信息：“本指令不能使用，因为本文档已在编辑锁定状态。”

在批注保护状态下，菜单项目【工具→保护文档(P)】会转变成【工具→解除文档保护(P)】。点击它之后，如果先前未曾设置密码，那么就是直接解除保护。如果先前曾经设置密码，便出现图 11-14 要求填写密码。

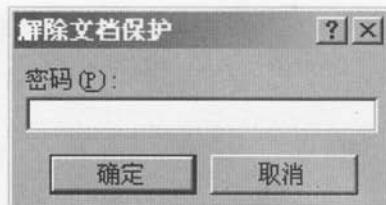


图11-14 文档保护的解除

11.6.2. 修订保护

如果您所传布的文档允许他人修订，但希望在真正落实之前获得您的确认，那么可将文档设为修订保护。只要点击【工具→保护文档(P)】+修订即可，见图 11-12。图中的密码可设可不设。如果设了，将来解除保护时也需回答密码，如图 11-14。

一旦文档被设置为修订保护，所有造成文档内容改变的 Word 操作（包括文字的增加、删除、格式变化），都将留下追踪修订记录。这些修订记录的表现形式由图 11-10 设置。在修订保护状态下也可以【插入→批注】。

在修订保护状态下，【工具→保护文档(P)】会变成【工具→解除文档保护(P)】。点击它之后，如果先前未曾设置密码，那么就直接解除了保护。如果先前曾经设置密码，便出现图 11-14 要求填写密码。

解除修订保护之后，用户才能逐一确认（接受或拒绝）每一项修订，如图 11-9。

11.6.3. 开启保护

点击【文件→另存为】+工具(L)+常规选项，可看到图 11-15。图左下方有一个打开权限密码(P)文本框，用来设置开启保护。只要填写它，将来任何人打算开启这份文档时，必须先有通关密语，如图 11-16。如果这份文档还曾经被设置防写保护，那么在通关密码之后 Word 还会询问您是否愿意保持只读状态？或是希望进入读写状态？如图 11-17。当 Word 询问开启保护密码时，如果您回答的是防写保护（修改权限密码）密码，没有用！

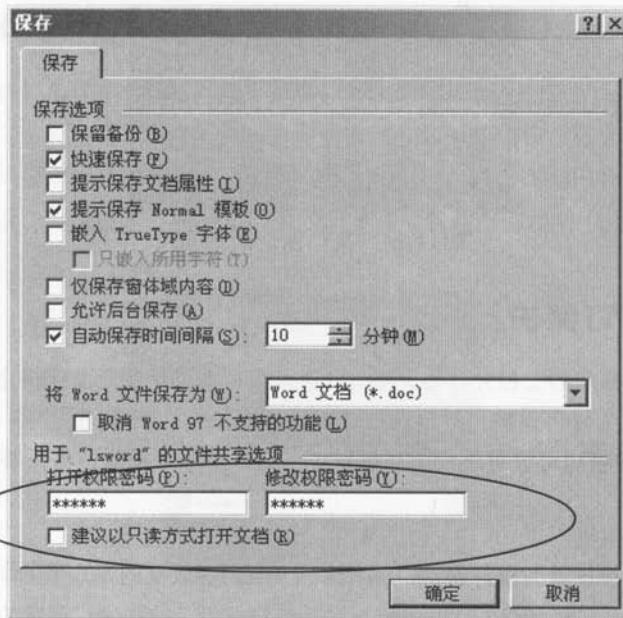


图11-15 【文件→另存为】+【工具(L)+常规选项】

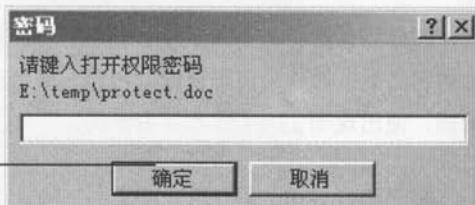


图11-16 文档之开启密码

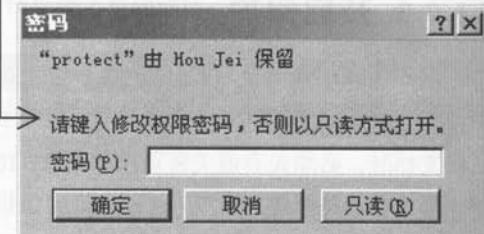


图11-17 防写（只读）模式之解除

11.6.4. 防写保护/只读保护

点击【文件→另存为】+【工具(L)+常规选项】，可看到图 11-15。图右下方有一个修改权限密码(Y)文本框，用来设置防写保护。只要填写它，将来任何人打开这

份文档时都是处于只读状态。这种状态下还是可以编修文档内容，但无法点击【文件→储存】，必须改用【文件→另存为】。

12. PDF 电子文档	349
12.1. 绪论	349
12.2. 便携式的条件	350
12.3. PDF 简介	351
12.4. PDF Reader	352
12.5. PDF Writer	352
12.6. PDF Distiller (蒸馏器)	352
12.6.1. 首选项	354
12.6.2. 字体位置	355
12.6.3. 作业选项：分辨率、压缩、字体、颜色	355
12.6.4. 安全性设置	357
12.6.5. 水印设置	358
12.6.6. 保留 Word 文档的书签、超链接、交叉引用	360
12.6.7. 一只臭虫	360
12.7. Adobe Acrobat	360
12.7.1. 安全性设置	362
12.7.2. 页面操作	362
12.7.3. 编排页码	366
12.7.4. 书签+导览窗格	369
12.8. PDF Maker	373

12

PDF 电子文档

12.1. 绪论

您经常遇到这种问题吗：

- 好不容易赶完一份重要文档。透过电子邮件传给同事，他们却无法开启，因为他们的软件版本太旧，或操作平台不同。
- 好不容易整理出绝佳的企划方案和相关图表。将文档传送给同事审阅，最后收到审阅结果时，却发现图表和版面布局都变更了。
- 将令人耳目一新的手册草案透过网络传给大家进行评论。在同事的电脑上文字歪歪斜斜，图像无影无踪，特别选用的优雅字体被替换成默认字体。只好将设计文案打印出来给同事看。同事则回应涂鸦般的批注和一大叠附件。

这就是“非便携式电子文档”带来的噩梦。

Word 从未宣称其文档是一种“便携式”电子文档。诚不欺也！无人可以预料 Word 文档搬到不同的软硬件平台上会有怎样的版面变化。举两个简单的例子：

- 如果满版的某页末尾有个被作者刻意置入的分页符（也许是为了保证紧邻的下一个标题一定出现在下一页起头），却因不同平台对 Word 文档的微调影响，使该分页符被挤到下一页，那就造成下一页完全空白！平白多出一页当然是排版上的一大疏失，是个失败作品。
- 文档创作端电脑安装了许多漂亮字体，并且被很协调地整合到文档设计之中。但是阅读端却是一台字体贫乏的电脑，举目望去全是“宋体”（Windows 内置的少数中文字体之一）。这不仅影响美观，也可能影响阅读的正确性，因为不少文档会以不同的字体来表现不同的涵义（本书就是这样）。

能够在不同平台上保持金刚不坏之身，当可被称为“便携式/可移植”电子文档。

PDF (Portable Document Format) 是其中翘楚，那是 Adobe 公司的 Acrobat 软件的编修格式。由于太有名了，很多人知道 PDF(文件格式)这个名称，却不知道 Acrobat 才是 PDF 编修软件。Acrobat (套件) 让您得以将任何文档转换为 PDF 格式。只要有 PDF Reader，任何人均可在各种硬件和软件上开启您的文档，其画面与原本设计分毫不差，版面、字体、链接、影像均原封不动。

本章把焦点放在“PDF 转换，页面增减，书签(目录)的建立”上面，而非“Acrobat 对文档内容的编修能力”上。也就是说，本章焦点只放在我曾经有过的 Word+Acrobat 协同作战经验上。

12.2. 便携式的条件

作为一个在电子平台上流通的良好介面（格式），便携式电子文档应该具备以下优点：

- 在不同软硬件平台上有绝对相同的表现。亦即在不同的机器、不同的操作系统上都显现完全相同的样貌，绝不能变幻莫测。
- 所谓“展现完全相同的样貌”，并不要求阅读端的电脑上安装任何特殊字体或插件（add-in）工具。
- 允许进行文字查找。
- 具备压缩能力。以免网络上大胖子横行。
- 具备保全能力。允许设置不同层次的密码，区分可读、可改……权限。
- 普及度高。便携式电子格式本身的良窳固然会影响接受程度，凭良心说，厂商的实力和积极度也不容忽视。
- 有极佳的配套软件。如果便携式电子文档的制作软件本身就是优秀排版软件，自然最好；如若不是，至少必须有很方便的做法将其他排版软件所产生的文档转换过来。

哇咧，以上简直就是 PDF 的专门说帖。PDF 具备上述所有条件！

12.3. PDF 简介

PDF 的全名是 **Portable Document Format**（便携式文档格式），是 Adobe 公司开发的一种格式。配套软件是 **Adobe Acrobat¹**，本章简介其中的主要套件：**PDFReader**、**Distiller**、**Acrobat**、**PDFWriter**、**PDFMaker**。以下沿用这些英文名称。

PDF 主要是利用 PostScript 语言²中的影像模型来代表文字和图形；它对页面的描述与 PostScript 做法相同，会在一个选取区内放置“颜色”用以绘制页面，此举可免“遭遇不同装置和分辨率的影响”，因而可在各种硬件和操作系统环境中畅行无阻地携带使用。PDF 可吃进二进制数据和以 ASCII 编码的数据。为了缩小文件，PDF 支持 JPEG、CCITT Group 3、CCITT Group 4、ZIP、LZW 等工业标准压缩技术。

PDF 文件内包含“显示嵌入字体”或“替代字体”时所需的信息。是的，PDF 文件包含“文档内每一种字体”的字体描述器，负责描述字体名称、字符公制（character metrics³）、以及样式信息。如果检视端的电脑系统可提供文档所用字体，抑或字体被嵌入 PDF 文件中，那么用上的就是该字体。但如果字体未得供应，或未被嵌入，PDF 系统就运用特殊的 serif 或 sans serif “多重字模字体”来模拟该字体。这个解法可套用于 Type1 字体和 TrueType 字体⁴。至于符号（Symbol）字体或专家（Expert）字体，会自动被嵌入 PDF 文件内，或转换成图形。

“PDF 替代字体”并不会造成文档的重新格式化，因为从 serif 和 sans serif “多重字模字体”建构出来的替代字体会保留原字符的宽度和高度。

PDF 档还包含有交叉引用表格，可协助直接使用（而无需再计算）文件中的页面，或找出其他重要对象。正因为 PDF 文件内拥有这个被称为 xref 的交叉引用表格，所以检阅特定页面所花的时间，和文档总页数几乎毫不相关。

¹ “acrobot”的意思是“马戏班特技杂耍人”。

² PostScript 是一种“页面描述语言”（page description language），是一个威力强大，可用来创作复杂字体（如 PostScript 字体）、图型（如 EPS 图）的语言。

³ character metrics 意指字符型体的种种量测值、设计值，例如宽、高、上边缘距离、下边缘距离等等。译为“字符公制”实为过于生硬，并不妥当。

⁴ Type1 字体就是 PostScript 字体。至于 TrueType 则是 Adobe PostScript 字体以外的另一种字体科技。

PDF 档由图层架构而成：第一图层包含文档的文字和影像内容。第二图层包含增强功能，例如书签、超链接、文章信息。第三图层包含文件基本信息，例如字体信息，以及用以导引 PDF 文件的交叉引用表格。

正是由于这种图层式结构，我们得以在更换 PDF 页面的同时，继续保留已建立的链接（links）。由于更换页面时移除的是第一层的内容图层，而不是第二层的增强图层，所以增强信息仍然保留下。这可以节省“维持电子文档”所需的许多工作，因为用户不需要重新建立链接。

PDF 的功能是可以扩展的，也就是说我们可以透过插件，轻易将新功能增入文件格式中。这已经是相当高级的应用了，本章并不讨论它。

以下分别讨论五个 Acrobat 套件。

12.4. PDF Reader

PDFReader 是个免费软件，由 Adobe 公司提供。一般认为，PDF 格式的高度普及，除了其自身的优越性之外，和 PDFReader 的免费策略也有很大关系。许多网站上都可以看到这样一个免费下载标记：。

任何人均可从 Adobe 网站 (www.adobe.com) 免费下载 PDFReader。进入 Adobe 网站首页后，可在右上角选择网页所用语文（亦即进入各地代理商所建网页）。

12.5. PDF Writer

PDFWriter 和 Distiller 一样，都是可将电子文档转换为 PDF 格式的一种工具。稍后将对 PDFWriter 和 Distiller 做一番比较。

12.6. PDF Distiller (蒸馏器)

Acrobat 套件中的 PDFWriter 和 Distiller 两者都可以用来将外部文档转换为 PDF 格式。这两个工具程序都包括在 Acrobat 默认安装项目中。PDFWriter 是一个打印机驱动程序，可将其他软件的文件直接转换为 PDF 格式。Distiller 是可以将 PostScript 文件转换成 PDF 格式的工具。很多情况下，Distiller 的输出品质比 PDFWriter 的输出品质高，但 PDFWriter 的处理速度比 Distiller 快。如果只是要转换简单的商业备忘录或只含文字的文档，两者都适用。如果您需要精确控制转换过程，或如果您所转换的文档含有丰富的图形，Distiller 比 PDFWriter 适用。如果要在 PDF

档中嵌入“亚洲语字体”，也必须使用 Distiller。

Distiller 用来把输入资料“蒸馏”为 PDF 格式。程序主画面如图 12-1。蒸馏过程所采用的各种规格（分辨率、压缩比、字体等等），可在 12.6.3 节的作业选项中设置。

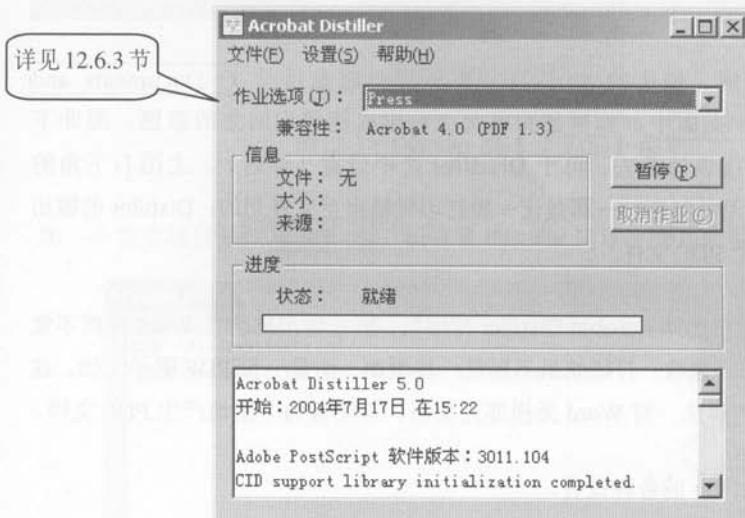
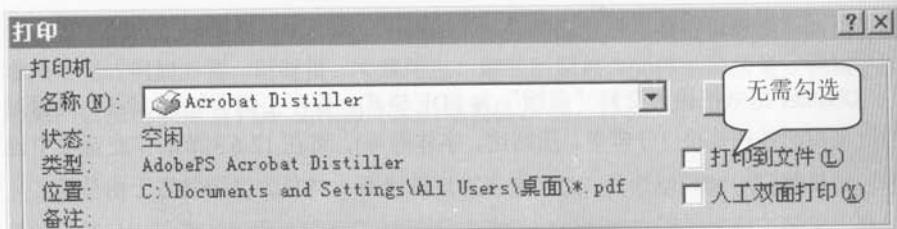


图 12-1 Acrobat Distiller 主画面

什么东西可以成为 Acrobat Distiller 的输入呢？Word 的.doc 文档就是，PowerPoint 的.ppt 文件也是，Excel 的.xls 文件也是。事实上几乎所有软件的打印机输出都可以是 Distiller 的输入。Distiller 的“蒸馏”概念是：将自己打扮成打印机，由“转换源”软件将文件内容输出至 Distiller 打印机，再由 Distiller 于幕后拦截“转换源”软件的打印机正常输出⁵，而后进行格式转换。

只要 Acrobat Distiller 安装妥当，任何 Windows 应用软件的【文件→打印(P)】所引发的【打印】对话框（如下）中的打印机名称便会出现 Acrobat Distiller 选项：

⁵ “在幕后拦截其他软件对打印机的输出？”这真是有点匪夷所思。Windows 系统编程领域极著名的一本书：《Windows 95 System Programming Secrets》by Matt Pietrek, Addison Wesley 1995（中译本《Windows 95 系统程式设计大奥秘》侯捷译，旗标 1997）书中第 10 章探讨了这种鬼斧神工。此书中英文版俱已绝版，感兴趣的读者可至侯捷网站（<http://www.jjhou.com>）免费下载完整中文电子版。



画面上可看到，输出的 PDF 文件默认置于磁盘目录 **C:\Documents and Settings\All Users\桌面** 中。如果您希望每次输出前能够询问您的意愿，而非不管三七二十一地往该处丢去，可于 Distiller 之中设置（详后）。上图右下角的 **□ 打印到文件(L)** 无需勾选——那是让一般打印机输出至文件用的；Distiller 的输出无论如何一定是个 PDF 文件！

一旦 Word 用户选择 Acrobat Distiller 打印机，所有输出都会在 Word 浑然不觉的情况下被 Distiller 截收。打印结果不再是一堆纸面，而是一份 PDF 电子文档。这是一种完全透通的解法，对 Word 无损亦无负担，却能极为方便地产生 PDF 文档。

以下介绍 Distiller 的各种设置。

12.6.1. 首选项

点击 Distiller 的【文件→首选项(P)】，即可得画面图 12-2。勾选其中的 **□ 询问 PDF 文件的目标**，可令 Distiller 于每次输出前询问输出地点（目录路径）。

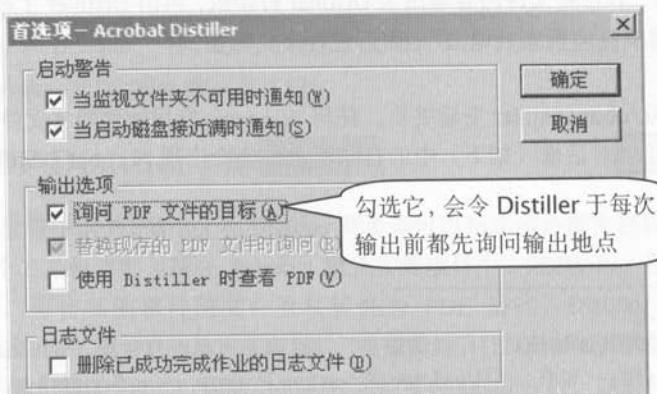


图12-2 Acrobat Distiller 首选项

12.6.2. 字体位置

点击 Distiller 的【设置(S)】，可看到图 12-3。

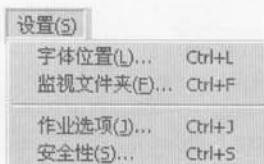
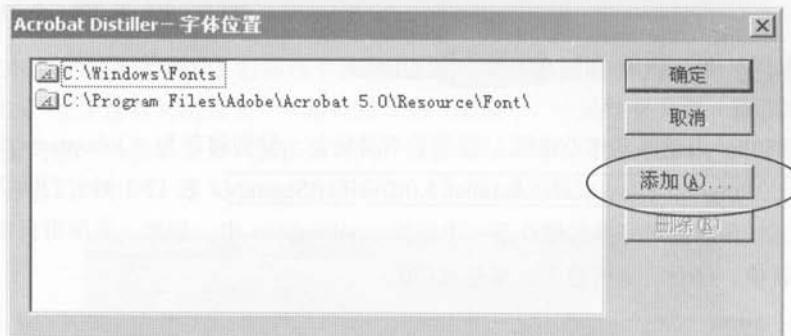


图 12-3 Distiller 的【设置(S)】菜单

第一个菜单项目是字体位置(L)，记录着 Distiller 应该上哪儿抓字体：



光是这样，Distiller 便已经可以将 Windows 系统所安装的字体都嵌入 PDF 文件中。这样是否符合版权精神？大概可以吧，我想这是厂商之间该去洽谈的事，不是终端用户的责任。

老实说，我不甚理解是否该在上图的对话框中加上 Windows 放置字体的磁盘目录 C:\Windows\Fonts。我总是加上它②

12.6.3. 作业选项：分辨率、压缩、字体、颜色……

Distiller 的【设置(S)】第三个菜单项是作业选项(J) (Job Options)，意指 Distiller 对于分辨率、兼容性、字体、压缩、颜色、以及其他高级属性的设置。下面是作业选项+一般画面。请注意，每一组作业选项可给予一个名称而后储存于文件中，方便以后再使用；如果您对任何作业选项做过任何改动，最好点击作业选项右方的另存为(A)，否则会盖掉原先的设置。下图是图 12-1 所示的

“Press” 作业选项的设置：

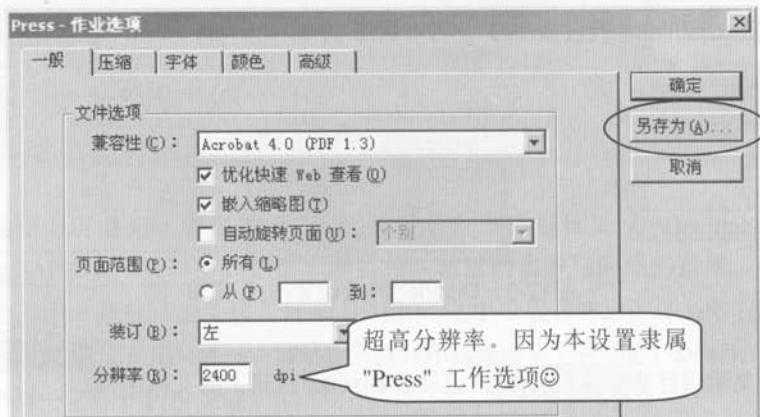


图 12-4 是 “Press” 作业选项的 **字体** 设置画面。

Distiller 内建四套作业选项，各用于不同场合，分别储存为 *joboptions 文件，位于 **C:\Program Files\Adobe\Acrobat 5.0\Distiller\Settings**。表 12-1 对它们做了一番整理。任何改动最好是另储存于一个新的 .joboptions 中，如此一来便可保留原先的默认值，这对将来的参考比对是有用的。

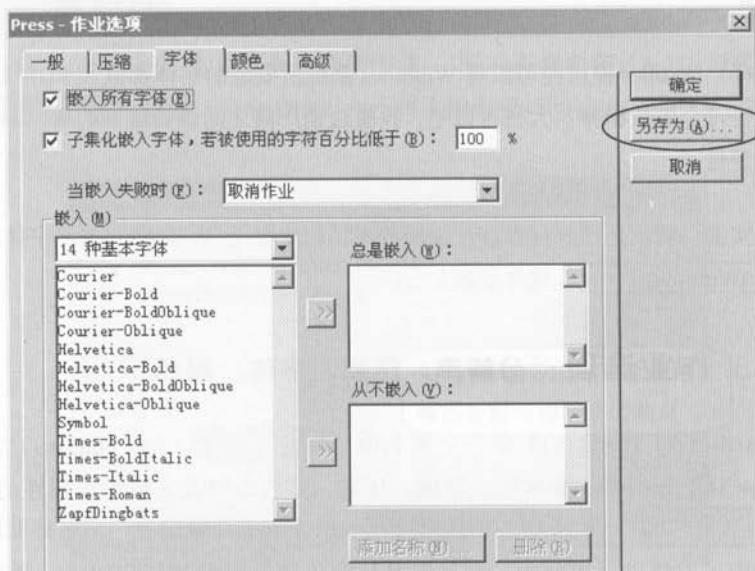


图 12-4 作业选项分为一般、压缩、字体、颜色、高级五大类

表12-1 Acrobat Distiller 的作业选项 (Job Options)

作业选项文件	用途	属性 (所有属性列于图 12-4)
CJKScreen.joboptions	中、日、韩文屏幕显示	分辨率 600。压缩。所有字体不嵌入。不嵌入缩略图。转换所有颜色到 RGB。
Screen.joboptions	屏幕显示	分辨率 600。压缩。所有字体不嵌入。不嵌入缩略图。转换所有颜色到 RGB。
Print.joboptions	打印机输出	分辨率 1200。压缩。嵌入所有字体。不嵌入缩略图。所有色彩管理标签不转换。
Press.joboptions	印刷	分辨率 2400。压缩。嵌入所有字体。嵌入缩略图。不变更颜色。

12.6.4. 安全性设置

Distiller 的【设置(S)】的第四个菜单项是【安全性(S)】(Security)，引发对话框如图 12-5。安全性有两层意义：一是针对开启（阅读），一是针对变更（修改）。后者涵盖前者——意思是如果您在被询问开启密码时，给的是变更密码，一样可以进入（开启）PDF 档。这和 Word 文档不太一样（11.6.3 节）。

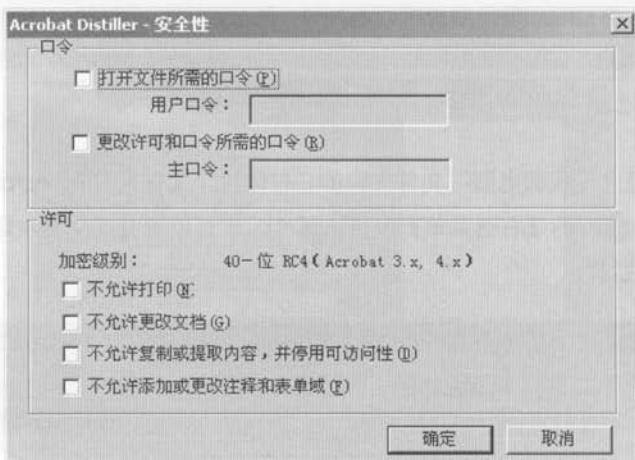


图12-5 Acrobat Distiller 的【安全性】对话框

我们还可以在上图中勾选“绝对不允许 PDF 用户执行”的某些事项。很多 PDF 电子书将打印功能取消，只允许在屏幕上阅读⁶（是一种宣传，也是一种良好而双赢

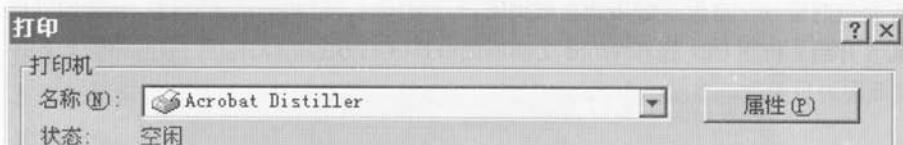
⁶ 道高一尺，魔高一丈，还是有破解法。世间法原是防君子不防小人。小人无法可防！

的商业模式），甚至不允许读者圈选其内的文字或图片（以防 copy/paste），这些设置都在图 12-5 对话框中完成。

12.6.5. 水印设置

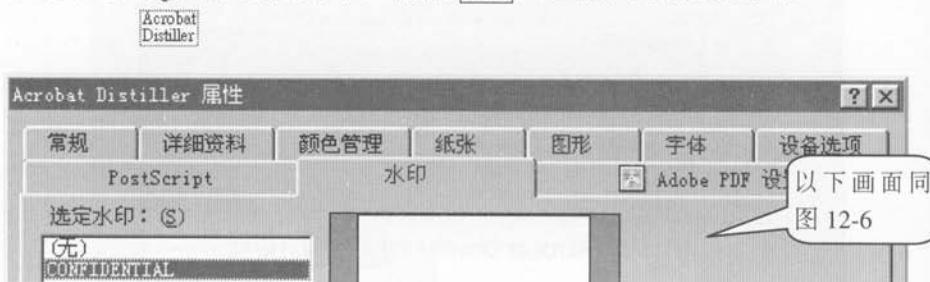
所谓水印，是指浮印在页面上的某种标记，可能是文字也可能是图案（但在 Distiller 中只能设置为文字）。水印既有某种标示作用，又（通常）不会影响文档在屏幕上的阅读。它虽然不如钞票的水印一样有防伪功能，但如果经过精心设计，也可以在文档被复印或被打印时带来干扰效果，对于防止盗版似乎也有那么一点点功效。

由于水印是 Distiller 提供的功能，您可能会直观地在 Distiller 菜单中寻找切入点。我也是这样的想法，但遍寻不着，最后才在 Word 打印对话框中获得切入点。是的，点击 Word 的【文件→打印(P)】取得打印对话框，在其打印机栏目中点击属性(P)：



便可获得图 12-6。该处可设置水印。

此外，进入“我的电脑”文件夹中的“打印机”文件夹，在 Acrobat Distiller 打印机图标 的【右键菜单】中点击属性，也可以查看或变更水印。



Distiller 默认的水印文字有 CONFIDENTIAL (机密)、COPY (副本)、DRAFT (草稿) 三种，各使用不同的颜色。点击图 12-6 的新建(W)，可增加新的水印文字；点击图 12-6 的编辑(E)，则可对这些水印文字加以编辑。不论新建或编辑，画面都如图 12-7，其中的描述十分直观，这里就不多作说明了。

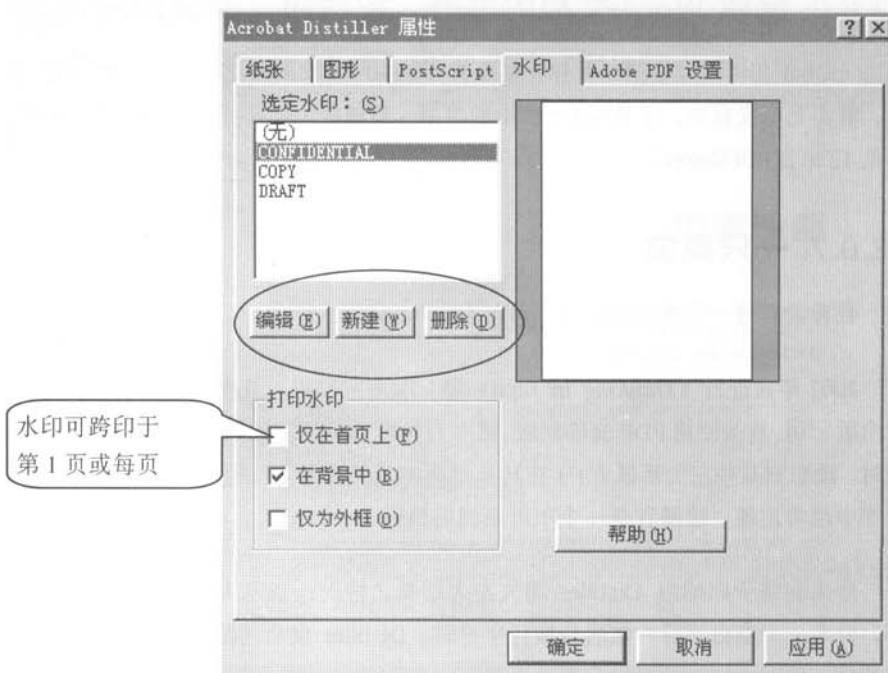


图12-6 在 Acrobat Distiller 打印机中设置水印



图12-7 新水印

12.6.6. 保留 Word 文档的书签、超链接、交叉引用

Acrobat 套件可以根据“大部分桌面排版软件所建立的文档目录”产生 PDF 书签。事实上不仅目录，连 Word 文档中的书签、超链接、交叉引用，统统可以保留。详见 12.8 节 PDFMaker。

12.6.7. 一只臭虫

我曾经有过一个深刻的经验，事关 Distiller 的臭虫。

2002 年我进行《Thinking in Java》第二版后制作时，首次尝试直接交付 PDF 给出版公司，直接使用 PDF 文件制版。然而当我将 Word 文档以 Distiller 转换为 PDF 档时，没想到某些左大括弧 ('{') 在转换过程中竟然遗失了！左大括弧 ('{') 在 Java 程序中经常出现，这简直是一个不可原谅的错误。

后来我终于归纳出 Distiller 遗失左大括弧之情况：当左大括弧之后无任何文字，只有一个空白字符，然后是换行字符时，Distiller 便会遗漏该左大括弧。但是在简短的测试文档中这只臭虫又不显现，无法重制。真是只棘手的臭虫！

12.7. Adobe Acrobat

Distiller 和 PDFWriter 都只是用来将其他类型的文档转换为 PDF 格式。如果有必要对 PDF 档进行页面编修，就需要使用 Acrobat。我所谓页面编修意指对 PDF 页面的插入、删除、裁切和替换，以及编排页码、建立书签（主要作为目录）、设置密码等动作。本节要谈的便是这些。

Acrobat 也允许团队成员以电子签名审批 PDF 文件，或使用电子批注、反白、删除线等工具，直接为 PDF 档增加注释（comments）。这些动作类似第 11 章所讨论的 Word 电子文档团队作业，但我不认为以 Acrobat 对文档进行内容编修是个好主意，因为 Acrobat 的内容编修功能远逊于 Word。

图 12-8 是 Acrobat 主画面。这个画面非常类似 Word 的文档结构视图。视窗左侧的导览窗格不一定会自动出现，也许您必须手动点击工具栏上的相关按钮。有趣的是 Word 称呼这种视窗为结构图，而 Acrobat 称此为导览窗格。导览窗格可呈现“书签”（如图 12-8）和“缩略图”（如图 12-9）两种东西。

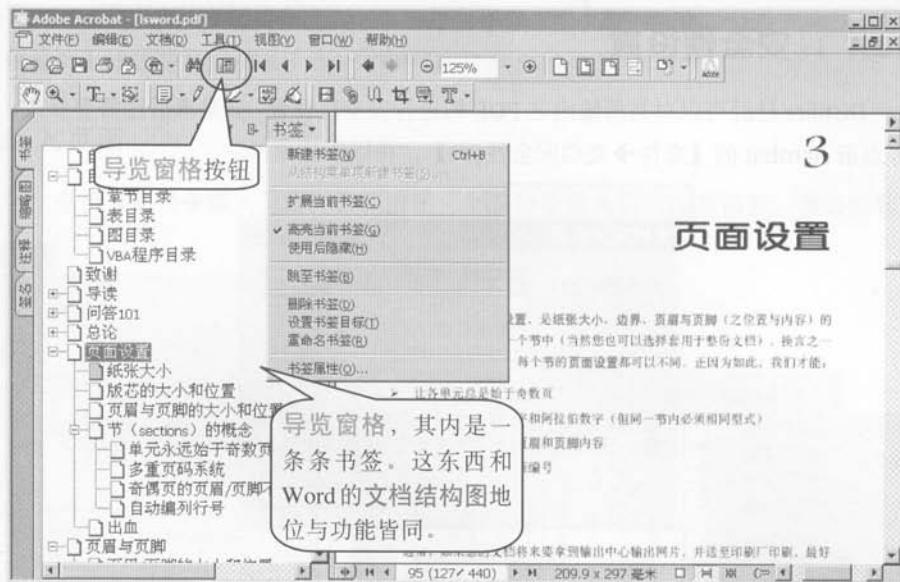


图12-8 Acrobat 主画面，左为导览窗格

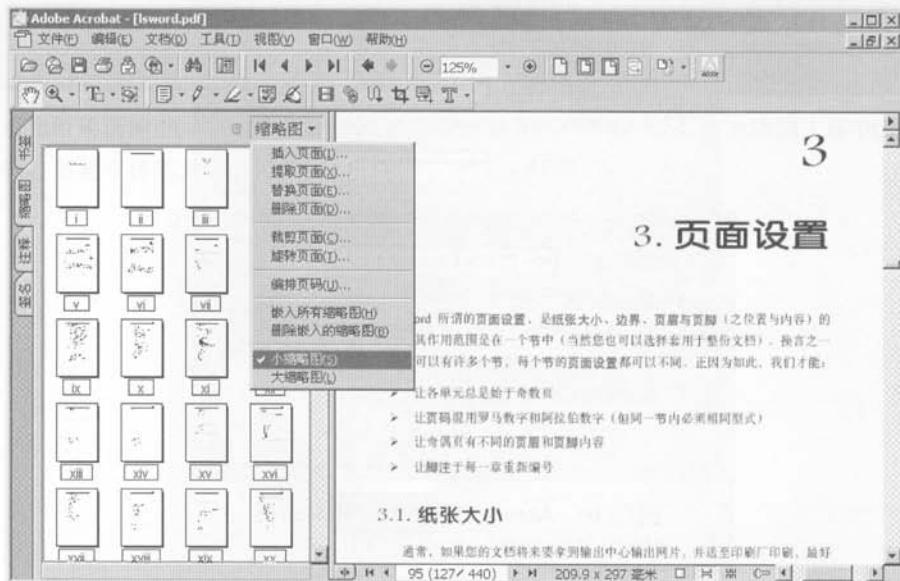
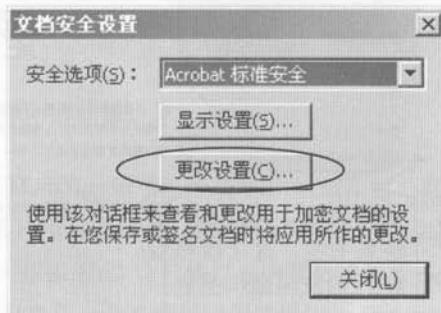


图12-9 Acrobat 主画面，左为小型缩图

12.7.1. 安全性设置

Distiller 已经可以对其所输出之 PDF 档进行安全性设置。如果事后还需要修改，可点击 Acrobat 的【文件→文档安全性(Y)】，得以下画面：



点击【更改设置(E)】，可得图 12-10 的【标准安全】对话框，在其中进行设置。

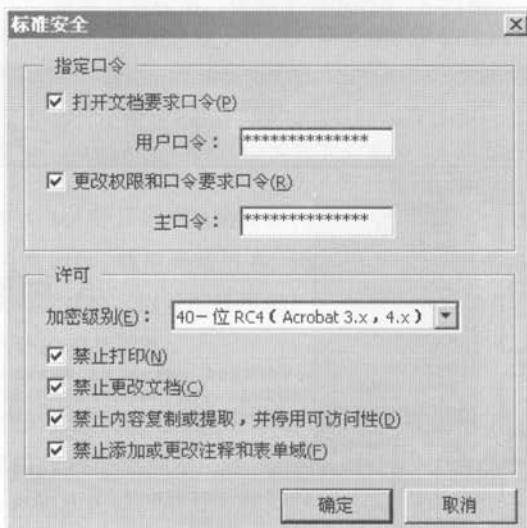


图12-10 Acrobat 的【标准安全】对话框

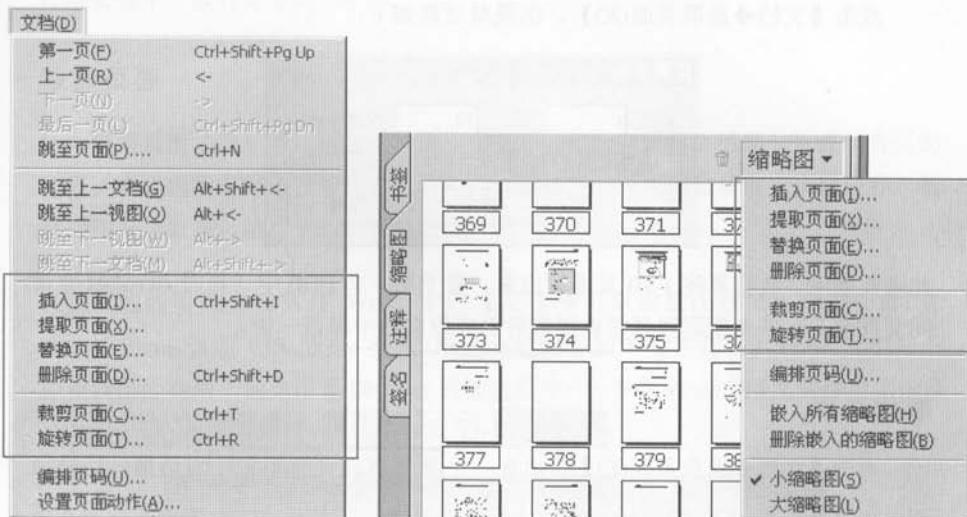
12.7.2. 页面操作

Acrobat 允许我们删除 PDF 中的某些页面，或是替换某些页面，或是将两份 PDF 合并为一份。两个地方可以做这些动作：

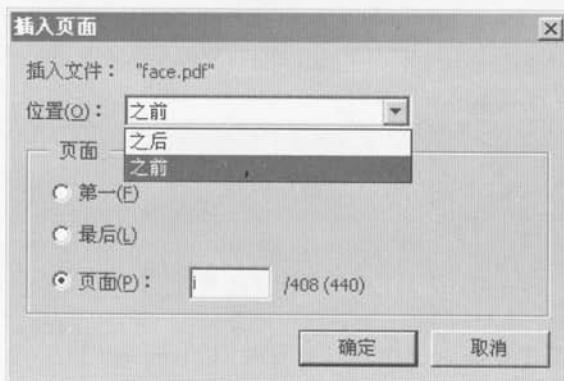
过去我的 Word 功力尚浅，一本书总得做成三个文件（文前、正文、索引），便需要上述这些页面增减手法，才能完成整本 PDF 电子书。

插入页面

点击【文档→插入页面(I)】，出现一个选择要插入的文件对话框。您必须知



道您所挑选的 PDF 档的防写密码（如果有的话），否则 Acrobat 不允许插入该 PDF 档。回复文件名之后，Acrobat 出现以下对话框：



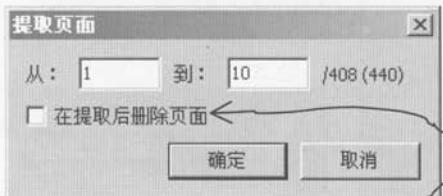
这时候便可挑选任何一个页面（包括◎第一或◎最后）"之前" 或 "之后" 作为插入点。再点击确定即可。

请注意上述画面中的号码表示法：408(440) 表示整个 PDF 档目前共 440 页，

其中流水编号 408 页，余为非流水编号（详 12.7.3 节）。**页面(P)**栏中填入的 1 表示取流水编号第 1 页。如果想要挑选非流水编号（例如 i,ii,iii... 或 p-1,p-2,p-3），就请把该非流水编号填入**页面(P)**栏。

提取页面

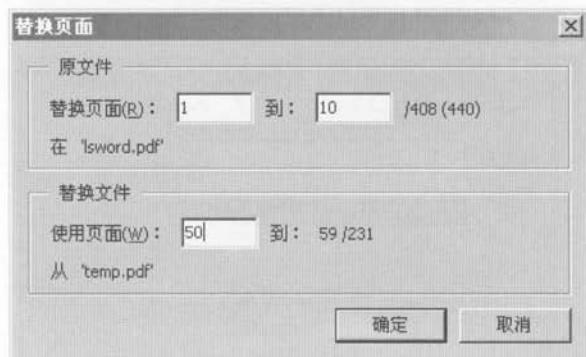
点击【文档→提取页面(X)】，出现对话框如下：



此画面表示，我打算将 1-10 页提取出来，置于另一个 PDF 中（稍后 Acrobat 会询问文件名），并希望那些页面被提取后仍在原文档中保留一份。

取代页面

点击【文档→取代页面(E)】，出现一个**选择有新页面的文件**对话框。您必须知道你所挑选的 PDF 档的防写密码（如果有的话），否则 Acrobat 不允许引入该 PDF 档。回复文件名之后，Acrobat 出现以下对话框：



此画面表示，我打算将 tass.pdf 中的 1-10 页，以 temp.pdf 第 50 页起的 10 个页面取代之。点击**确定**即可实施。

删除页面

点击【文档→删除页面(D)】，出现对话框如下：

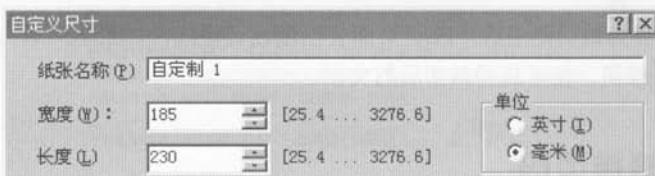


此画面表示，我打算删除 1-6 页。

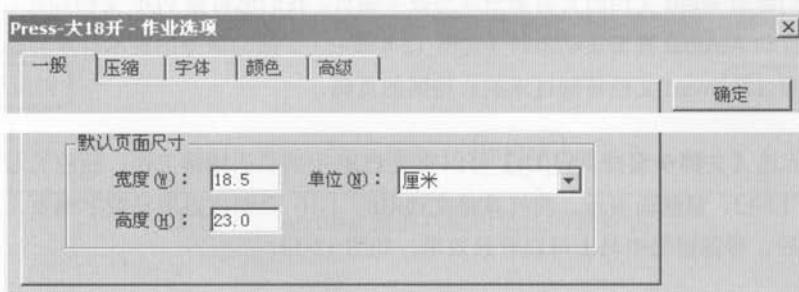
裁剪页面

裁剪页面主要是为了修改文档的页面大小。您可以调整文档某一页或所有页的边界，或个别指定每一页的边界。您可以设置参数，也可以目视设置页面边界。裁剪动作无法取消；裁剪动作并不会降低文件的体积。

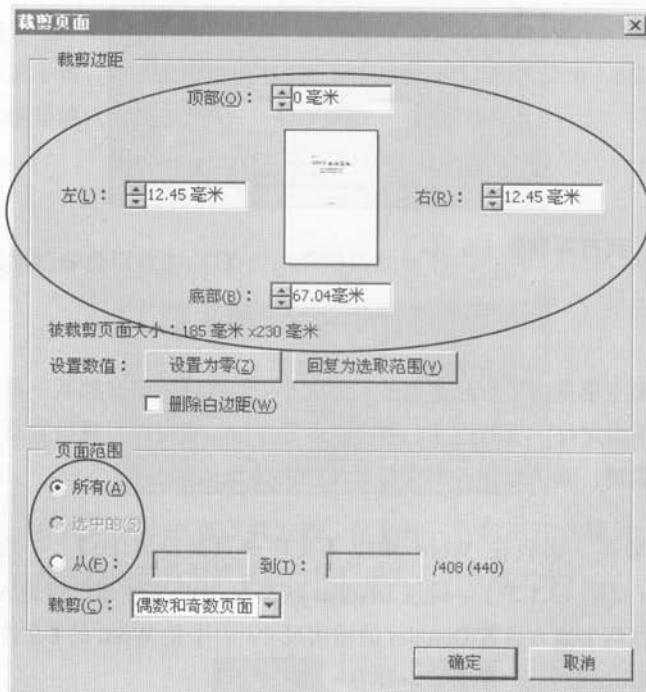
我的经验是，当我在 Word 中以【文件→页面设置(U)】 + 纸型(S)设置 $185 \times 230\text{mm}$ 页面（3.1 节），经过 Distiller 转换为 PDF 文件却仍然是 A4 大小——即使我已经在两个地方设置 Distiller 的输出尺寸：一个是 Word 的【文件→打印(P)】+Acrobat Distiller 打印机+属性(P)+纸张+自定义(T)：



另一个是 Distiller 的【设置→作业选项】+一般：



于是我只好求助于 Acrobat 的页面裁剪功能。首先将 PDF 文档载入 Acrobat（页面尺寸为 A4，因为上述两动作没有发生效果），而后点击【文档→裁剪页面】（或工具栏上的相应图标 \square ），出现对话框如下：



填妥边界的裁切范围后，PDF 页面会出现裁切虚线，如图 12-11。此时再点击上述的 **确定** 按钮，便可获得裁切后的文档。

12.7.3. 编排页码

当 Distiller 为我们制作出一个崭新的 PDF 档时，其内完全使用流水页码。但正如我们希望 Word 文档的文前采用罗马数字编页，我们也希望 PDF 文档如此。这么一来拉动 Acrobat 视窗滚动条时（见图 12-8 最右），滚动条旁的提示框才会显现出与内容（从 Word 文档转制过来的）相同的页码。

点击【**文档→编排页码(U)**】可以完成 PDF 中的页码编排工作。相应的对话框如图 12-12。该画面表示，我打算将文档中的 1-10 页改以罗马数字编页。点击 **确定** 后，导览窗格中马上可以看到效果，如图 12-13。

此后，当我们拉动 Acrobat 视窗滚动条时（见图 12-8 最右），如果出现 **iv (4/20)**，表示拉至总共 20 页中的第 4 页，页码为 iv。如果出现 **4 (14/20)** 则表示已拉至总共 20 页中的第 14 页，页码为 4。



图12-11 Acrobat 裁剪功能所造成的裁剪线

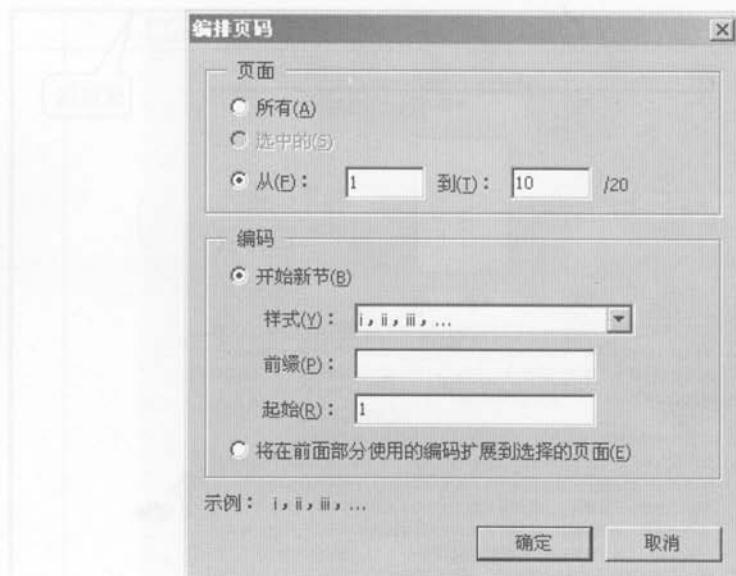


图12-12 Acrobat 的编排页码对话框

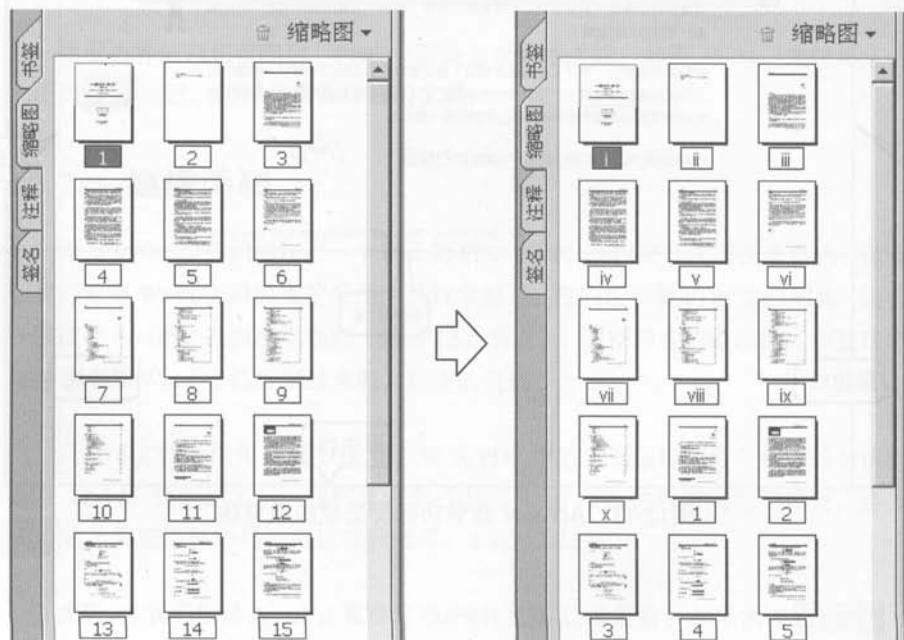


图12-13 图左全部是阿拉伯数字编页，图右前10页为罗马数字编页

12.7.4. 书签+导览窗格

PDF 导览窗格和 Word 文档结构图（图 2-1），不论就长相、功能、地位而言，完全相同。都提供给阅读者一个“文档概览”和“快速链接”。

但是 PDF 书签和 Word 书签（11.3 节）不太一样，虽然两者作用有相似之处。PDF 书签就是出现在导览窗格中的那些超链接，可由 PDF 文档制作者手动加上，也可以由 PDFMaker（Acrobat 提供的一个 Word 宏）进行自动化抽取制作。Word 书签则是一种由 Word 文档作者手动加上的隐藏记号，可作为文档编撰时交叉引用（cross reference）的对象。

PDF 书签本质上是一种电子记号，允许读者跳到 PDF 文档或外部文档的重要某处，等于是为 PDF 文档加上导览辅助。PDF 书签还可以链接至另一个画面，或启动某个动作（action），像是播放一段声音等等。

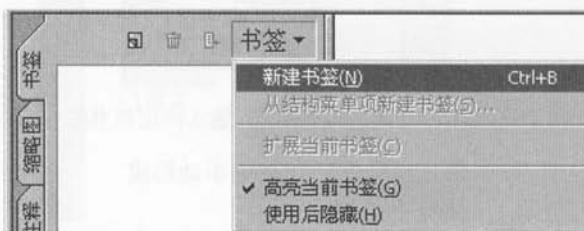
Acrobat 视窗左侧的导览窗格并不是天生就有的，很多时候您得先按工具栏上的对应按钮 。导览窗格的内容（亦即 PDF 书签）也不是先天就有的（难道天上会掉馅饼）。Distiller 刚做出来的热腾腾的 PDF 导览窗格一开始其实是一片空白（除非使用 12.8 节的 PDFMaker）。

PDF 书签的地位就像“具备超链接能力的目录”一样。在一份长文档中，导览窗格+书签对读者具备极大的吸引力。

书签的制作可分手动和自动化两种。以下分别介绍。

手工制作书签

一开始，刚出炉的 PDF 文档没有任何书签（除非使用 12.8 节的 PDFMaker）：



点击上图的新建书签(N)，导览窗格中出现一个空白文本框，让使用者键入书签名

称。此时不妨进入原始 Word 文档，将其中的目录内容一一拷贝/粘贴到这里来。不过一本书的目录项高达数百条，即使“拷贝/粘贴”也是不小的任务！下图是制作流程：

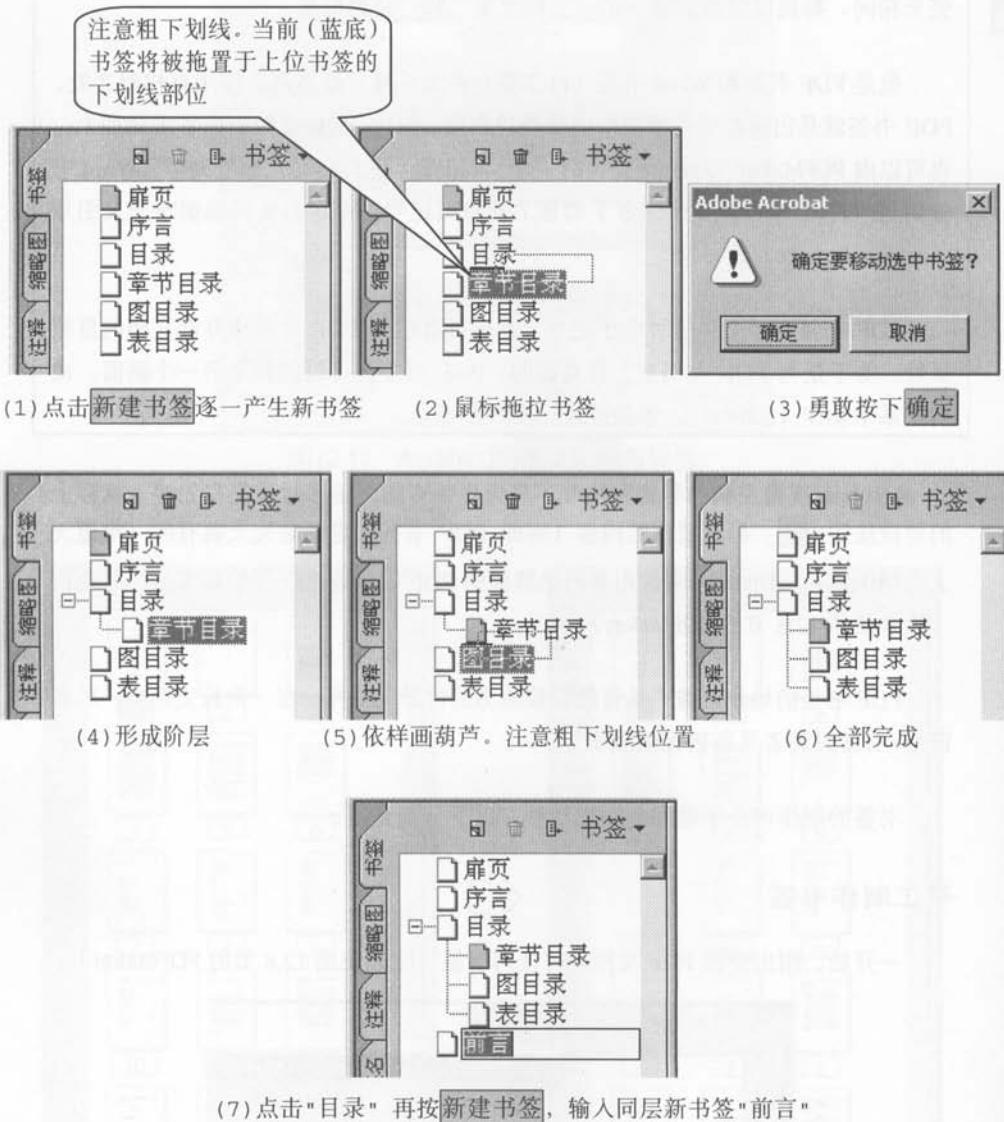
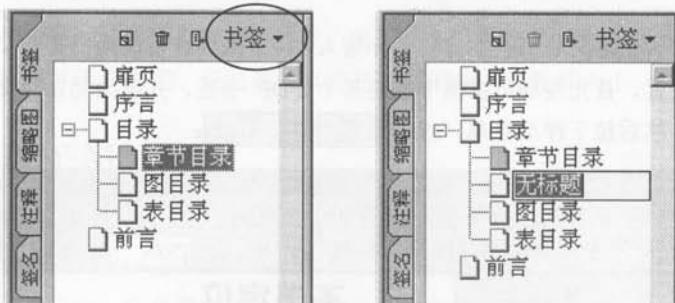


图12-14 PDF 书签之手动形成

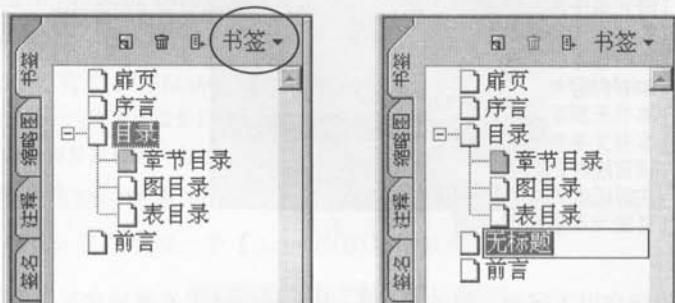
请注意，在图 12-14 的各阶段中，点击任何旧书签后，按下**新建书签**，即可于该层书签之下获得同层新书签。例如：



(1) 点击“章节目录”再按新建书签

(2) 获得“同层”新书签

图12-15 添加新书签, 例一



(1) 点击“目录”再按新建书签

(2) 获得“同层”新书签

图12-16 添加新书签, 例二

所有书签都可以任意拖放至任何位置。只要注意上述所说的粗下划线，就能够清楚知道拖放的新位置。如果书签有误，也可以点击它之后按删除书签(D)，如图 12-17。

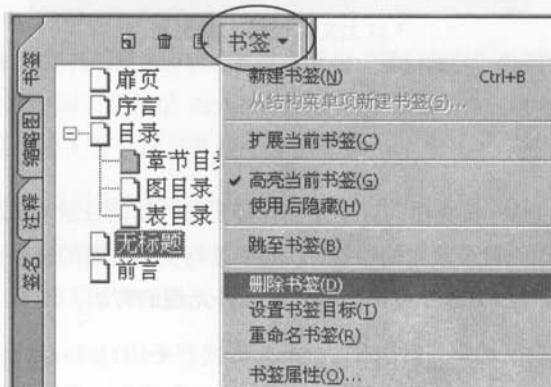
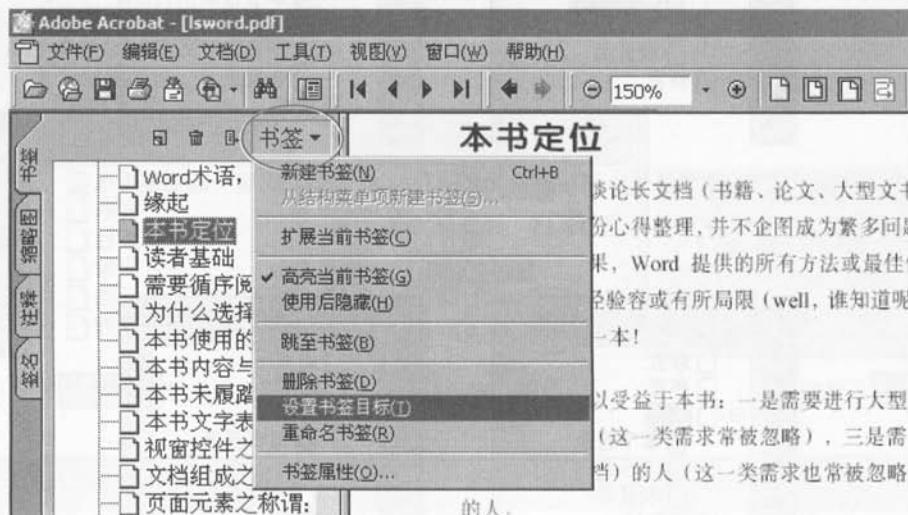
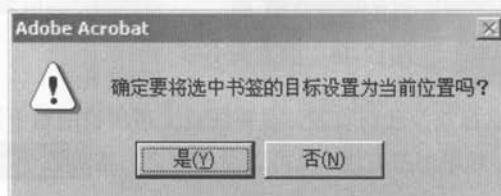


图12-17 删除某个书签

到目前为止都只是书签本身的名称输入。接下来要进行的是书签与文档页面之间的链接设置。首先在导览窗格中选定某个 PDF 书签，并以滚动条将文档卷至您要的地点，然后拉下浮动菜单，点击**设置书签目标(T)**：



Acrobat 会询问您以下问题，回答**(Y)**，从此以后该书签就链接到目前页面上：



自动化制作书签

一份完善的 PDF 长文档，目录是很有必要的。然而目录动辄数百条，若以手工逐一完成那些作为目录使用的书签，以及书签与实际页面的链接设置（如上图），实在是太辛苦了。PDFMaker 可协助我们做这件无趣的劳动。详见 12.8 节。

更动页面会影响书签吗？

书签主要用来链接（link）页面。然而由于 PDF 链接存在于文档的分离图层中，

并不和文档的图片或文字自然产生关联，因此在文档中取代或新增页面时，其他页面的旧有链接并不受影响，仍然维持其在链接层的相对位置和大小。

12.8. PDF Maker

PDFMaker 是 Acrobat 提供的一个专为 Microsoft Office 服务的工具。它的地位相当于在 PDF Distiller 或 PDFWriter 之上架设一层介面，接受用户的各种设置（详后）。实际动作时，它处理它该处理且能够处理的部分（例如 Word 文档的书签、目录、超链接、交叉引用），再将“转换为 PDF”的底层工作交给 PDF Distiller 或 PDFWriter。因此您不再需要像以前一样地在 Word、Excel、或 PowerPoint 的 **打印** 对话框中选择 Acrobat Distiller 作为打印机（当然您当然还是可以那么做，而且没有任何坏影响）。有人说 PDFMaker 只是个宏，唔，固然的确有个 PDFMaker.dot 针对 Word，也有个 PDFMaker.ppa 针对 PowerPoint，但整个 PDFMaker 恐怕不只是单纯意义上的宏而已。

只要您的电脑在安装 Acrobat 之前装有 Microsoft Office，Acrobat 安装过程便会自动为 Office 软件增加一个【Acrobat(B)】菜单如下：

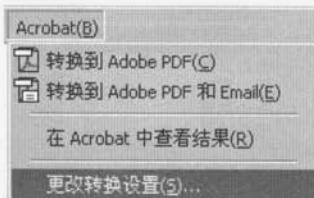


图12-18 Acrobat 在 Word 身上加装的菜单，for PDFMaker

请注意，由于 PDFMaker 与 Microsoft Office 之间有密切的合作关系，因此当您需要用到 PDFMaker 时，手上的 Acrobat 版本也因此和 Office 版本（甚至 Windows 版本）有了较强的相关性，这不算是个好消息。

- Acrobat 4.0：适用于 Windows 98 + Office 97 (98)
- Acrobat 5.0：适用于 Windows 2000 + Office 2000
- Acrobat 6.0：适用于 Windows XP + Office XP

点击图 12-18 的【Acrobat (B)→转换到 Adobe PDF(C)】，会调用底层的 PDF Distiller 或 PDFWriter 进行转换。点击【Acrobat(B)→更改转换设置(S)】，则可在对话框中做各种设置，如图 12-19，其中的 **设置**、**安全性**、**显示选项**，和 12.6 节 PDF Distiller 所提供的差不多，这里不再多做说明。

图 12-19 的 **Office** 属性页主要让我们决定在转换 PDF 的过程中是否保留 Word 文档的各种特殊构件如超链接、交叉引用、批注……。通常我们会想要保留它们，有益无害嘛。

图 12-20 的 **书签**属性页，允许指定是否将 Word 中的标题（heading）或样式（styles）转换为 PDF 书签。是的，我们已经知道，Word 可以抽取被指定的**标题 1**~**标题 9**（或其他样式）作为目录（Table of Contents⁷），并放入结构图。类似情况，PDFMaker 也可以抽取**标题 1~标题 9**（或其他样式）做成具备层次结构的 PDF 书签，放进导览窗格。这就大大解决了 12.7.4 节所说“手工转换数百个目录项目成为 PDF 书签”的困扰。

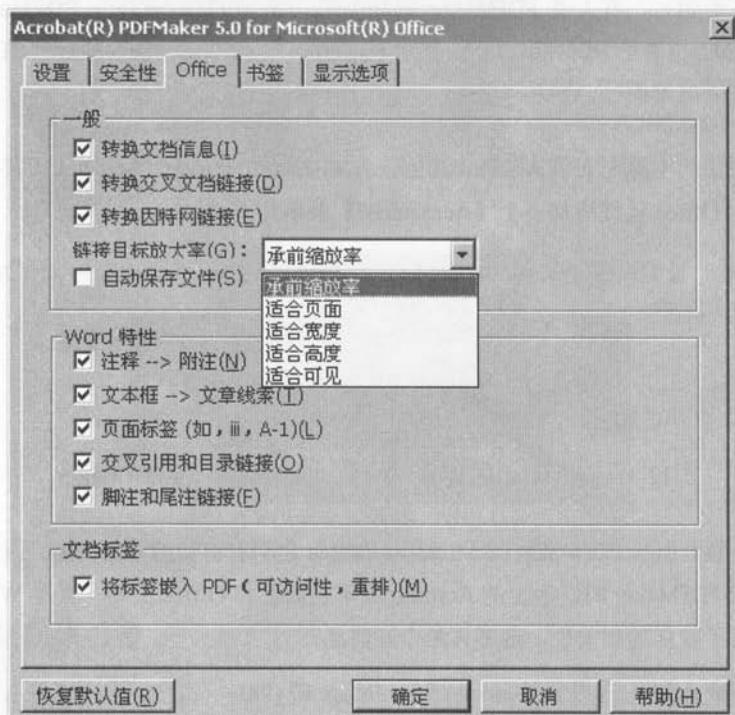


图 12-19 PDFMaker for Microsoft Office, **Office**

⁷ “Table of Contents” 指的是“章节目录”，有别于 Word 另外还提供的 “Table of Figures” “图表目录”。

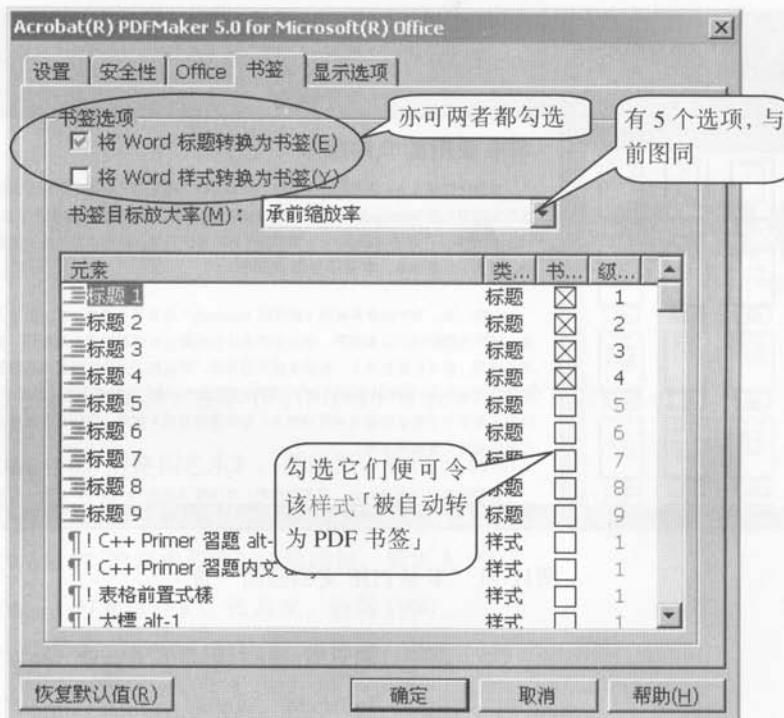


图 12-20 PDFMaker for Microsoft Office, [书签]

做完图 12-20 的所有设置之后,点击【确定】。那么,一旦您将来点击图 12-18 的【Acrobat(B)→转换到 Adobe PDF(C)】,将 Word 文档转换为 PDF, PDFMaker 会自动为您产生您所选择的书签。不幸的是,我以 PDFMaker 5.0 制作本书之 PDF 文件,却总是当机,不知是否因为文档太过复杂(用了太多交叉引用与书签……)?唔,我真的不知道。于是我只好舍弃 PDFMaker,改而直接使用 PDF Distiller。图 12-21 便是以 PDF Distiller 做出的本书 PDF 文档:

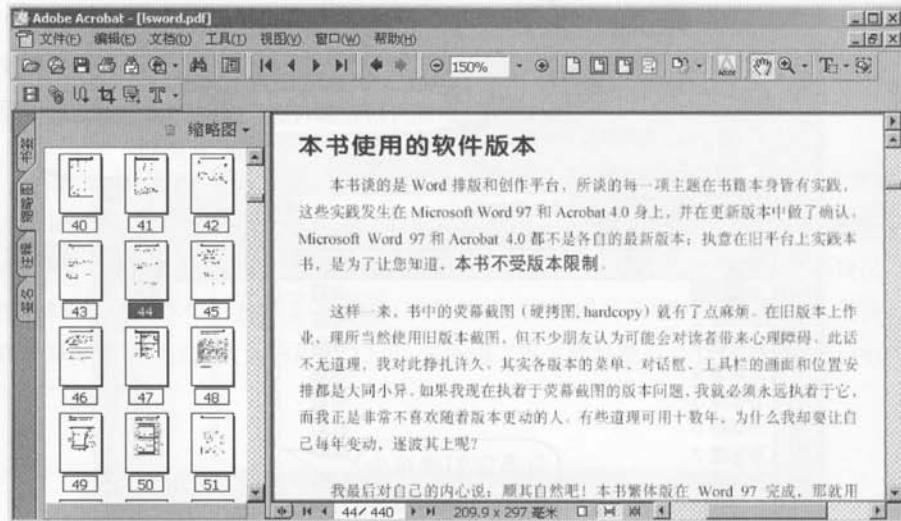


图12-21 本书PDF文档画面一瞥

A

A 参考资料

以下书籍和资料，在本书写作过程中曾经带给我有用的信息。

- 《Word 6.0 的使用艺术》，薛晓岚，资讯人 1995
- 《Word 7.0 的使用艺术》，薛晓岚，资讯人 1996
- 《Word 97 的使用艺术》，薛晓岚，资讯人 1997
- 《Word 巨集设计》，张惠斌，松岗 1996
- 《Word 2000 长文件设计》，侯俊耀，学贯 1999
- 《Word 97 On-line Help》，Microsoft Corp.
- 《Acrobat 4.0 On-line Help》，Adobe Systems Inc.
- 《Word 2000 VBA Programmer's Reference》，Duncan Mackenzie，Wrox 1999

B

B 粗粮的细作

— 邹震 —



这是一篇刊载于 2004/03/03《中华读书报》的书评文章，写得极好。蒙作者邹震先生首肯，我把文章放在这儿以飨读者。《中华读书报》是大陆知名的知性报纸，常有许多书评，包括 IT 技术类书评。邹震先生为大陆人士，可能有些用语对台湾读者不那么熟悉。语气不影响理解，所以我没有做任何用语转换。

※文中所说的币值是人民币。人民币和新台币的汇率目前大约是 1：4。

※文中所说的出版社都是大陆出版社。

※两岸用语对照：软件 ⇔ 软体。信息 ⇔ 资讯。项目管理 ⇔ 专案管理。程序 ⇔ 程式。

对生活条件日渐优越的人们来说，粗粮正在愈来愈成为一种高档的食品。粗粮由于富含人体所必需的一些微量元素，从而成为了人们不可缺少的食物，但粗粮糟糕的口感却往往又拒食客于千里之外，因此，粗粮转化为高档食品的前提是厨师能把粗粮也做出好的味道。曾经有一位上了年纪的同事，由于年轻时吃了太多的粗粮，对贴饼子熬小鱼等粗粮做成的所谓“农家菜”实在提不起兴趣，但当厨师把蒸好的窝头用油炸透，再沾以炼乳来食用时，我的这位同事却又开始了大嚼特嚼。世界上相通的事物很多，退回到计算机图书市场，今天办公软件类图书所处的地位就和粗粮的情况相仿。

粗粮式的办公软件

虽然利用 Office 处理办公事务曾经一度成为是否会使用电脑的标志，但到了今天，办公软件类图书已成为了没有细做前的“粗粮”，甚至沦为了不折不扣的“鸡肋”。

说办公软件类图书像粗粮，是因为办公软件同粗粮一样，“口感”往往很差。在人们的印象中，办公软件已得到了大面积的普及，会利用 Office 处理办公事务只

是一项很平常的本领，从而只能成为人们求职的基本条件，而不可能成为提薪或升职的条件，于是人们就开始轻视办公软件的技能。另一方面，人们却都在忽略一个事实：绝大多数人只用到了不超过 Office 套装软件 30% 的功能¹，因此许多高级事务不是 Office 做不了，而是人们没有把 Office 研究透，从而不会用 Office 来做。这样在实际工作中，一方面人们不知道 Office 能做某些事情，于是就想当然地把 Office 归入“低档”的电脑技能，另一方面，人们却面对许多实际问题一筹莫展。造成这对矛盾存在的原因，还要归咎于目前办公软件类图书的写法。

当前办公软件类图书基本上只介绍办公软件基本的操作功能，而没有和实际工作中遇到的问题结合起来²，这样，也就失去了介绍办公类软件其余 70% 功能的必要性。要使一本电脑图书成为畅销书，定价的高低往往也会成为一个决定性的因素，而深入介绍一个软件的应用时，图书的篇幅必然随之增大，定价也会随之增高，编辑这时就会被一个问题所困惑：是否存在肯于花 80 元买办公软件类图书的读者。再说明白一些，这个问题就是粗粮式的办公软件类图书有没有细做的必要。

粗粮是否需要细做

要找到“办公软件类图书是否需要细做”这个问题的答案，我们不妨来研究一下读者的购买目的。以编程类图书来说，华中科技的《深入浅出 MFC》售价达到了 80 元，而电力社的《C++ Primer 中文版》的售价甚至到了 128 元，而这两本的销量却又都达到了一个惊人的数字。类似的例子还有很多，这就说明价格并不是唯一左右读者是否购买的唯一因素，读者同样会考虑图书的实用性等因素。

我们再来看一组数字。从理论上说，信息化是国内多数企业都需要的，然而，面对总投入上百万甚至上千万的大型管理信息系统，能实施大型管理信息系统的企业和不能实施的企业就形成了一个 3:7 的比例。再来看实施得起大型管理信息系统的企业，以 ERP 软件为例，实施成功的企业和实施不成功的企业又形成了一个 3:7 的比例。这样，超过七成的国内企业不是不需要信息化，而是需要以一种更为经济，更为便捷的方式来实现（如暂不需要 BPR），这实际上就给办公软件留

¹ 侯捷眉批：我认为 80% 的用户只用了 Office 软件 20% 的功能◎

² 侯捷眉批：大多数 Office 书籍并不是由“实际使用这些软件于实务工作”的行业人士所写（或提供经验），而是由电脑书籍写手所写。行业人士从领域知识切入、从解决问题的角度切入，电脑书籍写手则只能从 menu 切入！

下了巨大的市场空间。在管理软件厂商看来，以办公自动化实现为标志的 OA 系统，是低档次的代表，这种观点使得许多传统企业也耻于谈论 OA。于是，办公软件就处于一个不上不下的地位。然而企业的实际需求却还是实际存在着，如果这种需求得不到满足，产生需求的工作就会以一种病态的方式来运作。

以决策系统而论，国外的企业哪怕规模再小，也会设一个建模师的职位，这个人的职责就是把企业决策时遇到的问题综合成一个数学模型来考虑，进而找出最佳方案，而这在国内中小型企业中则成了一件不可思议的事情。国内许多企业还是延用了“三拍”式的决策方式：事前靠拍脑门决策，如果决策失误则拍大腿后悔，当决策失误造成重大损失时，则拍拍屁股走人。一个小小的决策系统的缺失，给企业带来的损失往往是无法估量的。

再对目前很热的项目管理而言，尽管项目管理已变成了一个炙手可热的名词，但在人们印象里更多的还是认为只有建筑工程和软件开发才会用到项目管理。而所谓项目，不过就是我们只做一次的工作，这样不论是设计一架飞机，还是创作一条企业的宣传标语，只要这项工作有开始和结束，有明确的产出结果，都可以作为项目来对待。因此，从医疗保健到制造行业，从软件设计到矿产资源开发，各行各业无不涉及项目管理。离开了项目管理，会造成所有的项目同时处于资源需求的最高峰或是项目同时结束，而前者会导致企业背负沉重的负担，后者则会使得企业不得不面临裁员的困境。再如项目估算、预算、控制、沟通以至审计等其他方面，企业往往也会因安排不合理而遭遇危机。其实一句中国的老话已经形象地概括了项目管理的功用——吃不穷，花不穷，算计不到就是穷。不知企业的领导者们有没有考虑过因为没有进行项目管理这种“算计”工作，而少赚了多少钱。

值得提到的还有知识管理领域，来自 Gartner 研究机构的报告表明：一家不具备协作性质的公司，只能对仅仅 20% 的企业资源进行真正的优化与管理。而另外 80% 的有用资源则分别残留在电子邮件里、桌面上、内发的文件中、从网络下载的各种页面上，以及由企业应用程序生成的可读报告里。因此，实施与没有实施知识管理的企业在资源利用方面的差距就会非常明显，在信息已成为资产的一部分的今天，没有利用好 80% 的信息资源，就等同于没有管理好同样价值的固定资产。然而，在一个管理完善的公司，没有一个领导会允许下属对 80% 的固定资产管理不善，同时，80% 的领导却不会想到对 80% 的信息资源的管理同样重要。

事实上，企业所遇到的上述问题都可以利用办公软件加上一点管理思想来解

决。特别值得提出的是，许多企业实施 ERP、CRM 系统成功后，往往只有一年的相关数据，这样就会使得系统的许多工作由于没有历史数据的积累而无法完成，ERP、CRM 的使用效果在实施初期也会大打折扣。因此，也并不是说有财力、有计划实施大型管理信息系统的企业就不需要利用办公软件把自己的业务管理进行数字化改造。由此看来，粗粮式的办公软件类图书就实在有细做的必要了，而接下来所要解决的就应该是粗粮如何细做这个问题。

粗粮怎么细做

综合考虑计算机图书市场和 IT 应用市场，我们不难发现办公软件和办公软件类图书正在构成一个恶性的死循环。对于办公软件，人们习惯于通过图书自学。而由于在人们心目中办公软件的低档化，使得办公软件类图书只介绍了软件的基本功能。当人们在实际工作中遇到某些问题时，找不到合适的教材，又会妨碍办公软件的深入使用。因此，粗粮细做的本质就是要打破这个死循环。

实际上，这也不难办到。以决策系统为例，只要具备数据模型决策的相关知识，Excel 就是一个很好的实现软件。而项目管理的实现手段不过就是微软的 Project 软件，项目管理对人们之所以陌生，是因为以实用项目管理操作为内容的 Project 类图书没有出现。看似神秘的知识管理，虽然有 SAP、PeopleSoft 这样的大型厂商介入，其实初级实现手段不过就是 IBM 的 Lotus K-station、微软的 Sharepoint 这样的小软件。由于国内企业会计最先实现了电算化，因此许多企业的 ERP 实施往往是以财务模块为核心展开的。因此，为 ERP、CRM 系统积累资源，利用 Excel 软件同样可以办到。曾经有一段时间，电子社的《Excel 在财会管理中的应用》和《Microsoft Excel 会计信息系统开发》、北京大学出版社的《Excel2002 与财务金融》、科学出版社的《电子会计师：Excel2002 学用 DIY》、清华大学出版社的《中文 Excel2002 及其在财务管理与投资分析中的应用》都曾有过不小的影响。Excel 软件在财务系统的这种应用思路，实在值得策划编辑们仔细研究一番。

可以说，办公软件是体，而用户利用办公软件解决信息化问题是用，体用结合就会产生一片新的市场。这样看来，办公软件类图书沦为“鸡肋”的原因，就在于只见到了体，而没有从用的角度动脑筋。事实上，体是有限的，而用是无限的，粗粮的细做本身就体现了一种以无法为有法，以无限为有限的思想。“用”就在企业用户的需求中，由此再回答我们前文中提出的是否有人会出 80 元买办公软件类图书，答案就是如果这书给用户带来的价值远远高于这本书的定价，高性价比当然

会吸引算得过账来的读者掏腰包买单。要不信，您去点一盘炸窝头片，沾上炼乳食用过之后，结账时你就会发现这粗粮细做以后的价钱可是贵多了。

当然，说到底，还是在于“粗粮”与“细作”的辩证关系。通常认为，粗粮是天然的、健康的，但往往粗鄙、粗糙；而细作则是精细的、考究的，但往往高傲、矜持。其实，粗粮与细作，都是事物的两个方面，不可偏废。如果一味地追求“粗”，那就成了“粗鄙”，成了“粗鄙”。如果一味地追求“细”，那就成了“矜持”，成了“矜持”。只有将两者结合起来，才能既保持事物的本真，又能够使事物更上一层楼。

所以，对于企业而言，只有选择对客户体贴入微的服务，才能赢得客户的青睐。否则，即使再好的产品，再好的服务，也可能会因为一个微不足道的疏忽，而失去客户。所以，企业要想在激烈的市场竞争中立于不败之地，就必须做到粗中有细，细中有粗。

当然，粗中有细，细中有粗，绝不是说企业可以忽略产品的质量，而是说企业在追求产品质量的同时，也要注重服务的质量。因为，服务的质量，往往决定了企业的形象。如果企业的产品质量很好，但服务的质量很差，那么，企业就很难赢得客户的信赖。相反，如果企业的产品质量一般，但服务的质量很好，那么，企业就有可能赢得客户的信赖。所以，企业在追求产品质量的同时，也要注重服务的质量。只有这样，企业才能在激烈的市场竞争中立于不败之地。

索引

这里所呈现的索引，是以本书第 8 章所提的技术做出来的。读者可就本索引，分别从中文和英文两个渠道查询词条，我会在绝大多数的中文词条上引导读者参阅对应的英文词条。由于中文术语（可能）不够统一，英文术语又（可能）不为大众所熟知，并且由于本书的目录极为详尽，因此我认为（或许）这份索引的实用功能比不上本书目录。无论如何，这份索引最重要的使命就是，让读者了解 Word 可以协助制做出这样效果的索引；制作过程如 8.5 节所描述。

Word 的索引制作功能，没有能够为索引自动做出明白的分区符号（也就是英文索引的 A, B, C... 和中文索引的笔画 1, 笔画 2, 笔画 3...）。一下各分区符号是在切断域的链接（将光标移至索引任何一个位置然后按 **Ctrl+Shift+F9**）之后，手工完成的。这不能说不是 Word 的一个失策⑧。

A

Acrobat

PDF 导览窗格, 51, 360, 369.

Adobe

Acrobat, 51, 350, 351, 360, 361, 362, 369
PDF Distiller, 40, 50, 51, 160, 352, 353,
354, 355, 356, 357, 358, 359, 360,
362, 369, 375

PDF Maker, 369, 372, 373, 374, 375

PDF Reader, 50, 51, 350, 352

PDF Writer, 352

Application(Word 对象), 268, 277, 280, 283,
284, 300, 329.

Array(VBA 型别), 269.

Author(域), 236.

Auto Numbering/自动编号, 29, 30, 31.
行号, 102, 129, 305

页码, 29.

章节, 30, 36, 70, 108, 127, 162.

尾注, 145

脚注, 30, 143.

图表, 29, 68, 70, 82, 157.

B

Bitmap(图片文件格式), 47, 52, 180, 202.

Bookmark/书签.

of PDF, 33, 51, 350, 352, 360, 369, 374,
375.

of Word, 30, 31, 49, 173, 175, 176, 241,
247, 248, 334, 335, 336, 337.

Boolean(VBA 型别), 268, 270, 271, 277,
278, 280, 323.

Break/分隔设置, 147.

分行, 44, 83, 100, 148, 195, 230, 305,
322, 325, 326

分页, 44, 83, 100, 148, 170, 171, 349.
 分节, 44, 83, 100, 148..
 自下页起, 100, 148.
 自下个奇数页起, 100, 101, 148, 212,
 214.
 自下个偶数页起, 100, 101, 148.
 接续本页, 100, 148.
 分栏, 44, 83, 100, 147, 148, 149, 221.
 ByRef(VBA 指令), 266, 323.
 Byte(VBA 型别), 268.
 ByVal(VBA 指令), 266.

C

Caption/题注, 37, 45, 47, 55, 82, 129, 157,
 159, 160, 161, 241, 296, 334, 337.
 CDR(图片文件格式), 181.
 Characters(Word 对象), 49, 278, 281, 282,
 285, 288, 290, 326, 328.
 Chr(VBA 函数), 272, 281, 310, 311, 317,
 318, 322, 324, 325, 326, 329.
 Close(VBA 指令), 272, 277, 315, 316, 319.
 Collection(VBA 型别), 269, 276, 277, 281,
 284, 286, 289, 290, 291, 293, 294,
 296, 300, 301, 320, 321, 324, 326.
 Column/栏. 参阅 Break+分栏
 Comment/批注, 49, 50, 338, 341, 342, 344,
 345, 374.
 CreateDate(域), 236.
 Cross Reference/交叉引用, 30, 45, 84, 109,
 172, 173, 174, 240, 334, 337, 360,
 373, 374.
 CStr(VBA 函数), 272, 281, 324, 325, 326,
 328.
 Currency(VBA 型别), 268.

D

Date
 as VBA 型别, 268.
 as 域, 236, 239.
 Date&Time/日期和时间, 239, 241, 242,
 338.
 Decimal(VBA 型别), 268.
 Dim As(VBA 指令), 267, 268, 269, 280,
 281, 284, 288, 291, 293, 295, 296,
 300, 302, 306, 307, 308, 309, 310,
 311, 315, 318, 321, 322, 323, 324,

326, 329.
 Do Loop Until(VBA 指令), 271.
 Do Loop While(VBA 指令), 271.
 Do Until Loop(VBA 指令), 271.
 Do While Loop(VBA 指令), 271, 316, 318.
 Document Map/文档结构图, 71, 73.
 Word 引导框, 31, 36, 70, 71, 72, 73, 75,
 130, 165, 167, 169, 171, 210, 360,
 369, 374.
 Document(Word 对象), 33, 50, 68, 76, 277,
 282, 284, 285, 286, 350, 351.
 Documents(Word 对象), 271, 277, 282, 283,
 284, 291, 292.
 Double(VBA 型别), 268.
 DXF(图片文件格式), 181.

E

EditTime(域), 236, 239.
 Else(VBA 指令), 270, 293, 319, 324
 EMF(图片文件格式), 181.
 Endnote
 as Word 对象, 278, 285, 297.
 as 尾注, 82, 129, 143, 144, 145, 173, 175,
 250, 334.
 尾注文本. 参阅 Footnote+脚注文本
 尾注引用. 参阅 Footnote+脚注引用
 EPS(图片文件格式), 181.
 Exit Do(VBA 指令), 271
 Exit For(VBA 指令), 271, 306, 323

F

Field/域, 20, 45, 56, 84, 106, 208, 237, 239,
 240, 241, 242, 244, 247, 248, 249,
 295, 296.
 代码, 235, 236, 239
 结果, 237
 FileName(域), 236.
 FileSize(域), 236, 239.
 Find(Word 对象), 275, 278, 280, 303, 304.
 Footer/页脚, 19, 32, 40, 80, 82, 109, 132,
 191, 278.
 Footnote
 as Word 对象, 278, 297.
 as 脚注, 19, 30, 68, 82, 143, 173, 175,
 250, 334.

脚注文本, 115, 143, 145.
脚注引用, 115, 143, 145.

For Each(VBA 指令), 271, 284, 288, 291, 293, 295, 296, 300, 307, 308, 322, 323, 325, 326, 328, 329.

For Input(VBA 指令), 272, 315, 316, 318.

For To(VBA 指令), 271, 302, 306, 307, 308, 309, 311, 322, 324.

FPX(图片文件格式), 180.

Function(VBA 指令), 267, 323, 324.

G

GIF(图片文件格式), 82, 180.

H

Header/页眉, 32, 39, 80, 83, 102, 105, 109, 132, 191, 278.

Hotkey/快捷键, 74, 83, 128, 132, 134, 135, 136

Hyperlink/超链接, 49, 332, 333, 334, 335, 337, 352, 360, 369, 373, 374.
vs. 交叉引用, 334

I

ICO(图片文件格式), 180.

If Then(VBA 指令), 270, 293, 306, 311, 319, 323, 324, 326, 328, 329.

Indent/缩进, 37, 53, 54, 55, 230, 287.

Index

as 域, 219, 224, 239, 240, 247, 291, 292, 297, 321.
as 索引, 19, 45, 52, 54, 68, 72, 83, 99, 169, 217, 223, 235, 242, 363.
索引项, 69, 84, 218, 223, 226, 247

Input(VBA 指令)

For Input. 参阅 For Input
Line Input. 参阅 Line Input

InStr(VBA 函数), 310, 311, 316, 318.

Integer(VBA 型别), 266, 267, 268, 269, 302, 318, 324.

J

JPG(图档格式), 180, 201, 202.

L

Label/标签, 29, 30, 52, 82, 157, 158, 160, 173, 176, 296, 334

Len(VBA 函数), 307, 309, 311, 316.

Line Input(VBA 指令), 272, 315, 318..

ListNum(域), 239

ListTemplate(Word 对象), 301, 302, 303.

ListTemplates(Word 对象), 285, 301, 302.

Long(VBA 型别), 267, 268, 278, 311, 316.

LTrim(VBA 函数), 307, 309, 311, 316.

M

Macro/宏, 12, 31, 32, 34, 39, 45, 47, 48, 52, 132, 134, 135, 136, 142, 214, 253, 254, 255, 257, 259, 261, 262, 275, 291.

VBA 编辑器, 136, 258, 259.

新建, 257

实例 1 程序代码加行号, 264, 309, 310

实例 2 两岸术语转换, 140, 271, 311

实例 3 列出文档所有字体, 320

实例 4 找出字体发生处, 327

录制, 255.

Mid(VBA 函数), 311, 317, 318.

Module/模块, 259.

MsgBox(VBA 函数), 271, 272, 280, 281, 283, 284, 285, 286, 287, 288, 291, 293, 295, 296, 300, 302, 322, 325, 326, 328, 329.

N

Next

as VBA 指令. 参阅 For Each 或 For To
as 域, 240.

NextIf(域), 240

Normal/普通视图, 33.

NumChars(域), 236, 240, 245.

NumPages(域), 236, 240, 245.

NumWords(域), 236, 240, 246.

O

Object

as VBA 型别, 268.

as Word 插入对象, 46, 179, 181, 183, 187, 188, 190, 193, 195, 197, 200.

Object Model/对象模型, 48, 253, 279, 282

Object Oriented/面向对象, 48, 130, 179, 275

OLE/对象链接与嵌入, 187, 188, 202

Open(VBA 指令), 271, 272, 277, 315, 316, 318.

Outline/大纲视图, 71, 74, 75, 78.

P

Page Number/页码, 19, 32, 40, 80, 90, 101, 109, 110, 173, 175, 176, 235, 244, 366.

Page(域), 38, 240, 244, 245.

PageRef(域), 240, 247.

Paragraph

 as Word 对象, 49, 277, 286, 287, 307.
 as 段落, 31, 37, 42, 43, 82, 84, 108, 114, 117, 118, 124, 126, 150, 175, 195, 198, 249, 250, 286, 301, 305.

Paragraphs(Word 对象), 49, 285, 286, 288, 306, 307, 308, 309, 311, 329.

PCD(图片文件格式), 180.

PCX(图片文件格式), 180.

PDF Distiller. 参阅 Adobe

PDF Maker. 参阅 Adobe

PDF Writer. 参阅 Adobe

PDF/便携式文档格式, 12, 33, 34, 50, 51, 68, 349, 350, 351, 352, 357, 366, 369, 370, 373, 375

Picture/图片

 文绕图, 190, 191, 195, 200
 浮贴图, 46, 190, 191, 192, 193, 197
 类字符图, 46, 190, 191, 192, 193, 194, 197

PNG(图片文件格式), 180.

PringDate(域), 240

Print Layout/页面视图, 70, 71, 72, 73, 79, 223.

PrintDate(域), 236.

Project/工程, 259.

Protection/保护

 防写保护, 345, 346.
 修订保护, 340, 341, 342, 343, 344, 345.
 批注保护, 341, 342, 343, 344.
 开启保护, 343, 345.

PSD(图片文件格式), 181.

R

Range(Word 对象), 49, 277, 287, 288, 291, 292, 296, 297, 298, 307, 308, 309, 310, 311, 326, 329.

RD(域), 240

Ref(域), 240, 248, 334.

Replacement(Word 对象), 275, 278, 280, 303, 304, 319.

RevNum(域), 236, 240.

S

SaveDate(域), 236, 241.

Section

 as Word 对象, 241, 291, 292, 293, 297
 as 域, 236, 241, 245, 291, 292, 293, 297.
 as 节, 17, 29, 31, 38, 40, 41, 52, 81, 95, 99
 分节符号, 148
 自下个奇数页起, 101, 212
 自下个偶数页起, 101

SectionPages(域), 236, 241.

Sections(Word 对象), 278, 282, 285, 291

Selection(Word 对象), 49, 268, 275, 277, 278, 280, 281, 286, 288, 297, 298, 299, 303, 307, 308, 309, 310, 311.

Sentences(Word 对象), 49, 277, 278, 282, 285, 288, 289.

Sentions(Word 对象).

Seq(域), 241, 248.

Set BookmarkBySet(域), 236

Set(Word 指令), 286, 288, 291, 302, 308, 309, 311, 324

Set(域), 241.

Single(VBA 型别), 268.

String(VBA 型别), 266, 267, 268, 269, 277, 278, 280, 281, 284, 306, 307, 309, 310, 311, 315, 316, 318, 322, 323, 324, 326.

Style

 as Word 对象, 49, 278, 287, 293, 294, 296.

 as 样式, 37, 42, 51, 54, 72, 73, 74, 83, 113, 115, 116, 117, 119, 121, 122, 123, 124, 125, 127, 128, 130, 132, 134, 135, 136, 142, 160, 165, 169, 277, 293, 315.

 内建样式, 43

 正文, 19, 40, 53, 54, 55, 73, 87, 129, 230

 目录(1-9), 54, 73, 87, 129, 130, 205,

- 207
 页码, 129
 索引(1-9), 54, 69, 73, 87, 129, 219,
 224, 230
 尾注文本, 129
 尾注引用, 129
 脚注文本, 54, 55, 87, 129, 145
 脚注引用, 54, 55, 129, 145
 批注文本, 129
 批注引用, 129
 超链接, 49, 55, 129
 默认段落字体, 54, 55, 56, 129, 294
 图表目录, 129, 374
 题注, 45, 47, 55, 129
 标题(1-9), 36, 52, 53, 69, 71, 72,
 105, 107, 108, 129, 130, 166, 169,
 170, 173, 205, 211, 246, 250
 字符样式, 294
 段落样式, 294
S
 StyleRef(域), 56, 108, 241, 247, 249, 250.
 Styles(Word 对象), 42, 49, 55, 128, 277,
 278, 286, 293, 322, 323, 324, 329
 Subject(域), 236, 241.
 Subroutine(VBA 指令), 266.
 Symbol(域), 236, 241.
- T**
 TA(域), 241
 Tab/制表位, 42, 43, 54, 83, 114, 119, 125.
 Table of Contents/目录, 19, 44, 45, 52, 62,
 68, 72, 81, 206, 210, 246.
 目录项, 246
 TableOfContents(Word 对象), 294, 295.
 TableOfFigures(Word 对象), 296, 297.
 TablesOfContents(Word 对象), 286, 294,
 295.
 TablesOfFigures(Word 对象), 286, 296.
 TC(域), 208, 241, 242, 246.
 TC(目录项), 34, 208.
 Template
 as Word 对象, 49, 300.
 as 域, 236, 241.
 as 模板, 20, 37, 39, 84, 132, 134, 135,
 136, 142, 241, 259.
 Normal.dot, 39, 48, 135, 136, 137, 140,
 261, 284, 300
 内含宏, 253
 内含自定工具栏, 253
 内含快捷键, 253
 内含样式, 253
 全局模板, 140
 Templates(Word 对象), 49, 269, 282, 284,
 300.
 Then(VBA 指令), 270, 293, 306, 311, 319,
 323, 324, 326, 328, 329
 TIF(图档格式), 180.
 Time(域), 236, 241.
 Title(域), 236, 241.
 TOA(域), 240, 241
 TOC(域), 240, 242, 246, 337.
 Track Change/追踪修订, 49, 50, 339, 340,
 341, 342, 343.
 Trim(VBA 函数), 317, 318, 319.
- U**
 UFO(图片文件格式), 181.
 UserAddress(域), 236, 242.
 User-Defined(VBA 型别), 269.
 UserInitials(域), 236, 242.
 UserName(域), 236, 242.
- V**
 Variable, 267
 Variants(VBA 型别), 268.
VBA 函数
 MsgBox. 参阅 MsgBox
 字符串函数
 Chr. 参阅 Chr
 CStr. 参阅 CStr
 InStr. 参阅 InStr
 Len. 参阅 Len
 LTrim. 参阅 LTrim
 Mid. 参阅 Mid
 Trim. 参阅 Trim
VBA 型别
 Array. 参阅 Array
 Boolean. 参阅 Boolean
 Byte. 参阅 Byte
 Collection. 参阅 Collection
 Currency. 参阅 Currency
 Date. 参阅 Date
 Decimal. 参阅 Decimal

- Double. 参阅 Double
 Integer. 参阅 Integer
 Long. 参阅 Long
 Object. 参阅 Object
 Single. 参阅 Single
 String. 参阅 String, VBA 字符串处理函数
 User-Defined. 参阅 User-Defined
 Variants. 参阅 Variants
- VBA 指令**
- ByRef. 参阅 ByRef, Function
 ByVal. 参阅 ByVal, Function
 Close. 参阅 Close
 Dim As. 参阅 Dim
 Do Loop Until. 参阅 Do Loop Until
 Do Loop While. 参阅 Do Loop While
 Do Until Loop. 参阅 Do Until Loop
 Do While Loop. 参阅 Do While Loop
 For Each. 参阅 For Each
 For Input. 参阅 For Input
 For To. 参阅 For To
 Function. 参阅 Function
 If Then. 参阅 If Then
 Input. 参阅 For Input or Line Input
 Line Input. 参阅 Line Input
 Next. 参阅 Next
 Open. 参阅 Open
 Subroutine. 参阅 Subroutine
 With. 参阅 With
- View(视图)**
- Document Map. 参阅 Document Map,
 结构图
 Normal. 参阅 Normal
 Outline. 参阅 Outline
 Print Layout. 参阅 Print Layout
 Web Layout. 参阅 Web Layout
- W**
- wdSectionBreakContinous/接续本页. 参阅
 Break
 wdSectionBreakEvenPage/自下个偶数页起.
 参阅 Break
 wdSectionBreakNext/自下页起. 参阅
 Break
 wdSectionBreakOddPage/自下个奇数页起.
 参阅 Break
 Web Layout/Web 版式, 71, 76, 77.
 With(VBA 指令), 272, 275, 280, 286, 287,
 288, 291, 303, 328.
- WMF(图片文件格式), 181.
 Words(Word 对象), 49, 277, 278, 282, 288,
 289.
- Word 对象模型. 参阅 Object Model**
- Application. 参阅 Application
 Characters. 参阅 Characters
 Document. 参阅 Document
 Documents. 参阅 Documents
 Find. 参阅 Find
 ListTemplate. 参阅 ListTemplate
 ListTemplates. 参阅 ListTemplates
 Paragraph. 参阅 Paragraph
 Paragraphs. 参阅 Paragraphs
 Range. 参阅 Range
 Replacement. 参阅 Replacement
 Sections. 参阅 Sections
 Selection. 参阅 Selection
 Sentences. 参阅 Sentences
 Style. 参阅 Style
 TableOfContents. 参阅 TableOfContents
 TableOfFigures. 参阅 TableOfFigures
 TablesOfContents. 参阅
 TablesOfContents
 TablesOfFigures. 参阅 TablesOfFigures
 Template. 参阅 Template
 Templates. 参阅 Templates
 Words. 参阅 Words
- X**
- XE(域), 219, 220, 223, 226, 227, 242, 247
- 笔画 3**
- 大纲视图. 参阅 Outline, View
 工程. 参阅 Project
- 笔画 4**
- 内建样式/Built-in-Styles. 参阅 Style+内建
 样式
 分隔设置. 参阅 Break
 分行. 参阅 Break+分行
 分页. 参阅 Break+分页
 分节. 参阅 Break+分节
 自下页起. 参阅 Break
 自下个奇数页起. 参阅 Break
 自下个偶数页起. 参阅 Break
 分栏. 参阅 Break+分栏
 书签. 参阅 Bookmark
 引导框
 PDF 导览窗格. 参阅 Adobe Acrobat

Word 结构图. 参阅 Document Map
文档结构图. 参阅 Document Map, View
水印, 358, 359

笔画 5

出版流程, 24, 176
目录. 参阅 Table of Contents, TOC
目录项. 参阅 TC
对象. 参阅 Object
对象模型. 参阅 Object Model
电子文档
 保护. 参阅 Protection
 书签. 参阅 Bookmark
 追踪修订. 参阅 Track Change
 批注. 参阅 Comment
 超链接. 参阅 Hyperlink
写作平台
 交叉引用. 参阅 Cross Reference
 自动编号. 参阅 Auto Numbering
 题注. 参阅 Caption

笔画 6

交叉引用. 参阅 Cross Reference
印刷出版术语
 出血, 92, 103
 开本, 26, 27
 落版, 41
 制版, 23, 60
 网片, 34
多层次样式, 108, 162
页脚. 参阅 Footer
页眉. 参阅 Header
页首. 参阅 Header
页码自动编号. 参阅 Auto Numbering+页码
 页面视图. 参阅 Print Layout
 自动化章节编号, 108, 162
 自动编号. 参阅 Auto Numbering
 页码. 参阅 Auto Numbering+页码
 章节. 参阅 Auto Numbering+章节
 脚注. 参阅 Auto Numbering+脚注
 图表. 参阅 Auto Numbering+图表
导引框
 PDF 导览窗格. 参阅 Adobe Acrobat
 Word 结构图. 参阅 Document Map

笔画 7

快捷键. 参阅 Hotkey
尾注. 参阅 Endnote
批注. 参阅 Comment
宏. 参阅 Macro

笔画 8

制表位. 参阅 Tab
版面元素
 天, 32, 40, 80, 96, 191
 地, 32, 40, 80, 96, 191
注释/程序代码说明, 265, 309
视图(View)
 大纲视图. 参阅 Outline
 文档结构图. 参阅 Document Map
 普通视图. 参阅 Normal
 Web 版式. 参阅 Web Layout
 页面视图. 参阅 Print Layout
图表自动编号. 参阅 Auto Numbering+图表

图片文件格式

bitmap. 参阅 Bitmap
bmp. 参阅 Bitmap
cdr. 参阅 CDR
dxf. 参阅 DXF
emf. 参阅 EMF
eps. 参阅 EPS
fpx. 参阅 FPX
gif. 参阅 GIF
ico. 参阅 ICO
jpeg. 参阅 JPG
jpg. 参阅 JPG
pcd. 参阅 PCD
pcx. 参阅 PCX
png. 参阅 PNG
psd. 参阅 PSD
tif. 参阅 TIF
tiff. 参阅 TIF
ufo. 参阅 UFO
wmf. 参阅 WMF

笔画 9

面向对象. 参阅 Object Oriented
保护. 参阅 Protection
 防写保护. 参阅 Protection+防写保护
 修订保护. 参阅 Protection+修订保护
 批注保护. 参阅 Protection+批注保护

开启保护. 参阅 Protection+开启保护
段落. 参阅 Paragraph
追踪修订. 参阅 Track Change
标签. 参阅 Label
栏. 参阅 Break+分栏
便携式文档格式. 参阅 PDF

笔画 10

索引. 参阅 Index
索引项目. 参阅 XE
样式. 参阅 Style
样式库. 参阅 Template

笔画 11

章节自动编号. 参阅 Auto Numbering+章
节

脚注. 参阅 Footnote
文本. 参阅 Footnote+脚注文本
引用. 参阅 Footnote+脚注引用
脚注自动编号. 参阅 Auto Numbering+脚
注
域. 参阅 Field
Author. 参阅 Author
CreateDate. 参阅 CreateDate
Date. 参阅 Date
EditTime. 参阅 EditTime
FileName. 参阅 FileName
FileSize. 参阅 FileSize
Index. 参阅 Index, XE
ListNum. 参阅 ListNum
Next. 参阅 Next
NextIf. 参阅 NextIf
NumChars. 参阅 NumChars
NumPages. 参阅 NumPages
NumWords. 参阅 NumWords
Page. 参阅 Page
PageRef. 参阅 PageRef
PrintDate. 参阅 PrintDate
RD. 参阅 RD
Ref. 参阅 Ref
RevNum. 参阅 RevNum
SaveDate. 参阅 SaveDate
Section. 参阅 Section
SectionPages. 参阅 SectionPages

Seq. 参阅 Seq
Set. 参阅 Set
StyleRef. 参阅 StyleRef
Subject. 参阅 Subject
Symbol. 参阅 Symbol
TA. 参阅 TA
TC. 参阅 TC, TOC
Template. 参阅 Template
Time. 参阅 Time
Title. 参阅 Title
TOC. 参阅 TOC, TC
UserAddress. 参阅 UserAddress
UserInitials. 参阅 UserInitials
UserName. 参阅 UserName
XE. 参阅 XE, Index

笔画 12

插入
分隔设置. 参阅 Break
日期和时间. 参阅 Date&Time
域. 参阅 Field
目录. 参阅 Table of Contents
交叉引用. 参阅 Cross Reference
页码. 参阅 Page Number
书签. 参阅 Bookmark
索引. 参阅 Index
脚注. 参阅 Footnote
批注. 参阅 Comment
超链接. 参阅 Hyperlink
图片. 参阅 Picture
题注. 参阅 Caption
超链接. 参阅 Hyperlink
普通视图. 参阅 Normal, View

笔画 14

管理器, 142, 261
模块. 参阅 Module
模板. 参阅 Template

笔画 15

题注. 参阅 Caption

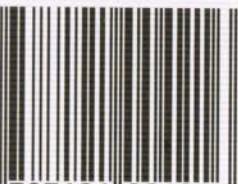
Word 排版艺术

以 Word 编写及排版大型文件

本书告诉您 Word 的能与不能

- 本书让创作者悠游于Word创作平台的各项自动化功能之上，包括多级符号自动化、页码自动化、目录自动化、索引自动化、图表题注（captions）自动化、各种交叉引用（cross references）的更新自动化、脚注编号自动化、文档结构图……
- 本书让您的协同创作（例如师生共同创作论文 or 公司编写大型技术文档）过程平顺，包括修订、批注、保护……
- 本书让排版者悠游于Word的各项排版功能之上，包括页面设置、页码、页眉、页脚、标题、正文、脚注、表格、图片插入、目录、索引……也包括复用性和方便性，如模板（templates）、快捷键（hotkeys）的设置与管理。
- 本书让您成为Word最高级用户：以宏/VBA程序设计对Word进行二次开发，并附数个实用例子。
- 本书谈如何将Word的工作成果轻松转换为PDF格式，并保留所有目录、书签、交叉引用……

ISBN 7-121-00421-6



9 787121 004216 >



责任编辑：周筠 郑兆昭
技术协作：陈硕 方舟
责任校对：张兴田
版式设计：侯捷
封面设计：暮峰图书公司

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。

ISBN 7-121-00421-6 定价：48.00元