

页面描述语言与格式

孔玲君 姚海根 编著

上海科学技术出版社

内 容 提 要

本书围绕目前在印刷和电子出版领域获得广泛应用的两种页面描述技术 PostScript 语言和 PDF 文档格式展开讨论。全书分为上、下两篇，其中上篇介绍 PostScript 语言，共八章；下篇讨论 PDF 文档格式，分成六章。第一章是对电子页面和电子页面描述语言的简要介绍，第二章讨论 PostScript 语言的基本特点，第三章介绍页面图形的基本描述方法，第四章讨论复杂图形设计与色彩控制，第五章内容涉及文本处理与描述，第六章介绍图像输出与模板设计，第七章讨论图形输出控制与彩色还原技术，第八章是对 PostScript 操作符的总结，第九章介绍 PDF 技术的起源和发展历史，第十章讨论组成页面的基本对象，第十一章简要叙述 PDF 文件结构，第十二章全面展开对文档结构的讨论，第十三章介绍 PDF 文档的公共数据结构，第十四章简要介绍 PDF 页面描述符。

本书较为全面地介绍了页面描述语言 PostScript 和页面描述技术 PDF，两者分别是印前和电子出版生产事实上的工业标准，其中，PostScript 是 PDF 的基础，PDF 则提供更多的功能，例如跨媒体出版和电子书应用等。本书在内容选择和表达方式上尽量考虑实际应用的需要，可供高等院校印刷、电子出版和包装类专业学生使用，也适用于印前公司、电子出版公司、广告公司和商业印刷公司的专业人士阅读。

前 言

现代印刷工业以计算机为主要生产工具开始于 20 世纪 80 年代中期，由 Adobe 公司于同期推出的 PostScript 语言是现代印刷作业方式的标志，引发了印前领域的一场革命，使印前生产告别了“铅”与“火”，走向“光”与“电”，PostScript 语言逐步成为印前领域事实上的工业标准。随着印前技术的发展和逐步成熟，计算机及其外围设备的处理速度和性能不断提高，印前不再满足于为后端印刷工艺准备“素材”，而是积极地向其他领域扩展，电子出版逐步成熟起来。Adobe 公司看到了这一市场需求，并考虑到 PostScript 语言的某些弱点，于 20 世纪 90 年代初期开始推出可移植文档格式 PDF，这是一种兼顾印刷出版和电子出版的页面描述技术，获得了市场的普遍认同。

PostScript 与 PDF 同为页面描述技术，但两者间却存在不少差别，这些差别主要体现在页面描述的技术本质上，并由此而带来使用功能上的差异。众所周知，PostScript 兼有编程语言和控制页面输出的双重功能，而 PDF 却不是编程语言，不存在变量和过程等控制结构，因而尽管 PDF 使用了 PostScript 语言的成像模型，但由于不允许通用编程结构而使成像操作简单得多，因为 PDF 是与描述页面对象有关的文档格式约定。任何一个 PDF 文档均不能通过 PostScript 解释器翻译，但 PDF 文档对页面的描述却可以转换为 PostScript 语言程序，这意味着只要 PDF 文档描述上没有错误，就可以在解释器转换功能的支持下形成 PostScript 数据流，控制照排机、数字印刷机和桌面打印机一类输出设备在目标介质上成像和记录。

为了让印刷和电子出版专业人士掌握页面描述的基本知识，了解 PDF 与 PostScript 两种技术的共同点和差异，在印前和电子出版素材准备时做到更有针对性，提高图文处理和输出作业的效率，编写一本同时而不是独立介绍 PostScript 和 PDF 的读物是有必要的。但是，本书在编写时又考虑到 PostScript 和 PDF 毕竟是有差异的，因而采用了保持两者相对独立的方法，分成上、下两篇讨论，必要时注明两者的差别和联系。读者在使用本书时可以从上、下两篇的首章了解到不少背景知识，例如这两种技术的用途、特点和差别，这些知识对更好地理解 PostScript 和 PDF 大有益处。

对于从事电子出版、网络出版和电子书内容准备的专业人士，需充分重视本书的 PDF 部分，因为这些领域要求实现的功能大多包含在 PDF 中，例如注解、书签、动作等与电子出版物阅读功能有关的内容。相对而言，本书的 PostScript 部分更适合于从事印前作业的专业人士阅读，但 PDF 对他们同样重要，尤其是那些希望了解数字工作流程的读者，因为 PDF 文档格式是形成数字工作流的基础，任何 PostScript 或 EPS 文件一旦转换成了 PDF 文件，则页面相关性不再存在，且便于在文档输出前作预飞检查，能最大程度地减少输出错误。

本书上篇由孔玲君编写，下篇由姚海根编写，在写作过程中得到了杜飞龙、滕跃民、程杰铭、徐东、樊丽萍、顾萍和周樊华等老师的大力支持，他们提出的宝贵意见对本书的最后成稿起了很大的作用；谭美贤、滕莉、滕少武和徐建新等则在本书的校对、修改等方面提供了很大的帮助，如果没有他们的帮助，要完成本书将是十分困难的，作者愿借本书出版的机会向他们表示衷心感谢。

作 者

2004 年 1 月于上海

目 录

上 篇 PostScript 语言及其程序设计

第一章 概 论.....	2
1.1 电子页面与页面描述语言.....	2
1.2 PostScript 语言简介.....	3
1.2.1 PostScript 语言的基本特征.....	3
1.2.2 PostScript 语言的特点与功能.....	5
1.3 PostScript 解释器的作用与工作原理.....	9
1.3.1 PostScript 解释器的作用.....	9
1.3.2 PostScript 解释器对 PostScript 程序的处理方式.....	9
1.3.3 PostScript 解释器与应用程序的交互模式.....	10
1.4 关于 PostScript 文件.....	11
1.4.1 PostScript 文件的特点.....	11
1.4.2 PostScript 文件的创建.....	12
1.4.3 PostScript 文件的使用.....	12
1.5 文本结构约定与 PostScript 文件的构成.....	14
1.5.1 文本结构约定.....	14
1.5.2 PostScript 文件的构成特点.....	15
第二章 PostScript 语言的基本特点.....	17
2.1 ASCII 编码与注释表示.....	17
2.2 数据类型与对象.....	19
2.2.1 概述.....	19
2.2.2 对象的属性.....	20
2.2.3 PostScript 对象详述.....	21
2.3 PostScript 程序的执行环境.....	26
2.3.1 堆栈.....	26
2.3.2 虚存.....	28
2.3.3 标准输入输出文件.....	31
2.3.4 PostScript 程序的初始执行环境特点.....	32
2.4 PostScript 对象的执行方式.....	33
2.4.1 立即执行.....	33
2.4.2 推迟执行.....	34
2.4.3 各类对象的执行特点总结.....	36
2.5 早期名字联编的概念与具体实现.....	36
2.5.1 即演名字.....	37

2.5.2	bind 操作符	37
2.6	常用操作符	39
2.6.1	堆栈操作符	39
2.6.2	算术和数学操作符	40
2.6.3	数组、词典和串操作符	41
2.6.4	输出到标准输出文件的操作符	43
2.7	PostScript 资源、函数和过滤器	43
2.7.1	资源的概念与种类	43
2.7.2	函数的特点	45
2.7.3	过滤器的概念与作用	46
第三章	页面图形描述基本方法	50
3.1	PostScript 语言的成像模型与图形状态	50
3.1.1	PostScript 语言的成像模型	50
3.1.2	图形状态	52
3.2	坐标系及其转换	56
3.2.1	坐标空间的种类与特点	57
3.2.2	坐标空间变换的矩阵表示法	57
3.2.3	使用矩阵操作符修改用户空间	58
3.2.4	用户空间特点的保存与恢复	61
3.2.5	程序举例	61
3.3	路径构造	63
3.3.1	路径的特点与创建方法	63
3.3.2	当前点	64
3.3.3	常见的路径构造操作符	64
3.3.4	程序举例	68
3.4	着色处理与图形生成	69
3.4.1	描绘路径产生轮廓图形	69
3.4.2	填充路径	74
3.4.3	快速的图形生成方法	76
3.5	裁剪路径的使用	77
3.6	程序举例	79
第四章	复杂图形设计与色彩控制	81
4.1	使用条件循环与递归操作符	81
4.1.1	条件执行的控制方法	81
4.1.2	循环控制	82
4.1.3	程序举例	86
4.2	颜色空间与颜色表示	86
4.2.1	基本颜色空间	86
4.2.2	PostScript 语言支持的颜色空间	89

4.2.3	颜色表示与颜色值	89
4.3	颜色空间描述与色彩控制	90
4.3.1	设备颜色空间	90
4.3.2	CIEBased 颜色空间	92
4.3.3	特殊颜色空间的特点与彩色定义	104
第五章	文本的处理与描述	115
5.1	关于 PostScript 字符与字库	115
5.1.1	PostScript 字符与字库的特点	115
5.1.2	字库词典的构成要素	116
5.1.3	字符编码	119
5.1.4	字符尺度信息与写模式设定	120
5.2	按图形方式处理文本的过程	121
5.2.1	设置当前字库	122
5.2.2	显示字符	123
5.2.3	程序举例	123
5.3	创建字符的特殊效果	124
5.3.1	彩色字符与沿任意方向排列的字符	124
5.3.2	字符轮廓的应用	125
5.3.3	使用 stringwidth 操作符制作字符下划线	126
5.3.4	修改字符间距	126
5.3.5	自定义字符形状	128
5.4	综合实例	130
第六章	图像输出与模板设计	132
6.1	图像的特点与特征参数	132
6.2	PostScript 语言对图像的处理	133
6.2.1	PostScript 语言所具有的图像描述功能	133
6.2.2	图像操作符对图像信息的处理方式	134
6.2.3	图像坐标空间与图像变换矩阵	134
6.3	图像输出具体方法	136
6.3.1	图像词典构成	136
6.3.2	画图像的具体方法	137
6.4	蒙版的使用	144
6.4.1	模板蒙版与 imagemask 操作符	144
6.4.2	直接二值图像蒙版与 Type 3 图像词典	145
6.4.3	颜色键蒙版与 Type 4 图像词典	147
6.5	使用与设计模板	148
6.5.1	模板的作用与特点	148
6.5.2	模板的创建过程	149
第七章	图形输出控制与彩色还原技术	152

7.1	概 述	152
7.2	颜色空间转换	152
7.2.1	CIE-Based 颜色空间到设备颜色空间的转换	152
7.2.2	设备颜色空间的相互转换	154
7.3	gamma 校正	157
7.4	半色调技术	158
7.4.1	半色调技术原理	158
7.4.2	半色调网屏的设置	163
7.4.3	半色调加网的基本规则与算法原理	166
7.4.4	半色调网屏设置实例	171
7.5	扫描转换	173
第八章	PostScript 操作符汇总	175
8.1	关于操作符使用表示法	175
8.2	操作符总结	176

下 篇 PDF 文件格式

第九章	PDF 的起源与发展历史	192
9.1	简 介	192
9.1.1	PDF 的主要用途	192
9.1.2	PDF 的使用	193
9.2	PDF 的发展历史和版本简介	194
9.2.1	PDF 第二版 1.1	194
9.2.2	第三版 1.2	195
9.2.3	PDF 的 1.3 版本	196
9.2.4	语法标记	199
9.3	PDF 文件的产生和使用	199
9.3.1	PDF 文件的产生	200
9.3.2	PDF 文件的使用	201
9.4	PDF 文件的通用特点	202
9.4.1	成像模型	202
9.4.2	可移植性	203
9.4.3	数据压缩	204
9.4.4	字体独立性	205
9.4.5	单向通过文件	206
9.4.6	页面随机访问	207
9.4.7	文档递增更新和可扩展性	208
9.5	PDF 文档表示、输出与内容	208
9.5.1	PDF 和 PostScript 的主要区别	209

9.5.2	PDF 文档在 PostScript 打印机上输出.....	210
9.5.3	PDF 文档部件.....	211
第十章	对 象.....	212
10.1	布尔、数字与数组对象.....	212
10.1.1	布尔对象.....	212
10.1.2	数字对象.....	212
10.1.3	数组对象.....	212
10.2	串对象与名字对象.....	212
10.2.1	串对象.....	212
10.2.2	名字对象.....	214
10.3	词典对象.....	215
10.4	流对象.....	215
10.4.1	流对象结构.....	216
10.4.2	流对象属性.....	216
10.4.3	ASCIHexDecode 过滤器.....	217
10.4.4	ASCII85Decode 过滤器.....	217
10.4.5	LZWDecode 过滤器.....	218
10.4.6	FlateDecode 过滤器.....	220
10.4.7	LZW 和 Flate 的 Predictor 功能及比较.....	220
10.4.8	RunLengthDecode 过滤器.....	222
10.4.9	CCITTFaxDecode 过滤器.....	222
10.4.10	DCTDecode 过滤器.....	223
10.5	间接对象.....	224
第十一章	文件结构.....	226
11.1	PDF 文件结构概要.....	226
11.1.1	PDF 文件.....	226
11.1.2	文件头与文件体.....	227
11.1.3	交叉参考表.....	227
11.1.4	文件尾.....	229
11.2	递增更新和文档加密.....	231
11.2.1	PDF 文档的递增更新.....	231
11.2.2	PDF 文档的加密.....	232
第十二章	文档结构.....	234
12.1	概 述.....	234
12.2	目 录.....	235
12.2.1	目录在 PDF 文档中的地位.....	235
12.2.2	目录属性.....	236
12.2.3	文档观看优选项.....	237
12.2.4	页面标签词典.....	238

12.3	页面树与页面对象	239
12.3.1	页面树属性	239
12.3.2	页面树属性继承	240
12.3.3	页面对象	241
12.3.4	页面属性	242
12.3.5	演讲稿模式	243
12.4	注 解	245
12.4.1	概述	245
12.4.2	注解属性	245
12.4.3	注解边框和边框风格	247
12.4.4	注解加亮	248
12.4.5	注解外观	249
12.4.6	文本注解	250
12.4.7	链接注解	251
12.4.8	电影放映机注解	251
12.4.9	声音注解	254
12.4.10	特殊形式注解	254
12.5	PDF 补漏白	256
12.5.1	PDF 补漏白	256
12.5.2	TrapNetwork 注解	257
12.5.3	TrapNetwork 注解属性	258
12.6	大纲树	258
12.6.1	大纲与大纲属性	259
12.6.2	大纲项目属性	260
12.6.3	目标	261
12.7	动 作	262
12.7.1	动作类型和通用动作属性	262
12.7.2	动作激发点	263
12.7.3	转向动作	264
12.7.4	转为遥控动作	265
12.7.5	启动动作	265
12.7.6	文章线索动作	266
12.7.7	URI 动作	267
12.7.8	声音动作	268
12.7.9	电影放映动作	269
12.7.10	设置状态动作	270
12.7.11	隐藏动作	270
12.7.12	命名动作	270
12.7.13	NOP 动作	271

12.8	名字词典.....	271
12.9	信息词典.....	272
12.10	文章与文件标识符.....	273
12.10.1	文章.....	273
12.10.2	文件标识符.....	274
12.11	加密词典.....	275
12.11.1	标准加密.....	275
12.11.2	算法 6.7 计算加密关键词.....	276
12.12	Acrobat 表单.....	276
12.12.1	Acro 表单.....	277
12.12.2	域的一般概念.....	278
12.12.3	域词典.....	278
12.12.4	装饰注解.....	280
12.12.5	包含可变文本的域.....	280
12.12.6	按钮域.....	282
12.12.7	按钮、复选框与单选按钮.....	282
12.12.8	选择.....	283
12.12.9	文本域.....	284
12.12.10	AcroForm 动作.....	284
12.13	声 音.....	285
12.13.1	声音对象.....	285
12.13.2	外部声音.....	286
第十三章	公共数据结构.....	287
13.1	矩 形.....	287
13.2	日期.....	287
13.3	文件规格.....	288
13.3.1	文件规格串.....	288
13.3.2	文件规格词典.....	289
13.3.3	安全路径名.....	290
13.4	资源词典.....	290
13.5	字 体.....	291
13.5.1	Type 1 字体.....	291
13.5.2	TrueType 字体.....	292
13.5.3	字体子集与多主 Type 1 字体.....	293
13.5.4	Type 3 字体.....	293
13.5.5	Type 0 字体.....	294
13.5.6	CID 字体.....	295
13.5.7	垂直书写.....	295
13.5.8	字体编码.....	297

13.5.9	字符映射	297
13.5.10	字体文件	299
13.6	外部对象	299
13.6.1	图像	299
13.6.2	解码数组	300
13.6.3	颜色映射方式	300
13.6.4	表单外部对象	301
13.6.5	PostScript 外部对象	301
13.6.6	OPI 词典	302
13.7	函 数	302
13.7.1	基本概念	302
13.7.2	采样函数 Type 0	303
13.8	半色调功能	304
13.8.1	半色调名字与 Type 1 半色调	304
13.8.2	预定义网点函数	305
13.8.3	Type 5 半色调	314
13.8.4	Type 6 半色调	314
13.8.5	Type 10 半色调	315
13.8.6	Type 16 半色调	315
13.9	图 案	316
13.9.1	PDF 图案分类和图案矩阵	316
13.9.2	“瓷砖”图案	317
13.9.3	“平滑阴影”图案	318
13.10	阴 影	318
13.10.1	阴影的通用属性	319
13.10.2	颜色空间与插值计算	320
13.10.3	基于函数计算的阴影	321
13.10.4	轴向阴影	321
13.10.5	放射阴影	322
13.10.6	基于三角形网格体的自由阴影	324
13.10.7	“格子”阴影	326
13.10.8	“浣熊”块网格体构造阴影	326
13.10.9	张量阴影	330
第十四章	页面描述	332
14.1	概 述	332
14.1.1	图文对象类型	332
14.1.2	图文对象与操作次序	333
14.2	图文状态	334
14.2.1	专用图文状态	334

14.2.2	通用图文状态	335
14.3	颜 色	336
14.3.1	颜色参数	336
14.3.2	颜色操作符	337
14.4	路 径	338
14.4.1	路径段操作符	338
14.4.2	路径着色操作符	338
14.4.3	路径裁剪操作符	339
14.5	文本状态	339
14.5.1	文本状态参数和操作符	340
14.5.2	文本对象操作符	341
14.5.3	文本定位操作符	342
14.5.4	文本还原	342
14.5.5	文本串操作符	342
14.5.6	多字节字体的文本串	343
14.6	外部对象	343
14.7	行内图像	344
14.8	其他操作符	345
14.8.1	Type 3 字体操作符	345
14.8.2	兼容性操作符	345
14.8.3	标记内容操作符	345
	参考文献	347

上 篇

PostScript 语言及其程序设计

第一章 概 论

1.1 电子页面与页面描述语言

一、电子页面的概念

随着计算机技术和网络技术的迅猛发展，我们进入了一个高度信息化和数字化的时代，各种各样的信息通常组合在一起，以各种介质为载体，采用不同的方式和途径进行传播。但无论是利用哪一种传播方式，欲传播的信息往往首先要进行数字化处理，然后借助于某种技术或方法加以整合，形成一个包含了多种信息的电子页面。

电子页面是指由计算机生成的虚拟页面。它没有任何实体形态，而是以电子文件的形式存在，各个页面之间也没有实际的先后存放顺序，而只有逻辑上的先后关系。这些电子页面可以是与印刷品相对应的页面，也可以是电子出版中用于屏幕显示或在电子书阅读器上显示的页面。因此在实际的传播过程中，电子页面将根据印刷传播或电子传播方式的不同，在各种不同的光栅输出设备上输出。例如，在计算机显示器上直接显示，或直接应用于数字印前与数字印刷，在胶片照排机、印版照排机或数字印刷机上输出。

创作电子页面文件主要是通过计算机排版软件来完成的，而这样的排版软件多种多样，且其存在的文件格式也各不相同。但是，由于信息传播过程中的传播人员与接收者和阅读者往往会处于不同的工作和阅读环境之中，他们所使用的计算机种类及其操作系统、应用软件和输出设备一般会有所差异。如果信息传播的创作者们没有注意到这一情况，那么对于大多数电子页面文件来说，信息的传播必将遇到很大的障碍，出现因操作系统不兼容而产生的输出结果不一致现象，甚至根本无法打开等问题，或者因输出设备不同而产生色彩的巨大差异与失真，以及最终的信息内容无法真实地再现创作者们的最初意愿和传播目标。例如用 Microsoft Word 软件创作的 Word 文档，如果采用计算机屏幕显示的方式输出，就必须要求该电子页面文件阅读者的计算机中也安装了相应软件，否则将无法正确地显示页面内容与版面布局等。

上述提到的问题和现象的产生，其最根本的原因就在于传播过程中所创建的某些电子页面文件，是与它的生成环境和输出设备密切相关的，它的显示与输出完全依赖于操作系统、应用软件及输出设备，即使是在同一类输出设备上输出信息，在具体的每一个输出设备上得到的最终效果也会千差万别。

二、关于页面描述语言

理论上，任何页面上的图形、文字和图像等信息都可以用像素阵列来描述，但由于像素阵列非常庞大，且与设备有关，因此人们一般都不会采用这种方式来表示整个页面信息。

为了保证电子页面在传播过程中不因具体传播环境的不同而产生种种问题，早在 20 世纪 80 年代，许多公司的专业人员便开始致力于这方面的研究与开发，从而产生了各种不同的计算机语言。同时，根据电子页面最终输出时的目标和使用方式的不同，描述电子页面

时所采用的计算机语言也各不相同。例如，如果一个电子页面最终是为印刷出版和跨媒体出版服务，那么人们通常是采用 PostScript 等页面描述语言来描述这个电子页面。如果一个电子页面只是为网络出版服务，则主要是使用 HTML 或 XML 语言来描述页面内容。

应该说，凡是用于描述电子页面的计算机语言都可以称作页面描述语言 PDL (Page Description Language)。利用这些计算机语言可以得到一个紧凑格式的页面描述文件，这种描述有利于存储和传输，且与设备和操作系统无关。

但是，我们这里所讨论的页面描述语言的主要功能是在印刷或屏幕显示页面上描述文字、图形和图像。这一描述方法通常是以页面上的某一特征点为基准，把相应的页面内容定位于页面的某一位置，从而实现对电子页面内容和版式的描述与控制。而且，由于页面描述语言采用与输出设备且与输出介质无关的方式描述页面内容。因此这种描述文件的输出结果独立于任何设备和介质本身所具有的不同特性(例如，打印机的分辨率)，这便意味着所描述的电子页面不管是打印到纸上、照排输出到胶片、印版，还是在屏幕上显示，都不需要经过任何修改便能获得一致的视觉效果。

目前使用较多的页面描述语言有 Adobe 公司的 PostScript 语言、施乐公司的 Interpress 语言、Image 公司的 DDL 语言和惠普公司的 PCL5 语言等，其中最著名的就是 PostScript，其使用也最为广泛。

1.2 PostScript 语言简介

1.2.1 PostScript 语言的基本特征

一、关于 PostScript 语言

PostScript 语言作为现代印刷技术中的一种通用的页面描述语言，它是一种解释性的编程语言，具有很强的图形功能。PostScript 语言以 Adobe 公司的成像模型理论作为页面描述的依据，其主要应用是在印刷页面或显示页面上描述文本、图形形状和采样图像。利用 PostScript 语言写成的程序可以实现文档描述从组版系统到印刷系统的通信，或控制图文对象在显示器上的显示结果。由 PostScript 语言产生的描述是一种高等级的描述，且这种描述采用了与设备无关的方式。

PostScript 语言起源于 1976 年美国的 Evans & Sutherland 计算机公司，后几经修改，直到 1982 年由 Chuck Geschke 和 John Warnock 创建的 Adobe System 公司再次实现这一语言时才正式定名，并于 1985 年首先应用在 Apple 的激光打印机 Laser Writer 上，因此 Laser Writer 是世界上第一台配有 PostScript 语言解释器的激光打印机。

由于 PostScript 语言具有丰富的图形功能、高效地描述复杂页面的能力，以及设备无关性等优点，PostScript 语言在创建后的几年时间内便获得了广泛的应用，成为高质量图文印刷输出不可缺少的重要组成部分。目前，许多打印机、胶片照排机、印版照排机和数字印刷机等输出设备都安装了 PostScript 解释器；许多 RIP 处理软件也以 PostScript 解释器作为其核心技术；许多图像处理软件、图形设计软件和排版软件均支持 PostScript，其中最典型的有 Photoshop、CorelDraw、Illustrator、Freehand、QuarkXpress 和 PageMaker 等。

PostScript 语言的重要性已越来越多地被人们所认识，它不仅在印刷业中获得广泛应用，

成为数字印前、数字印刷以及相关领域事实上的工业标准，用于印刷图文信息的存储、传输、交换及输出的标准格式。PostScript 语言对电子出版也具有极其重要的意义，在 PostScript 语言基础上发展而来的 PDF 格式文件是电子出版中主要的电子页面文件格式，也是跨媒体出版中组织信息的主要方式。

二、PostScript 语言的基本特征

PostScript 语言的页面描述能力和交互式图文处理能力包括以下特征，这些特征可以任意的组合方式进行使用。

(1) 直线、弧线、矩形和三次曲线等基本图形可以构成任意的形状，且这些形状是可以自我包含的，允许有不连续的部分和孔洞。

(2) 着色操作符使得某一形状可以采用任意的线宽进行描绘，也可以任何颜色填充，或者用作裁剪路径去裁剪任何其他图文对象。用于描绘或填充形状的颜色可采用多种颜色模型来指定，包括灰度空间、RGB 颜色空间、CMYK 颜色空间或者基于 CIE 的颜色空间。其他某些特征也可用作特殊的颜色，例如重复出现的图案、平滑过渡的阴影、颜色映射和专色等。

(3) 文字完全按照图形方式进行处理。在 Adobe 的成像模型中，内置字库和用户自定义字库中的字符都作为图形形状来处理，PostScript 语言的任何图形操作符均适用于文字。

(4) 采样图像可以是自然界中的已有图片，例如照片经扫描后得到的图片，也可以是人工产生的图像。PostScript 语言可以按任意分辨率并根据不同的颜色模型来描述采样图像，而且提供了多种在输出设备上复制图像的技术。

(5) 通用坐标系统支持所有线性变换的组合，包括平移、缩放、旋转、反射和歪斜，这些坐标变换可一致地应用于文本、图形形状和采样图像等页面元素上。

PostScript 对页面的描述可在打印机、显示器或其他输出设备上还原，这可通过把页面描述送入 PostScript 解释器来控制输出设备产生相应的物理页面。当解释器执行控制命令对字符、图形形状和采样图像作着色处理时，该解释器按特定的输出设备将高端 PostScript 描述转换为栅格化的数据格式。

PostScript 程序可以采用 ASCII 源文本的形式来建立、传递和解释，PostScript 语言的各个版本均支持采用可打印字符和空白字符来描述页面中的图形、图像和文字。这些特点使程序设计者能方便地建立、操作和理解 PostScript 程序，也便于在不同种类的计算机和操作系统间存储和转移文件，增强了 PostScript 的设备独立性。

在某些合适的控制环境下，PostScript 语言程序也可采用二进制编码方式，例如当能保证 PostScript 语言程序与 PostScript 解释器之间具有完全透明的通讯路径时。但 Adobe 公司建议，当 PostScript 程序用作文件交换或存储时，应该采用 ASCII 编码表示方式。

三、PostScript 语言的发展版本

PostScript 语言自 1985 年正式面世后，因其强有力的编程功能、高效率和灵活性而获得了很大的扩展，比较典型的表现就是 PostScript 语言已经能适应新的成像技术或系统环境。尽管该语言增加了许多新的功能，进一步增强了语言的灵活性，但其基本的成像模型却保持不变。

PostScript 语言的功能扩展采用语言级 (Language Level) 的形式来表示。迄今为止，PostScript 语言分成三个等级，分别编号为 Level 1、Level 2 和 Level 3，较高等级的语言版

本包含低一级语言的全部功能以及大量的新增功能。Level 1 就是在 1985 年所创建的最初版本，Level 2 则于 1992 年在 Level 1 的基础上新增了许多扩展功能后命名产生的，1996 产生了 Level 3 版本。

对于任何一个支持某一语言等级功能的 PostScript 解释器来说，它必须能够实现该语言等级以及更低语言等级的全部功能，例如在 PostScript Level 2 版本中能够实现的功能也能在 Level 3 版本中实现。

PostScript 解释器还支持其他功能上的扩展，尽管这些功能没有包含在其基本语言等级中。其中有些扩展功能用于特殊的应用，而另外一些扩展功能则将作为普遍的应用要求而包含在以后新版本中。

对实际应用最有意义的是为显示 PostScript 系统（Display PostScript System）扩充出一套功能，这些功能使得工作站应用程序也能使用 PostScript 语言和 Adobe 公司的成像模型，管理页面对象的显示效果，并在工作站视窗操作系统环境下作交互式的显示。由于显示 PostScript 具有其特殊性，因而本书将不作具体介绍，有兴趣的读者可参阅由 Adobe 公司发布的有关显示 PostScript 的手册。

1.2.2 PostScript 语言的特点与功能

PostScript 是一种具有强大的图形描述功能的解释性编程语言。这一语言对页面内容的描述是高质量的，且与输出设备无关。

PostScript 语言最主要的功能就是根据 Adobe 的成像模型，描述印刷页面或显示页面上的文本、图形和采样图像的外观与页面位置。由此可见，通过 PostScript 语言，我们可以控制印刷页面或电子页面上的文字、图形、采样图像等的空间位置及其色彩信息。

PostScript 语言功能的实质是控制了输出设备的记录动作，即“告诉”显示器、打印机、照排机、数字印刷机等输出设备，如何在记录介质上产生最后的页面输出。因此，PostScript 语言也可称为是一种打印控制命令。

对 PostScript 语言而言，一方面它是一种具有很强图形描述功能的通用程序设计语言，另一方面又是一种具有一般程序设计语言特性的页面描述语言。因此，PostScript 语言具有通用程序设计语言和页面描述语言的双重特点。下面我们分别介绍 PostScript Level 1、PostScript Level 2、PostScript Level 3 三种版本的基本特点与功能。

一、PostScript Level 1 版本

PostScript Level 1 版本是 PostScript 语言的基本版本，其双重特点表现如下。

1. 通用程序设计语言方面

1) 数据以对象方式处理

PostScript 语言处理的每一个数据均称为 PostScript 对象。在 PostScript 语言中，可以管理和处理的对象类型有布尔型（boolean）、整型（integer）、实型（real）、数组（array）、串（string）、名字（name）、词典（dictionary）、操作符（operator）等。这些对象又可分为简单对象和复合对象。每一个对象都有自己的类型、属性和值。

2) 采用逆波兰表示法

PostScript 语言采用逆波兰表示法，又称后缀表示法，即采用操作数在前，操作符在后的编程方法。