

Q/HD

《中国学术期刊（光盘版）》电子  
杂志社有限公司企业标准

Q/HDXSQ0003-2015

---

XML 数据在不同场景的呈现模  
型

2015-12-03 发布

2015-12-03 实施

---

《中国学术期刊（光盘版）》电子杂志  
社有限公司 发布

# 目录

图目录.....	4
1 前言.....	1
2 范围.....	1
3 规范性引用文件.....	1
4 专业术语.....	1
4.1 可扩展标记语言 (XML) .....	1
4.2 电子出版 EPUB(Electronic Publication).....	1
4.3 OEBPS(Open eBook Publication Structure).....	1
4.4 都柏林核心集 Dublic Core.....	2
5 应用原则.....	2
5.1 规范使用主体.....	2
6 EPUB 规范.....	2
6.1 EPUB3 组成结构.....	2
6.2 OEBPS 与 Dublic Core 的关系.....	2
6.2.1 <dc:Title></ dc:Title >.....	3
6.2.2 <dc:Creator></ dc:Creator>.....	4
6.2.3 <dc:Subject></ dc: Subject >.....	4
6.2.4 <dc:Description></ dc: Description >.....	4
6.2.5 <dc:Publisher></ dc: Publisher >.....	4
6.2.6 <dc:Contributor> </dc:Contributor>.....	4
6.2.7 <dc:Date> </dc:Date>.....	5
6.2.8 <dc:Type> </dc:Type>.....	6
6.2.9 <dc:Format> </dc:Format>.....	6
6.2.10 <dc:Identifier> </dc:Identifier>.....	6
6.2.11 <dc:Source> </dc:Source>.....	6
6.2.12 <dc:Language> </dc:Language>.....	6
6.2.13 <dc:Relation> </dc:Relation>.....	6
6.2.14 <dc:Coverage> </dc:Coverage>.....	6
6.2.15 <dc:Rights> </dc:Rights>.....	7
6.3 EPUB 文件的基本结构.....	7
6.3.1 EPUB 文件的基本目录和文件结构.....	7
6.3.2 mimetype 文件.....	7
6.3.3 META-INF 文件夹.....	7
6.3.4 OPS 文件夹.....	8
7 EPUB 的排版和样式.....	9
7.1 基础知识.....	9
7.2 标签文本详解.....	9
7.2.1 标签.....	9
7.2.2 属性结构.....	10
7.2.3 元素.....	10
7.2.4 关闭标签.....	10
7.3 常用属性介绍.....	10

---

7.3.1	单位.....	10
7.3.2	常用属性.....	11
7.4	高级属性应用.....	11
7.4.1	属性.....	12
7.4.2	取值.....	12
7.5	CSS 外部样式表的应用.....	13
7.5.1	外部样式表结构.....	13
7.5.2	〈类选择器〉的应用.....	14
7.6	实例分析.....	14
7.6.1	在一个段落内使用多种尺寸的字体.....	15
7.6.2	悬挂缩进.....	15
7.6.3	首字下沉.....	16
7.6.4	引用文章.....	17
7.6.5	带边框与标题的图片.....	18
7.6.6	在狭小空间内插入多幅浮动图片.....	19
7.6.7	制作可爱的小提示.....	20

## 图目录

图 7-1 标签文本.....	9
图 7-2 标签嵌套框图.....	12
图 7-3 浮动示例图.....	13
图 7-4 外部样式示例.....	13
图 7-5 内联样式表转换成外部样式表.....	14
图 7-6 段落不同尺寸字体举例.....	15
图 7-7 悬挂缩进示例.....	15
图 7-8 首字下沉示例.....	16
图 7-9 引用文章示例.....	17
图 7-10 带边框与标题的图片.....	18
图 7-11 浮动属性出现问题.....	19
图 7-12 插入 clear 后的效果.....	20
图 7-13 提示示例.....	20

# 1 前言

本规范由《中国学术期刊（光盘版）》电子杂志社有限公司提出；  
本规范起草单位：《中国学术期刊（光盘版）》电子杂志社有限公司；  
本规范主要起草人：王明亮、张振海、熊海涛、梁洵、汪新红、丁慎训、万锦堃、李小红、欧坤、王国红、赵纪元、师庆辉、陈华、冯自强、康欢；  
本规范于 2015 年 12 月首次发布。

# 2 范围

本规范主要从 EPUB 的组成结构、与 DC 的关系以及排版和样式描述 XML 的呈现模型。  
本规范适用于学习需求驱动下的数字出版资源定制投送系统及应用示范项目。

# 3 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。

凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 1.1—2009	标准化工作导则
GB/T20000.1-2002	标准化工作指南
DB11/T 1000（所有部分）	企业产品标准编写指南
Open eBook Publication Structure 1.2	

# 4 专业术语

## 4.1 可扩展标记语言（XML）

标准通用标记语言的子集，是一种用于标记电子文件使其具有结构性的标记语言。

## 4.2 电子出版 EPUB(Electronic Publication)

是一个自由的开放标准，属于一种可以“自动重新编排”的内容，以最适于阅读的方式显示。

## 4.3 OEBPS(Open eBook Publication Structure)

开放式电子书出版结构。

## 4.4 都柏林核心集 Dublic Core

是基于网络信息资源的描述问题而建立的元数据，它对于目前搜索引擎对资源内容的抓取能起到准确定位的作用。

# 5 应用原则

## 5.1 规范使用主体

规范使用主体包括：

- a) 《中国学术期刊（光盘版）》电子杂志社有限公司；
- b) 浙江大学出版社有限责任公司；
- c) 清华大学图书馆。

# 6 EPUB 规范

EPUB 是一种基于 XML 的、对开发者友好的格式，不仅适用于图书，还包括：对文档打包以便离线阅读或者分发；打包博客文章或者其他 web 内容；使用常见的开放源代码工具创建、搜索和整理。可以使用各种开放源代码或者商业软件在所有主流操作系统、Sony PRS 之类的 e-ink 设备或者 Apple iPhone 之类的小型设备上阅读 EPUB 格式。

## 6.1 EPUB3 组成结构

EPUB3 是国际数字出版论坛表决通过的最新数字图书格式推荐规范。主要由四个独立的规范组合而成。每个规范都是 EPUB 数字图书的重要组成部分：(1)EPUB Publications 3.0，定义 EPUB 格式出版物层级的语意和整体符合性需求，包括包装文件的格式及此文档与其他出版资源（如音视频、图像、样式表、嵌入字体、脚本等）产生关联的规则，从而创建符合 EPUB 规范的出版物；(2) 定义了 EPUB 格式出版物的 XHTML、SVG 和 CSS 配置；(3) 定义了一种档案格式与处理模型，以封装一组相关资源至一个单一档案（ZIP）的 EPUB 容器；(4) 定义了同步多媒体整合语言（SMIL）、包装文件、EPUB 样式表，以及 EPUB 内容文件的用法，以实现出版物的文字和语音同步。

## 6.2 OEBPS 与 Dublic Core 的关系

设计都柏林核心集的目的：在提供足够有用数据的同时，将作者和出版者的编目负担减到最小。电子图书的内容提供者必须同时提供最基本的元数据元素，而且应该结合附加的元数据，使读者能够发现感兴趣的出版特征。

元数据元素用来提供整体出版物的信息。包含使用 dc-metadata 元素来记录一个 DC 元数据，并且使用 X-metadata 补充元数据。由 DCMI 定义的 dc-metadata 包含特定的出版等级

元数据。可选的 `x-metadata` 元素，若存在则必须包含一个或多个元数据实例，但必须作为整体应用于出版物。`x-metadata` 元素允许内容提供者超出都柏林规范中所描述的数据来表达任意元数据。本规范单独使用 OEBPS 包文件作为表达出版级都柏林核心元数据的基础。举例说明：

```
<metadata>
  <dc-metadata xmlns:dc=" http://purl.org/dc/elements/1.1/"
xmlns:oebpackage="http://openebook.org/namespaces/oeb-package/1.0/" >
  ...
  </dc-metadata>
  <x-metadata>
    <meta name="price" content="USD 19.99" />
  </x-metadata>
</metadata>
```

Xml 命名空间是用来识别都柏林核心元数据中无冲突的元素。并且在阅读系统中没必要处理命名空间。`dc-metadata` 元素可报刊任意都柏林核心元数据的任意数目的例子。都柏林核心元数据的命名以“dc:”为前缀，后面跟随一个大写字母。都柏林核心元数据可以以任何顺序出现，实际上相同元数据类型的多个实例可以穿插其他元数据元素是没有变化意义的。

每一个都柏林核心字段都是由一个元素表示，该元素的内容是字段的值。`dc-metadata` 元数据中至少包含 `dc:Title` , `dc:Identifier` 和 `dc:Language` 中的一个。都柏林核心元素，例如在 OEBPS 包文件中的其他元素，都有一个特定的 `id` 属性。从数据包中唯一标示符属性所引用的 `dc:Identifier` 中，至少一个 `dc:Identifier` 必须指定一个 `id`。

由于对于创造者和贡献者，都柏林核心元数据字段不能区分具体贡献者的身份，因此本规范也有对角色属性的区分。并且定义了可选文件作为元素属性以便于机器处理 `dc:Creator` 和 `dc:Contributor` 字段。该属性用于规定内容的标准化格式。本规范为 `dc:Identifier` 定义了一个 `scheme` 属性，以提供一个能够分离系统或者机构生成的标识符值的结构机制。本规范为 `dc>Date` 元素定义了一个事件属性，使得内容提供者区分各种出版物的日期。

例如：

```
<package unique-identifier="xyz">
  <metadata>
    <dc-metadata xmlns:dc="http://purl.org/dc/elements/1.1/"xmlns:oebpackage="http://
openebook.org/namespaces/oeb-package/1.0/">
      <dc:Title>Alice in Wonderland</dc:Title>
      <dc:Language>en</dc:Language>
      <dc:Identifier id="xyz"
scheme="ISBN">123456789X</dc:Identifier>
      <dc:Creator role="aut">Lewis Carroll</dc:Creator>
    </dc-metadata>
  </metadata>
  ...
</package>
```

## 6.2.1 <dc:Title></ dc:Title >

出版物题名，一个 OEBPS 包必须包含至少一个该元数据类型实例，也就是允许多实例。

任何一个向用户显示题名元数据的电子阅读系统应该使用第一个 `dc>Title` 或者所有的 `dc>Title` 元素。

## 6.2.2 `<dc:Creator></ dc:Creator>`

主要的出版物创作者或者作者。其他贡献者应该应该在 `dc:Contributor` 元素中列出来。当出版物有多个共同创作者时，应该提供多个 `dc:Creator` 元素，每一个元素中都包含一个作者。`dc:Creator` 元素的顺序用来定义其作者在阅读系统中显示的顺序。建议 `dc:Creator` 元素的内容存放一个单独的姓名文本，因为该元素将代表一个使用者。本规范为 `dc:Creator` 元素定义了两个可选的属性：角色和文件。角色设定的值与 `dc:Contributor` 元素所定义的值是相同的。文件属性应当用来指定一个标准化形式的内容。例如：

```
<dc:Creator file-as="King, Martin Luther Jr." role="aut">
    Rev. Dr. Martin Luther King Jr.
</dc:Creator>
```

如果一个阅读系统显示作者信息，那么该系统必须按照提供的顺序，使用适当的间距或者标点来显示所有的 `dc:Creator` 元素的内容。

## 6.2.3 `<dc:Subject></ dc: Subject >`

主题词或关键字，支持多个 `dc:Subject` 元素的示例，每一个都包含一个任意的短语或者关键字。

## 6.2.4 `<dc:Description></ dc: Description >`

出版物的描述。

## 6.2.5 `<dc:Publisher></ dc: Publisher >`

出版物的出版者。

## 6.2.6 `<dc:Contributor> </dc:Contributor>`

并没有在 `Creator` 元素中列出对资源的知识内容具有重要奉献的实体，所起贡献次于主要责任者。阅读系统可以随意选择显示的 `dc:Creator` 信息，不需要伴有 `dc:Contributor` 信息。本规范定义了 `dc:Contributor` 两个可选的属性，角色和文件。

用于角色属性的规范列表是由 MARC 代码列表定义的。当指定角色时，三个特性注册的 MARC 值必须在可适用的时候使用。由于该列表是广泛的，没有预先定义的一些所必需的角色是可以添加进来的。这些值必须以 'oth' 开始，并被视为其他子结构的分支。这些值是区分大小写的，并且必须完全以小写来编码。

依照 MARC 编码一览表列出以上提出的关系编码值：

Adapter[adp]，一个二次加工（通常是不同的媒介）的音乐作品人，改写电影或其他视



听媒体的作者。

Annotator[ann], 在出版条目中进行标注注解的人。

Arranger[arr], 在一个不同媒体录制音乐作品的制作人, 音乐实体保持不变。

Artist[art], 构思或者创作实现一个图形设计或者艺术作品的人, 例如图书插画家。

Associated name[asn], 在一个编目或者征文中相关的名字的责任人, 或者不能确定前所有人的原始出处。

Author[aut], 主要负责作品的知识或者艺术内容的个人或团体。也可用在责任人多于一个人或者团体的情况。

Author in quotations or text extracts[aqt], 他或她对作品没有直接贡献, 但是其作品被大量引用或提取。多数情况在展览目录或图片收藏中出现。

Author of afterword, colophon, etc. [aft], 负责后记、刊后语、版权页等的作者或者团体, 但都不是作品的主要作者。

Author of introduction, etc. [aui], 负责简介、前言、引言或者其他关键事件的作者或者团体, 但都不是作品的主要作者。

Bibliographic antecedent [ant], 负责日志记录的作者, 可能是适当的修改、成果、延续、索引等。

Book producer [bkp], 负责生产书籍或者其他印刷产品的个人或企业。

Collaborator [clb], 部分参与制定另外一个作者作品的个人或团体。

Commentator [cmm], 负责对一个记录、电影或者其他视听媒体的主题题材进行讨论分析的人。

Compiler [com], 在生产一个产品或者出版物时负责选择整理各种不同人或者机构的产品题材的人。

Designer [dsr], 负责设计的个人或机构。

Editor [edt], 负责出版不是自己主要作品的人, 例如通过阐述文案, 增加引言或者其他主要事项, 或者技术性指导的编辑人员。

Illustrator [ill], 负责构思、实现、设计或者插图的人, 书面文案的配图。

Lyricist [lyr], 写一首歌歌词的人。

Metadata contact [mdc], 负责编写和维护一个元数据集的初始描述。

Musician [mus], 演出音乐或者有主意作品的音乐内容的人。

Narrator [nrt], 某个行为、事件或事件过程的相关细节的发言人。

Other [oth], 没有相应的 MARC 表的其他人。

Photographer [pht], 在原始产品或者再出版中负责拍照的个人或组织机构。

Printer [prt], 负责印刷的类型和材质的个人或组织机构。

Redactor [red], 负责一个开发项目的框架而不是其内容的人。

Reviewer [rev], 负责审查书籍、视频或表演的个人或组织机构。

Sponsor [spn], 发表合约的个人或机构, 或者在其领导下进行编写, 印刷或者出版。

Thesis advisor [ths], 指导攻读学位人、发表论文、研究报告或论文的导师。

Transcriber [trc], 负责从原始材料中准备手写或者打印副本的人, 包括口述或者口头形式的记录材料。

Translator [trl], 负责翻译的人。

## 6.2.7 <dc:Date> </dc:Date>

与出版相关的时间。日期格式根据 ISO8601。特别地, 当日期没有时间时, 用

YYYY[-MM[-DD]]格式表示：4 位数年份、可选的 2 位月份，如果月份给出了，可选的两位表示月份中的天。dc:Date 元数据有一个可选的属性：事件。包括：创建、出版、变更。

## 6.2.8 <dc:Type> </dc:Type>

资源内容的内在属性、形态或类型，例如主页、小说、诗歌、手稿、技术报告、论文、词典等。

## 6.2.9 <dc:Format> </dc:Format>

资源外在的物理特征或数字形式，如资源的媒体、尺寸或周期，如软件、硬件、HTML 等。

## 6.2.10 <dc:Identifier> </dc:Identifier>

用来唯一标识资源的字串或数字。一个 OEBPS 包必须包含至少一个该元数据类型实例，也就是允许多实例。dc:Identifier 中必须指定一个标识符 id，用以被包中唯一属性来引用。dc:Identifier 元素具有可选属性，例如网络资源标识中的 URL、URN、DOI，以及其他通用唯一性标识，如 ISBN。

## 6.2.11 <dc:Source> </dc:Source>

二次资源的出处信息，如果当前资源为其原始形式，来源元素不可用。

## 6.2.12 <dc:Language> </dc:Language>

出版物学术资源内容的语言标识符。一个 OEBPS 包必须包含至少一个该元素类型实例，也就是允许多实例。该元素据的内容必须依照 RFC3166。

## 6.2.13 <dc:Relation> </dc:Relation>

出版物的参考资源或者有关系的资源标识符。

## 6.2.14 <dc:Coverage> </dc:Coverage>

资源内容的领域或范围；范围包括空间定位（地名或地理坐标），时代（年代、日期或日期范围）或权限范围。

## 6.2.15 <dc:Rights> </dc:Rights>

一个权限管理的陈述标识，或者指向一个权限管理的陈述标识。应当直接写出版权声明或者进一步的权利描述。

## 6.3 EPUB 文件的基本结构

### 6.3.1 EPUB 文件的基本目录和文件结构

EPUB 文件的基本目录和文件结构遵循如下样式

```
mimetype
META-INF/
  container.xml
OPS/
  content.opf
  title.html
  content.html
  stylesheet.css
  toc.ncx
  images/
    cover.png
```

### 6.3.2 mimetype 文件

每一本 EPUB 格式电子书中必须包含一个“mimetype”文件。这个文件内容十分简单，只有一条语句“application/epub+zip”，不能包含新行或者回车且必须作为 zip 压缩包中的第一个文件。其作用是规定电子书能通过 EPUB 阅读设备和 zip 压缩软件打开。

### 6.3.3 META-INF 文件夹

META-INF 是 OCF 抽象容器的必要目录。该目录主要描述电子书的位置、打开方式、信息加密、目录、书籍资料、权限管理等信息。其中必须要有 container.xml 容器文件，在这个文档中必须清楚指明容器所包含的 EPUB 电子书根文件的媒体类别与路径，该文件是不能加密的。文件内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<container version="1.0" xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
  <rootfiles>
    <rootfile full-path="OEBPS/content.opf"
      media-type="application/oebps-package+xml" />
  </rootfiles>
</container>
```

## 6.3.4 OPS 文件夹

OPS 文件夹存放了电子书的全部内容，电子书的每一个章节的内容、图片、音频、视频，在阅读器上显示的字体、样式以及电子书内部结构.opf 文件都在 OPS 中。

### 6.3.4.1 content.ops 文件

该文件制定了所有内容的位置并描述所有具体内容，是 EPUB 规范中最复杂的部分，打开文件后，根据规范中各个元件分别显示其内容语句。

#### 1.<package>元件

这是包裹文件的根容器，`unique-identifier="bookid"`用以识别<dc: identifier>元件。

#### 2.<metadata>元件

该元件没有定义属性，是<package>的第一个子元件，描述电子书的相关信息。例如：书名、作者、语言、出版社、单一识别符、语言等。

#### 3.<manifest>元件

EPUB 电子书的详细资源清单由<manifest>元件构成。它是<package>第二个必要的子元件，每一个资源用一个<item>元件表示。其中<id>属性必须在此文件范围内是唯一的；<href>属性制定所对应条目的位置；<media-type>属性说明条目中出版物的媒体类型和格式；<media-overlay>属性定义条目所描述的媒体叠加层文件。可包括：电子书字体、目录、版权页、书名页、定义封面和封面图片路径、电子书排版的 CSS 样式位置、书名页图片的读取地址、前言、简介、评论与引语等。

#### 4.<spine>元件

该元件制定 EPUB 电子书内容的预设阅读顺序。<itemref>元件是<spine>中的子元件，表示出版物资源的顺序。每个<itemref>元件都需要一个<idref>属性，并且和 manifest 中的某个 ID 匹配。包含元数据的 OPF content 文件如下所示。

```
<?xml version='1.0' encoding='utf-8'?>
<package xmlns="http://www.idpf.org/2007/opf"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          unique-identifier="bookid" version="2.0">
  <metadata>
    <dc:title>Hello World: My First EPUB</dc:title>
    <dc:creator>My Name</dc:creator>
    <dc:identifier id="bookid">urn:uuid:12345</dc:identifier>
    <meta name="cover" content="cover-image" />
  </metadata>
  <manifest>
    <item id="ncx" href="toc.ncx" media-type="text/xml"/>
    <item id="cover" href="title.html" media-type="application/xhtml+xml"/>
    <item id="content" href="content.html" media-type="application/xhtml+xml"/>
    <item id="cover-image" href="images/cover.png" media-type="image/png"/>
    <item id="css" href="stylesheet.css" media-type="text/css"/>
  </manifest>
  <spine toc="ncx">
```

```

    <itemref idref="cover" linear="no"/>
    <itemref idref="content"/>
</spine>
<guide>
    <reference href="cover.html" type="cover" title="Cover"/>
</guide>
</package>

```

## 7 EPUB 的排版和样式

### 7.1 基础知识

ePub 格式电子书是在 html 与 css 等规范上确立的一种 eBook 规范。就是将文本、图片、样式等所有 ePub 中允许使用的素材制作成我们平时看到的网页，并通过结构文件使得阅读器能够分章索引。

在 ePub 中，所有的都是通过标签文本实现的：如下图所示：



图 7-1 标签文本

被一对标签所包括的内容成为<元素>，这段代码就是一个<元素>，其中：p 是<开始标签>，代表一个段落的开始，style="text-align:center"被成为<属性>，声明了一个标签所具有的值。/p 被称为<关闭标签>，代表一个段落的结束。<开始标签>与<关闭标签>是成对的。所有的<标签>必须被关闭，“这是一段文本”是<元素>的内容。被一对标签包括的内容都成为<元素>。

### 7.2 标签文本详解

在 epubBuilder 中，〈标签〉则是用方括号（[]）所包括。epubBuilder 会自动为所有的段落添加 p 与 /p 用来标明段落，所以一般情况下不用你自行添加标签。为标签添加属性的做法叫做〈声明〉或者〈说明〉。标签文本必须使用小写字母与符号。

#### 7.2.1 标签

标签有很多种类，在 epubBuilder 中经常能看到的有：

[div]	代表一块内容
[p]	代表一个段落
[span]	用于段落中
[h3]	代表标题
[a]	代表链接

[img] 代表图片

## 7.2.2 属性结构

`style="text-align:center"`,在该属性中：`style` 是属性的类型；`text-align:center` 是属性的名称与赋值。属性写在〈开始标签〉中，并且与〈标签〉相隔一个空格；名称与赋值必须被引号包括。

多个赋值之间使用英文分号(;)隔开。例如：`[p style="text-indent: 0em;text-align:center"]`。

## 7.2.3 元素

`[p style="text-align:center"]`这是一段文本`[p]`。元素的显示方式由〈开始标签〉决定；以上内容被称为〈p 元素〉。如果标签改为 `div`，那么这个元素就被称为〈div 元素〉。元素可以是文本、图片，甚至是另一个元素。

例如：`[p style="text-align:center"]`这是一段`[span]`文本`[/span][p]`。上述范例中，一个〈p 元素〉中包括了一个〈span 元素〉。元素有包含顺序，你不能在一个〈span 元素〉中加入一个〈p 元素〉。

常用标签包含顺序：`〈html〉 → 〈body〉 → 〈div〉 → 〈p〉 → 〈span〉`。

另外，还有一些元素没有实际内容，这类元素被称为〈空元素〉。〈空元素〉不需要关闭标签，直接在开始标签中加 `/` 即可关闭。经常用到的空元素有：`[br /]` 代表一个空行。`[hr /]` 代表一条横线。

## 7.2.4 关闭标签

`(/p)`，在关闭标签中：`/`声明了这是一个关闭标签；`P` 则是声明了你要关闭哪一个标签。如果涉及元素包括，关闭标签必须按照顺序关闭：例如开始标签是 `[p][span]`那么必须先关闭 `span`，然后再关闭 `p`。`[div][p][span]元素[/span][p][/div]`,所有标签必须被关闭。

## 7.3 常用属性介绍

### 7.3.1 单位

以下是一些常用的计量单位。

%相对于默认值的百分比；

px 像素，1px 就是 1 个像素的高度或者宽度；

em 字符，1em 相当于 1 个字符的高度或者宽度。这是一个相对值。比如你设置了默认字符大小为 16px，那么 1em=16px。如果默认字符大小是 20px，那么 1em=20px。

## 7.3.2 常用属性

1. `text-indent`, `text-indent` 用来控制段首缩进。`epubBuilder` 默认 `text-indent=2em`，如果需要其他值，在属性中添加 `text-indent` 并设置数值。范例：`[p style="text-indent: 0em"]`这段文本没有段首缩进。`[/p]`

`text-align`, `text-align` 用来控制段落的水平对齐方式。默认值为 `<left>`。取值：

`left` 左对齐（默认）

`right` 右对齐

`center` 居中对齐

`justify` 两端对齐

范例：`[p style="text-align:center"]`这段文本将会居中显示`[/p]`。

2. `font-size`, `font-sizes` 用来控制文字的大小。默认值为 `<16px>`。`font-sizes` 也可使用 `large`、`medium`、`small` 等描述来赋值。范例：`[p style="font-size:25px"]`这段文字的大小是 25 个像素。`[/p]`，`[p style="font-size:1.3em"]`这段文字的大小是默认的 1.3 倍。`[/p]`。

3. `color`, `color` 用来控制文字的颜色，这个功能已经被集成在了 `epubBuilder` 中。

4. `font-weight`, `font-weight` 用来控制文字的粗细，默认值是 `<normal>`。虽然 `font-weight` 可以用 `b` 标签来取代，但是在标签文本中使用时不建议这么做。取值：

`bold` 加粗

`bolder` 超粗

`lighter` 变细

范例：`[p style="font-weight:bold"]`这段文本的字体加粗`[/p]`。

5. `font-style`, `font-style` 属性控制文字的样式，默认值为 `<normal>`。与 `font-weight` 属性一样，虽然 `font-style` 可以用 `i` 标签代替，但是在标签文本中不建议这么做。取值：`italic` 斜体。范例：`[p style="font-style:italic"]`这段文本会显示为斜体字。`[/p]`

6. `line-height`, `line-height` 属性用来控制行距。最小值是 0，不可以是负数。当 `line-height=0` 时，段落将不产生换行行为。超出浏览器的文字将直接显示到浏览器外，而不是换一行继续显示。注意：这个属性控制的是行与行之间的距离，而不是段落之间的距离。范例：`[p style="line-height:1em"]`这段文本的行距是 1 个字符的高度`[/p]`

## 7.4 高级属性应用

主要针对 `width`、`height`、`margin`、`padding`、`float` 这 5 个属性的基本应用。“每一个元素都是一个框！”，所有的元素都是一个框型结构。一个元素包括另一个元素的行为，就是一个大框中套着一个小框。例如：

`[div]`

`[p]`这是一个段落。`[/p]`

`[p]`这是一个段落。`[span]`这是一段文本。`[/span]``[/p]`

`[/div]`

可将其想象成下图：



图 7-2 标签嵌套框图

## 7.4.1 属性

1. **width**: 顾名思义，控制〈元素〉的宽度。也就是〈框〉的宽度。
2. **height**: 控制〈元素〉的高度。也就是〈框〉的高度。
3. **margin**: 控制元素的〈外边距〉。也就是〈框〉与其他〈框〉之间的距离。默认值为 0。
4. **padding**: 控制〈元素〉的〈内边距〉。也就是〈框〉与元素中所包含内容的距离。默认值为 0。
5. **float**〈浮动〉: 控制元素的〈浮动〉，取值是 **left** 与 **right**。

## 7.4.2 取值

1. **width** 与 **height** 使用数值，最小是 0。 **margin** 与 **padding** 使用数值，但是赋值方法比较特殊，有多种赋值方式；以 **margin** 属性为例：**margin: 1px**

方法一：在 **margin** 属性中对所有的〈外边距〉赋予同一个值。**margin: 1px 1px 1px 1px**。

方法二：在 **margin** 属性中同时对四个边距分别进行声明。顺序是“上、右、下、左”。**margin-top: 1px; margin-right: 1px; margin-bottom: 1px; margin-left: 1px**。

方法三：使用独立的 **margin** 属性，分别声明对应方向的〈外边距〉。

上述三种赋值方式，效果是一样的。如何使用，则要看具体需求。比如我只需要为元素右侧增加〈外边距〉时，可以使用如下代码：**[p style="margin-right: 150%"]**这段文字的右侧外边距是默认的 150%。**[/p]**

2. 下一个属性：**float**〈浮动〉的介绍。元素与元素之间默认的排列规则总是从上往下的。文本按照段落依次由上往下排列；第 6 行文本后插入的图片无法跑到第 5 行；如果要打破这个规则，使元素按照我们需要的方式排列呢？这时候就需要 **float** 属性，使元素从固定的位置〈浮动〉起来。浮动是一项很复杂的属性。**float**，控制元素的〈浮动〉，取值是 **left** 与 **right**。  
范例：**[p style="float: right"]**这段文字会向右浮动。**[/p]**

在制作 ePub 的过程中，**float** 属性主要应用于图片。如下图所示：





图 7-3 浮动示例图

在这个例子中，图片脱离了规则约束，〈浮动〉到了段落的右侧。而下方的文字则被上提，填补了这个空位。为了使〈浮动〉后的排版更加美观，可搭配 `margin` 属性一同使用。

## 7.5 CSS 外部样式表的应用

### 7.5.1 外部样式表结构

将所有的属性放在开始标签中，直接对元素进行〈声明〉的方法，称为〈内联样式表〉。〈内联样式表〉非常直观，可以说是“所见即所得”。但如果有些样式需要大量重复使用，或者制作一些比较复杂的样式时，〈内联样式表〉就显得比较麻烦了。

ePub 的精髓所在，css 〈外部样式表〉。由于 css 样式表是一个相当复杂的内容，这里只介绍基本应用方法。了解一下外部样式表的结构：



图 7-4 外部样式示例

1.选择器：确定你要改变的〈元素〉。上图所示即是对于〈p 元素〉所做的〈声明〉。

2.属性：属性需要用花括号（{}）包括。与〈内联样式表〉不同的是，你可以像上图所示那样，使用每行撰写一个属性的方式来进行〈声明〉，多个属性用英文分号隔开。这种做法更加方便查看。也可以写成这样的格式：

```
p {text-indent:2em;text-align:justify;line-height:130%;margin-top:5px;margin-bottom: 5px;}
```

css 〈外部样式表〉的优势在于可以很方便地定义全局属性。如果你在〈外部样式表〉中〈声明〉了〈p 元素〉，那么整个 ePub 中的所有段落都将根据你〈声明〉的样式来显示。

由于 epubBuilder 没有集成 css 编辑功能。所以如果我们要更改 css 〈外部样式表〉，需要打开制作好的 ePub 文件，将.css 文件修改好之后再替换回去。在 css 〈外部样式表〉中为〈p 元素〉增加一项属性 `color:red`。将 css 覆盖回去后，用阅读器打开 ePub。

## 7.5.2 〈类选择器〉的应用

〈类选择器〉是一个可以自定义名称的选择器。需要配合〈元素〉标签使用。这是一个典型的〈类选择器〉：

```
.red-big {
    font-size:1.5em;
    text-indent: 0em;
    color:red;
    font-weight:bold;
}
```

建立〈类选择器〉用小数点开始，它的名称是可以自定义的。任何字母、数字，以及横杠（-）、下划线（\_）等都可以使用。但是绝不可以使用数字开头。

```
.style-1 这样没问题
.1-style 这样是错误的
.123     同样是错误的
```

在定义名称时，建议使用自己能记住的、并且能联想到其中所〈声明〉的样式的名称。例如 **bigfont-red**、**float-img**，甚至是拼音。

看一下〈内联样式表〉是如何转换成〈外部样式表〉的〈类选择器〉的：



图 7-5 内联样式表转换成外部样式表

如上图所示，当需要在文档中使用〈类选择器〉所〈声明〉的样式时，需要利用〈class〉属性种类来调用它。

范例：`[p class="xxxx"]`这段文本将使用 `xxxx` 中〈声明〉的样式来显示。`[/p]`

也可以在一个标签中调用多个〈类选择器〉，中间用一个空格隔开：`[p class="xxxx oooo"]`这段文本将调用 `xxxx` 与 `oooo` 两个〈类选择器〉。`[/p]`

使用类选择器的优势在于，可方便地管理那些格式化的、经常使用的排版样式。而且对日后的编辑相当有好处。例如当发现文档中有一处错误：使用〈内联样式表〉的话，就要到数量庞大的文档中去寻找这条出错的〈声明〉；而使用〈外部样式表〉，只要打开 `css` 文件，找到这个〈类选择器〉修改即可。当熟练运用〈内联样式表〉之后，使用〈类选择器〉来替换所有的〈内联样式表〉。

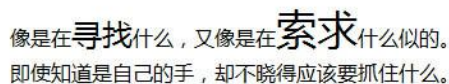
## 7.6 实例分析

举例说明分析其 **ePub** 制作思路。实例将以 `css` 〈类选择器〉的形式给出属性代码。包括以下几个部分：

1. 在一个段落内使用多种尺寸的字体
2. 悬挂缩进

3. 首字下沉
4. 引用文章
5. 带边框与标题的图片
6. 在狭小空间内插入多幅浮动图片
7. 制作可爱的小提示

## 7.6.1 在一个段落内使用多种尺寸的字体



像是在寻找什么，又像是在索求什么似的。  
即使知道是自己的手，却不晓得应该要抓住什么。

图 7-6 段落不同尺寸字体举例

代码:

```

<外部样式表>
.s1 {
font-size:1.5em;
}
.s2 {
font-size:2em;
}
<文本>

```

即使知道是自己的手，却不晓得应该要抓住什么。

分析：font-size 属性的一个简单应用；使用〈span〉行内标签对段落内需要放大的字体进行标记。

## 7.6.2 悬挂缩进

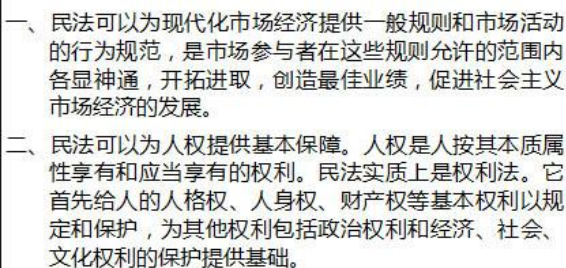
- 
- 一、民法可以为现代化市场经济提供一般规则和市场活动的行为规范，是市场参与者在这些规则允许的范围内各显神通，开拓进取，创造最佳业绩，促进社会主义市场经济的发展。
  - 二、民法可以为人权提供基本保障。人权是人按其本质属性享有和应当享有的权利。民法实质上是权利法。它首先给人的人格权、人身权、财产权等基本权利以规定和保护，为其他权利包括政治权利和经济、社会、文化权利的保护提供基础。

图 7-7 悬挂缩进示例

代码:

```

<外部样式表>
.s1 {
text-indent: -2em;
margin-left: 2em;
}

```

〈文本〉

[p class="s1"]一、民法可以为现代化市场经济提供一般规则和市场活动的行为规范，是市场参与者在这些规则允许的范围内各显神通，开拓进取，创造最佳业绩，促进社会主义市场经济的发展。 [/p]

[p class="s1"]二、民法可以为人权提供基本保障。人权是人按其本质属性享有和应当享有的权利。民法实质上是权利法。它首先给人的人格权、人身权、财产权等基本权利以规定和保护，为其他权利包括政治权利和经济、社会、文化权利的保护提供基础。 [/p]

分析：根据段首标题的字数，使用相应的负缩进值使段首的标题突出段落外；同时使用〈外边距〉属性将整个段落向右缩排，实现标题悬挂于段落之外。必须使用外边距将段落向右缩排，否则使用负缩进值时，段落将显示到浏览器外。

### 7.6.3 首字下沉

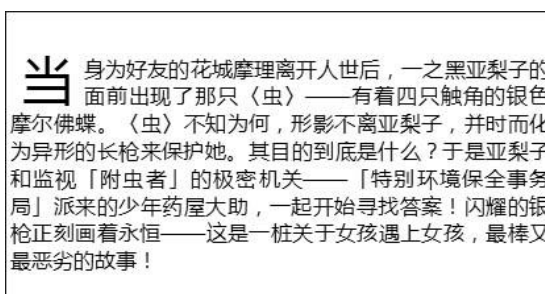


图 7-8 首字下沉示例

代码：

〈外部样式表〉

```
.s1 {
float:left;
font-size:2.5em;
margin:0.2em;
}
.indent-0 {
text-indent: 0em;
}
```

〈文本〉

[p class="indent-0"][span class="s1"]当[/span]身为好友的花城摩理离开人世后，一之黑亚梨子的面前出现了那只〈虫〉——有着四只触角的银色摩尔佛蝶。〈虫〉不知为何，形影不离亚梨子，并时而化为异形的长枪来保护她。其目的到底是什么？于是亚梨子和监视「附虫者」的极密机关——「特别环境保全事務局」派来的少年药屋大助，一起开始寻找答案！闪耀的银枪正刻画着永恒——这是一桩关于女孩遇上女孩，最棒又最恶劣的故事！ [/p]

分析：这个样式是在外文书中常见的样式。在中文书中偶尔也能见到。将段落的第一个字标记为〈span 元素〉，然后使用〈浮动〉属性，使其脱离文本悬浮于段落左上角，同时通过外边距属性控制与其他文本的距离。由于 epubBuilder 默认段首缩进 2em，所以制作首字下沉的时候，需要对段落使用〈txt-indent:0em〉覆盖默认属性。

## 7.6.4 引用文章

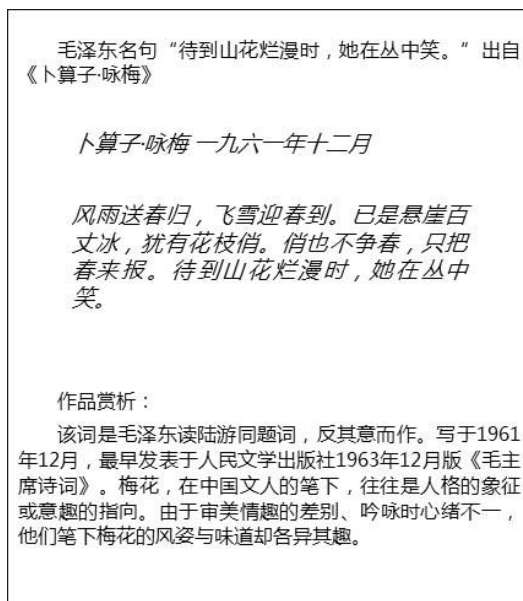


图 7-9 引用文章示例

代码:

〈外部样式表〉

```
.s1 {
text-indent: 0em;
font-size: 1.1em;
margin-left: 2.5em;
margin-right: 2.5em;
font-style: italic;
}
```

〈文本〉

毛泽东名句“待到山花烂漫时，她在丛中笑。”出自《卜算子·咏梅》

[br/]

[p class="s1"]卜算子·咏梅 一九六一年十二月[/p]

[br/]

[p class="s1"]风雨送春归，飞雪迎春到。已是悬崖百丈冰，犹有花枝俏。俏也不争春，只把春来报。待到山花烂漫时，她在丛中笑。[/p]

[br/]

[br/]

作品赏析:

.....

分析：这个例子是对外边距属性的一个简单应用。对段落的左右外边距进行缩排，使得引用的文本脱离正文；同时声明了较大并且倾斜的字体，使引用的文本更加醒目，易于辨认。为了使引用的文字脱离正文，这里使用了〈br〉标签插入了空行。在类似应用中，建议使用〈br〉标签插入空行，而不是使用外边距属性增加段落之间的距离。因为如今的 ePub

阅读器并不是全都支持外边距属性，但是对〈br〉标签的支持较好。

## 7.6.5 带边框与标题的图片



图 7-10 带边框与标题的图片

代码:

```

<外部样式表>
.img-s1 {
float:right;
border-style:solid;
border-width:1px;
margin:0.4em;
padding:0.4em;
max-width:50%;
text-align:center;
}
.s1 {
font-size:1.3em;
text-indent: 0em;
text-align:center;
font-weight:bold;
}
<文本>
[div class="img-s1"][img src="images/1.jpg"/][p class="s1"]霍华德·休斯[/p][div]
[br/]
霍华德·休斯
(Howard Robard Hughes, Jr.1905-1976)
.....

```

分析:

在这个实例中，我们引入了一个新的属性〈border〉（边框）。

border-style

决定一个元素使用什么样子的边框，默认值为〈none〉。

常用取值:

solid 实线  
 double 双线  
 dotted 圆点  
 dashed 虚线  
 border-width  
 决定边框的宽度  
 border-color  
 决定边框的颜色

在制作这个带标题的图片时，首先利用〈div 元素〉来制造一个（箱子），将图片与文字装在里面；接着使用〈border-style〉来为这个〈div 元素〉加上边框。与此同时对〈div 元素〉内的文本声明一个较大的字体并且居中。即可产生一个带边框与标题的图片。

## 7.6.6 在狭小空间内插入多幅浮动图片

〈浮动〉属性可以很方便的制作文字绕图的效果。但是有时候我们要在狭小的空间里插入多幅〈浮动〉的图片时，可能会变成如下情况：

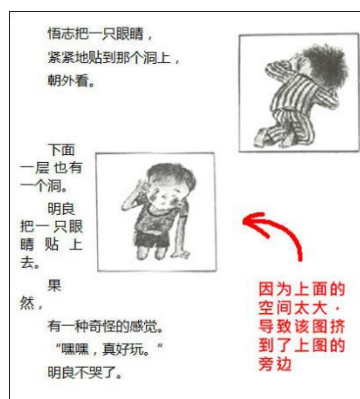


图 7-11 浮动属性出现问题

代码：

```

<外部样式表>
.img-float-right {
float:right;
margin:0.5em;
max-width:50%;
}
<文本>
[div class="img-float-right"][img src="images/1.png"][/div]
悟志把一只眼睛，
紧紧地贴到那个洞上，
朝外看。
[br/]
[br/]
[div class="img-float-right"][img src="images/2.png"][/div]
下面一层也有一个洞。
明良把一只眼睛贴上去。

```

这里希望在文本的右侧插入两幅图片，但是因为〈浮动〉属性的特点，两幅图片靠在了一起。如果遇到这种情况，就需要〈clear〉（清理）属性来解决这个问题。

**clear**

规定元素的两侧是否允许出现〈浮动〉元素。

取值：

**left** 右侧不能出现浮动元素

**right** 左侧不能出现浮动元素

**both** 两侧均不允许

在这个实例中加入〈清理〉属性后，就得到了想要的效果：



图 7-12 插入 clear 后的效果

代码：

```
.img-float-right {
float:right;
margin:0.5em;
max-width:50%;
clear:right;
}
```

## 7.6.7 制作可爱的小提示

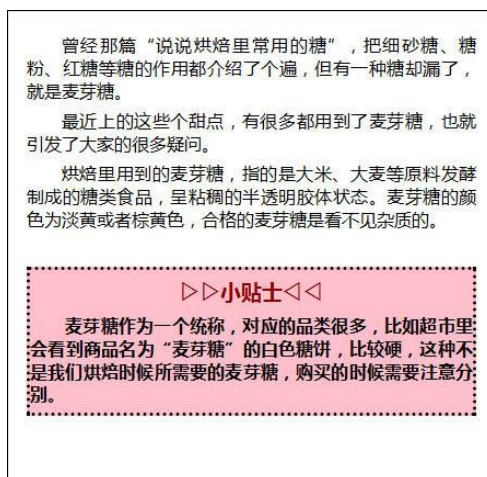


图 7-13 提示示例



代码:

```

<外部样式表>
.box {
background-color:pink;
border-style:dotted;
border-width:3px;
}
.s1 {
color:darkred;
font-size:1.2em;
text-indent:0em;
text-align:center;
font-weight:bold;
}
.s2 {
font-weight:bold;
}
<文本>

```

.....

.....

[div class="box"][p class="s1"]》小贴士 《《[p

[p class="s2"]麦芽糖作为一个统称，对应的品类很多，比如超市里会看到商品名为“麦芽糖”的白色糖饼，比较硬，这种不是我们烘焙时候所需要的麦芽糖，购买的时候需要注意分别。[/p][div]

分析:

这里使用了一个新属性:

<background-color> (背景颜色)

**background-color**

设置元素的背景颜色。取值与<color>属性相同。这个实例的制作思路与上面的“带边框与标题的图片”基本相同。对整个<div元素>声明了点状的边框与粉红色的背景颜色，同时声明了其中的文字样式。这个实例还可以配合<浮动>与<外边距>属性来制作出更加多变的式样。