

XML 原理与应用

夏天

中国人民大学

CH5 利用 CSS 格式化 XML



本章学习目标

- 掌握 CSS 与 XML、HTML 的关联方式
- 掌握 CSS 的常见选择器，理解 CSS 的属性继承与覆盖原则
- 掌握 CSS 的颜色、字体和文本属性以及 CSS 的盒状模型

- 什么是 CSS
- 关联 CSS 的方法
- CSS 语法基础
 - CSS 基本语法
 - CSS 选择器
 - 继承与覆盖
- CSS 重要属性
 - 颜色属性
 - 字体属性
 - 文本属性
 - 盒状模型相关属性
 - 可视格式化模型相关属性

5.1 什么是 CSS

CSS (Cascading Style Sheets), 又称为层叠样式表或级联样式表, 可以实现对页面布局、字体、颜色、背景和其他图文效果的精准控制, 在 XML 和 HTML 的呈现中得到了广泛使用。

- CSS 的优点
- CSS 的发展历史
 - CSS1
 - CSS2, CSS 2.1
 - CSS 3

5.2 关联 CSS 的方法

- 与传统网页的关联方式
- 与 XML 的关联方式

5.2.1 CSS 与传统网页的关联方式

- 独立引用方式
 - 通过 HTML 中的 link 标记链接外部独立的样式表文件。
- 网页内嵌方式
 - 在 HTML 的 style 标记对中嵌入样式代码块。
- 元素内联方式
 - 在 HTML 的一般元素上采用 style 属性直接设置元素的样式信息

示例 I

Code 1: 5-1.html

```
1 <html>
2   <head>
3     <title>CSS DEMO</title>
4     <meta http-equiv="Content-Type"
5       content="text/html;charset=utf-8"/>
6     <link rel="stylesheet" type="text/css" href="5-1.css"/>
7     <style type="text/css">
8       h1 {font-size:14pt;font-family:"Microsoft Yahie" 微软雅黑;}
9     </style>
10  </head>
11  <body>
12    <h1>图书列表</h1>
13    <table>
14      <tr>
```


示例 II

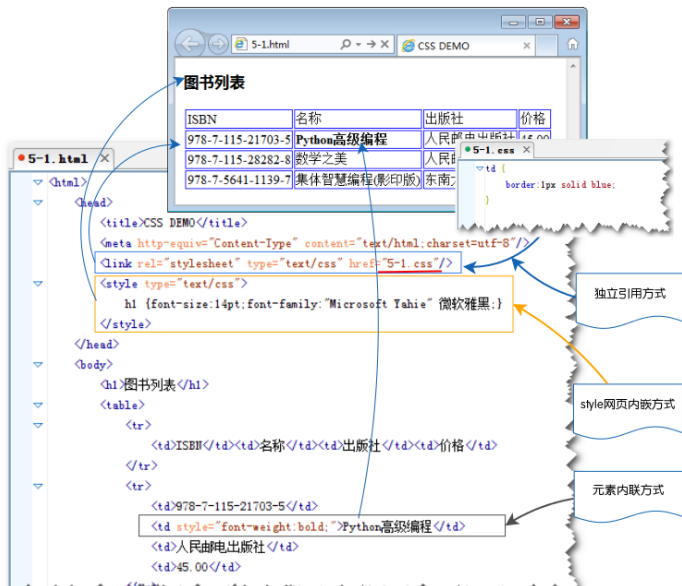
```
14      <td>ISBN</td><td>名称</td><td>出版社</td><td>价格</td>
15  </tr>
16  <tr>
17      <td>978-7-115-21703-5</td>
18      <td style="font-weight:bold;">Python高级编程</td>
19      <td>人民邮电出版社</td>
20      <td>45.00</td>
21  </tr>
22  <tr>
23      <td>978-7-115-28282-8</td>
24      <td>数学之美</td>
25      <td>人民邮电出版社</td>
26      <td>45.00</td>
27  </tr>
28  <tr>
29      <td>978-7-5641-1139-7</td>
```

示例 III

```
30      <td>集体智慧编程(影印版)</td>
31      <td>东南大学出版社</td>
32      <td>58.00</td>
33  </tr>
34 </table>
35 </body>
36 </html>
```

Code 2: 5-1.css

```
1 td {
2   border:1px solid blue;
3 }
```



5.2.2 CSS 与 XML 的关联方式

- 通过 `<?xml-stylesheet?>` 处理指令实现关联
- 非 W3C 规定，但得到主流浏览器的支持，成为事实标准

Code 3: books.css

```
1 book {display:block; margin:5px;}
2
3 title { width:200px;
4         border: 1px solid gray;
5         display:inline-block;
6         padding:2px;
7         padding-left:10px;
8     }
9
10 price { width:50px;
11         border: 1px solid gray;
12         display:inline-block;
13         padding:2px;
14     }
```

CSS 与 XML 关联示例 II

```
15
16 press { width:200px;
17         border: 1px solid gray;
18         display:inline-block;
19         padding:2px;
20 }
21
22 authors, pages, description, cover {display: none;}
```

Code 4: books.xml

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/css" href="books.css"?>
3 <books>
4   <book isbn="978-1449319793" id="b1">
5     <title lang="EN">Python for Data Analysis</title>
6     <price currency="dollar">25.40</price>
7     <authors>
8       <author>Wes McKinney</author>
9     </authors>
10    <press>OReilly Media</press>
11    <pages>470</pages>
12    <description>Python for Data Analysis...</description>
13    <cover>book-python.jpg</cover>
14  </book>
```

CSS 与 XML 关联示例 IV

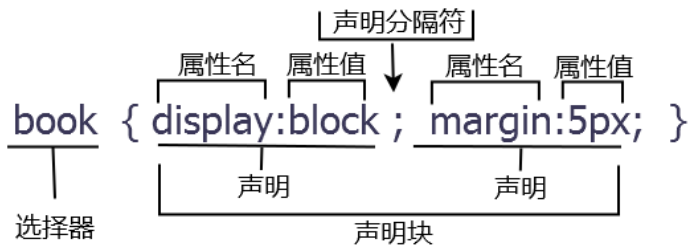
```
15 <book isbn="978-7-115-28282-8" id="b2">
16   <title lang="CHN">数学之美</title>
17   <price>45.00</price>
18   <authors>
19     <author>吴军</author>
20   </authors>
21   <press>人民邮电出版社</press>
22   <pages>304</pages>
23   <description>读了“数学之美”，才发现大学时...</description>
24   <cover>book-math.jpg</cover>
25 </book>
26 </books>
```


5.3 CSS 语法基础

- CSS 基本语法
- CSS 选择器
- 继承与覆盖

5.3.1 CSS 基本语法 I

- 选择器 + 规则声明
- 形式：选择符 {属性名: 属性值 [; 属性名: 属性值]...}



5.3.1 CSS 基本语法 II

- 选择符 (selector)
 - 被施加样式的元素标记名 (tag)、类 ([tag].class) 或标识符 ([tag]#id) 等
- 属性名称
 - 表示颜色、字体等需要指定样式的名称
- 属性值
 - 表示属性具体的取值，如颜色取 red，字体大小取 12pt 等
- 英文冒号和分号作分隔

5.3.2 CSS 选择器

- 元素选择器
- ID 选择器
- 属性选择器
- 类选择器
- 通用选择器
- 分组选择器
- 后代选择器

元素选择器

- 以元素名称作为选择器的表示方式，选择文档中的指定元素，是最常见的 CSS 选择方式
- 例子

```
title {color: blue;}
```

ID 选择器

- 选择文档中具有特定 id 的元素
 - 在通过在 id 值前面附加字符“#”作为选择器的表示方式
- 例子

```
#b1 {color: red;}
```

属性选择器

- 选择具有某个属性的元素集合

- 将具有属性名称为 lang 的 title 元素的字体颜色设为橙色：

```
title[lang] {color: orange;}
```

- 将元素名称更改为星号可选择具有特定属性名称的任意元素即可

```
*[isbn] {color: green;}
```

- 选择具有特定属性名称和属性值的元素集合

```
title[lang="CHN"] {color: red;}
```

类选择器

- 选择具有特性 class 属性的元素集合，应用于 HTML
- 例子

```
<h1 class="important">这是重要标题</h1>  
<p class="important">这是重要段落</p>
```

```
.important{color: red;}
```


通用选择器

- 星号“*”，匹配文档树中任何类型的元素名
- 例如：

```
* {color: blue;}
```

- 非单独出现的通用选择器可以省略

```
*[lang="CHN"] {color: red;} 等价于 [lang="CHN"] {color: red;}  
*#b1 {color: red;} 等价于 #b1 {color: red;}
```

分组选择器

- 一次选择多个元素
- 例子

```
title, press {color: red;}
```

后代选择器

- 把选中的元素限定在某些特定的文档结构中
- 例子

```
#b1 title {color: blue;}
```

- 将选中具有 ID 属性值为 b1 的元素内部的所有 title 子元素，并将其颜色设为蓝色
- 后代选择器可选中任意符合选择器要求的后代元素
- 子元素选择器
 - 限定子元素

```
book > title {color: blue;}
```

5.3.3 CSS 的继承与覆盖

- 属性继承
 - 子元素自动采用父元素中指定的样式作为本身的默认样式
- 属性覆盖
 - 一个元素最近指定的样式属性会覆盖之前指定的样式属性

支持继承的 CSS 属性

- color
- font 及相关属性
- letter-spacing
- line-height
- list-style 及相关的属性
- text 相关属性
- ...

不支持继承的 CSS 属性

- background 及相关属性
- border 及相关属性
- display
- float 和 clear
- margin 及相关属性
- ...

属性覆盖原则 -1

- 当因继承而发生样式冲突时，最近祖先的样式将覆盖更远祖先的样式

```
books {color:blue;}  
book {color:red;}
```

属性覆盖原则 -2

- 当直接指定的样式与继承样式冲突时，将采用直接指定的样式

```
books {color:blue;}  
book {color:red;}  
title {color:orange;}
```


属性覆盖原则 -3

- 当直接指定的样式发生冲突时，采用权重高者的样式

```
book {color:blue;}  
#b1 {color:red;}
```

- ID 选择器的权重高于元素选择器，因此，ID 为 b1 的图书内容将以红色字体显示

属性覆盖原则 -4

- 当样式权重相同时，采用最后指定的样式

```
book title[lang] {color: blue;}
```

```
book[isbn] title {color: red;}
```

属性覆盖原则 -5

- !important 的样式属性不被覆盖

```
book {color:blue !important;}  
#b1 {color:red;}
```

- 虽然 ID 选择器的权重高于元素选择器，但由于 book 元素选择器中的样式指定了 !important，从而使 ID 为 b1 的图书元素强制显示为蓝色

5.4 CSS 重要属性

- 颜色属性
- 字体属性
- 文本属性
- 盒状模型相关属性
- 可视格式化模型相关属性

5.4.1 颜色属性

- 名称表示方式

- black, white, blue, yellow, red, ...

- RGB 表示方式

- #rrggbb

- 两位十六进制数字 (00 FF) 分别表示红绿蓝三种颜色的分量取值, 如: #FF0000

- #rgb

- 等价于 #rrggbb , 如: #F00

- RGB(r, g, b)

- 以十进制指定红绿蓝颜色分量值, 如: RGB(255,0,0)

- RGB(r%, g%, b%)

- 用百分比表示红绿蓝分量比重, 如: RGB(100%, 0%, 0%)

颜色名称和 RGB 值



Black
#000000



Gray
#808080



Silver
#C0C0C0



White
#FFFFFF



Maroon
#800000



Red
#FF0000



Purple
#800080



Fuchsia
#FF00FF



Green
#008000



Lime
#00FF00



Olive
#808000



Yellow
#FFFF00



Navy
#000080



Blue
#0000FF



Teal
#008080



Aqua
#0000FF

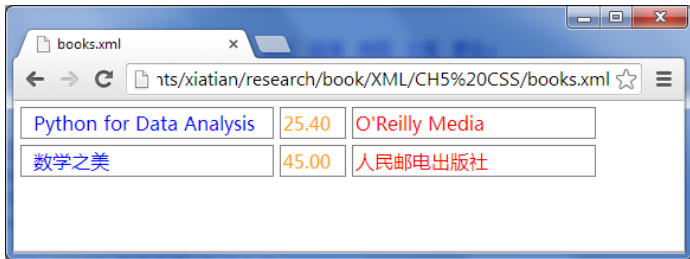


Orange (CSS 2.1)
#FFA500

颜色示例

```
1 book {display:block; margin:5px; }
2 title { width:200px; border: 1px solid gray;
3     display:inline-block;padding:2px;padding-left:10px;}
4 price { width:50px; border: 1px solid gray;
5     display:inline-block;padding:2px;}
6 press { width:200px; border: 1px solid gray;
7     display:inline-block;padding:2px;}
8 authors, pages, description, cover {display: none;}
9
10 title {color: blue;}
11 price {color: #FFA500;}
12 press {color:RGB(255, 0, 0);}
```

浏览器中的呈现效果



5.4.2 字体属性

- 字体族 font-family
- 字体风格 font-style
 - normal | italic | oblique | inherit
- 字体变体 font-variant
 - normal | small-caps | inherit
- 字体粗细 font-weight
 - normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit
- 字体大小 font-size

- 定义文本所使用的字体族
 - 字体族由一系列相关字体组成，如微软雅黑字体作为一个字体族会包括大小、粗体、斜体等不同显示效果的字体
- family-name (字体族名称)
 - 通常所说的“字体”，如“Arial”、“Times New Roman”、“宋体”、“微软雅黑”等
- generic family (族类名称)
 - 一组具有统一外观的字体族，它包含的范围比 family-name 更大，可看作是一组具有某些共同特点的 family-name 的集合

Generic Family: Serif、Sans-serif and Monospace

Times New Roman

Garamond

Georgia

这三个字体族属于serif族类。

它们的共同特点是，
笔画两端有“脚”。

Trebuchet

Arial

Verdana

这三个字体族属于sans-serif族类。

它们的共同特点是，
笔画两端没有“脚”。

Courier

Courier New

Andale Mono

这三个字体族属于monospace族类。

它们的共同特点是，
所有字符的宽度都一样。

- 绝对大小 <absolute-size>
 - xx-small | x-small | small | medium | large | x-large | xx-large
- 相对大小 <relative-size>
 - larger | smaller
- 指定长度 <length>
 - 相对长度单位：
 - em: the 'font-size' of the relevant font
 - ex: the 'x-height' of the relevant font
 - px: pixels, 像素, 相对于显示设备的最小显示单位
 - 绝对长度单位：
 - in: inches, 英寸, 一英寸约等于 2.54 厘米
 - cm: centimeters, 厘米

font-size II

- mm: millimeters , 毫米
- pt: pofints , 等于 1/72 英寸
- pc: picas , 等于 12 points
- 百分比 <percentage>
 - 相对于继承的字体大小的百分比

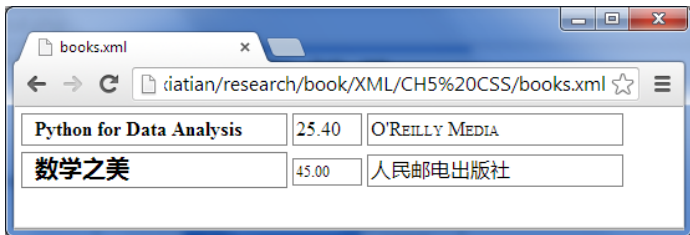
例子 I

Code 5: books-font.css

```
1 book {display:block; margin:5px; }
2 title { width:200px; border: 1px solid grey;
3     display:inline-block;padding:2px;padding-left:10px;
4 }
5 price { width:50px; border: 1px solid grey;
6     display:inline-block;padding:2px;
7 }
8 press { width:200px; border: 1px solid grey;
9     display:inline-block;padding:2px;
10 }
11 authors, pages, description, cover {display: none;}
12
13 book {font-family: "Times New Roman", "微软雅黑", Serif;}
14 title {font-weight:bold;}
```

例子 II

```
15 press {font-variant:small-caps;}  
16 book#b2 title {font-size: larger;}  
17 book#b2 price {font-size: 10pt;}
```



- 字体样式的简写方式

- 顺序：font-style | font-variant | font-weight | font-size | font-family

- 以下两段代码等价：

```
1 title {  
2   font-style: italic;  
3   font-weight: bold;  
4   font-size: 16px;  
5   font-family: arial, sans-serif;  
6 }
```



```
1 title {  
2   font: italic bold 16px arial,sans-serif;  
3 }
```

5.4.3 文本属性 I

- text-indent (文本缩进)
 - 设置文本块中首行文本的缩进或悬挂，缺省值为 0 (不缩进)

```
title { text-indent: 3em; }
```

- 例如将 title 元素内容缩进 3 个字
- text-align (文本对齐)
 - 设置文本段的对齐特性
 - 取值：
 - left (左对齐)
 - right (右对齐)
 - center (居中)
 - justify (两端撑满)

5.4.3 文本属性 II

- inherit (继承)
- text-decoration (文本修饰)
 - 取值可以为：
 - none (无修饰，为缺省值)
 - underline (下划线)
 - overline (上划线)
 - line-through (删除线)
 - blink (闪烁)
 - inherit (继承)
- letter-spacing (字母间隔) 和 word-spacing (词间隔)
 - 取值均可以为
 - normal (缺省值)

5.4.3 文本属性 III

- 指定长度
- inherit (继承)
- text-transform (文本转换)
 - 用于进行大小写转换
 - capitalize——每个单词的首字母大写，其他字母不变
 - uppercase——所有字母全大写
 - lowercase——所有字母全小写
 - none——不转换，使用默认值，为缺省值
 - inherit——继承
- white-space (空白)
 - 设置对文本中对空白符的处理方法
 - normal——收缩空白序列，自动换行，为缺省值

5.4.3 文本属性 IV

- pre——防止 (prevent) 空白序列收缩，保留空白符不变
- nowrap——收缩空白序列，但抑制换行
- pre-wrap——不收缩空白序列，在源文本的回车和换行处换行，并在需要的地方自动换行
- pre-line——收缩空白序列，在源文本的回车和换行处换行，并在需要的地方自动换行

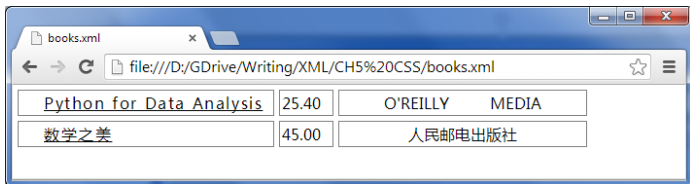
例子 I

Code 6: books-font.css

```
1 book {display:block; margin:5px; }
2 title { width:250px; border: 1px solid gray;
3         display:inline-block;padding:2px;padding-left:10px;}
4 price { width:50px; border: 1px solid gray;
5         display:inline-block;padding:2px;}
6 press { width:250px; border: 1px solid gray;
7         display:inline-block;padding:2px;}
8 authors, pages, description, cover {display: none;}
9
10 title {
11     text-indent:1em;
12     text-decoration:underline;
13     letter-spacing:0.1em;
14 }
```

例子 II

```
15 press {  
16     text-align:center;  
17     word-spacing:1cm;  
18     text-transform:uppercase;  
19 }
```



5.4.4 盒状模型相关属性



- 盒状模型

- margin
- padding
- border

- 简写方式：直接指定一个数值

```
title { padding:2px; }
```

- 按照上、右、下、左的顺时针方向顺序分别设置各边的内边距

```
title { padding: 2px 5% 5px 2%;}
```

- 分别设置
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left

margin

- 简写方式：直接指定一个数值

```
book { margin: 10px;}
```

- 按照上、右、下、左的顺时针方向顺序分别设置各边的外边距

```
book { margin: 5px 10px 5px 5%;}
```

- 分别设置
 - margin-top
 - margin-right
 - margin-bottom
 - margin-left

border

- border-width

```
title {border-width: medium;}  
price {border-width: 2px;}
```

- border-color

```
title {border-color: red;}
```

- border-style

```
book {border-style: solid;}
```

border 简写方式

- 以下两段代码等价

```
1 title {  
2   border-width: 2px;  
3   border-style: solid;  
4   border-color: red;  
5 }
```

```
title { border: 2px solid red;}
```

练习

- 通过 CSS 控制“books.xml”显示效果，在“books.css”的基础上，满足以下要求：
 - title、price 和 press 元素仅显示 2 个像素、灰色的实线下边框
 - 每行图书下面的外边距为 15 个像素
 - price 元素的左侧内边距为 20 个像素

5.4.5 可视格式化模型相关属性

- 可视格式化模型 VFM: Visual Formatting Model
 - 每个元素会依据盒状模型生成 0 个或多个盒子
 - 盒子的布局由盒的尺寸和类型、定位方案、文档树中元素间的关系、以及外部信息所控制
 - 包括显示、高度、宽度、定位、文本方向等属性

display 显示属性

- display 显示属性
 - 用于设置和改变元素所对应的盒子的显示类型
 - 元素既可以显示为独立的一块内容，即块元素
 - 也可以显示在一行中，即行内元素

display 属性常见取值及作用

display 属性取值	作用
block	使元素产生一个块状盒子，并换行
inline-block	产生一个行内 (inline) 块状盒子，不强制换行
inline	产生一个或多个行内盒子，缺省值
list-item	产生一个主块和一个表项行内盒
none	不产生盒子，即无布局效果

宽度和高度属性

- 宽度属性 width
 - 指定由块级元素和替换元素所生成的盒子的内容宽度
 - 取值可以为长度或百分比
- 高度属性 height
 - 用于对元素盒子的高度进行限定，
 - 取值同样为长度值或百分比
- 例子

```
title{height:30px; width:500px;}
```

- 指定 title 元素的高度和宽度分别为 30 个像素和 500 个像素大小

END

