

Web应用基础

Lab 3: CSS – 层叠样式表

2009.3.6

准备知识

1 对文档应用样式

样式可以通过多种方式指定。样式可以位于HTML的<head>标签中，也可以存放在外部的CSS文件中。一个HTML可以链接多个外部的样式表。

当对于一个HTML元素通过多种方式指定了多个重复样式的时候，最终那一种会生效呢？例如我们在多个样式表中对<body>元素定义了不同背景色，那么最终呈现出的颜色应该是那一种呢？

通常来讲，我们可以说样式通过下面的规则“层叠”为一个新的虚拟的样式表。为HTML元素指定样式的方法共有4中，分别是：

1. 外部样式：在<head>元素中的<link>元素中指定的CSS文件
2. 外部样式：在<head>元素中的<style>元素导入指定的CSS文件
3. 内部样式：在<head>元素中的<style>元素中编写CSS代码
4. 内联样式：在元素的style属性中编写CSS代码

上面4种方式优先级如下：内联样式(4.)优先级最高；<link>样式(1.)和<style>样式(2., 3.)按照文档中的排列顺序，越后优先级越高；<style>内，内部样式(3.)优先级高于外部样式(2.)。如果以同种方式引入了多个样式，则列在后面的样式优先级较高。如果多个样式表定义的样式没有冲突，那么这些样式将全部得以保留。

上面所述的应用样式的4种方法具体方法如下：

1. 链接外部样式表：

```
...
<head>
  ...
  <link rel="stylesheet" href="style.css" type="text/css" />
  ...
</head>
...
```

2. 使用内部样式表:

```
...
<head>
  ...
  <style type="text/css">
  <!--
  body { background-color: red; }
  ...
  -->
  </style>
  ...
</head>
...
```

3. 导入外部样式表:

```
...
<head>
  ...
  <style type="text/css">
  <!--
  @import url("style.css")
  ...
  -->
  </style>
  ...
</head>
...
```

4. 使用内联样式:

```
...
<div style="color: red; margin-left: 10px">...</div>
...
```

TIPS

为什么在<style>元素中定义样式或引入样式时需要将其标注为HTML注释(写在"

CSS的基本语法为：

```
selector { property: value; }
```

通过选择器定位元素，为其应用{}内指定的样式，样式由若干条属性-值对组成，用分号隔开，最后一条样式分号可选。

1. 元素选择器

类选择器用来查找特定类型的元素，比如段落(<p>)、标题(<h1>, <h2>, ...)或锚(<a>)元素。语法如下：

```
element { property: value; }
```

例如：

```
p { text-align: center; font-weight: bold; }
```

这个样式将文档中所有<p>元素设为文字居中对齐并为粗体。

2. 类选择器

类选择器用于寻找有指定类名的元素。在类名前加"."即可。语法如下：

```
.class_name { property: value; }
```

例如：

```
.right { text-align: right; }
```

这个样式将具有属性class="right"的元素的文字对齐方式设为右对齐。

3. ID选择器

类选择器用于寻找有指定ID的元素。在类名前加"#"即可。语法如下：

```
#id { property: value; }
```

例如：

```
#intro { font-size: 20px; }
```

这个样式将具有属性id="intro"的元素的文字大小设为20px。

4. 后代选择器

将不同选择器使用空格连接，就构成了后代选择器。例如，

```
p #intro { font-size: 20px; }
```

指定的元素为任何<p>元素内id为intro的元素。

```
.main li a { font-weight: bold; text-decoration: none; }
```

指定的元素为class为main的元素内所有元素内的<a>元素。

5. 伪类选择器

有时候，希望根据文档结构之外的条件对元素应用样式，如表单元素或者链接的状态。这就需要通过伪类选择器来完成。

```
/* 将未访问的链接设为蓝色 */
a:link { color: blue; }

/* 将已访问的链接设为绿色 */
a:visited { color: green; }
/* 将鼠标移到链接上时以及点击链接时的颜色设为红色 */
a:link { color: red; }

/* 将表格的行在鼠标移入时的背景颜色设为红色 */
tr:hover { background-color: red; }

/* 当激活input元素时将其背景颜色设为黄色 */
input:focus { background-color: yellow; }
```

`:link`和`:visited`称为链接伪类，只能应用于锚元素(`<a>`)。`:hover`、`:active`和`:focus`称为动态伪类，理论上可应用于任何元素。但在一些包括IE6在内的老式浏览器只注意应用于锚链接的`:active`和`:hover`选择器，完全忽略`:focus`。CSS 2.1版本不仅有上述伪类，更多的伪类(但很少使用)定义可以参考W3C官方的规范。

6. 高级选择器(选看)

CSS 2.1定义了许多其他有用的选择器。可是包括IE 6在内的一些老式浏览器并不支持这些高级选择器。不过如果使用了老浏览器所不支持的选择器，那么相应的规则会被忽略，并不影响其他规则。由于考虑到IE 6的市场占有率，采用高级选择器就意味着忽视掉大量的访客。所以，**在对于站点功能很重要的任何元素上应该避免使用这些高级选择器。**

◆ 子选择器和相邻同胞选择器

后代选择器选择第一个元素的所有后代，而子选择器使用"`>`"来只选择元素的直接后代。语法如下：

```
parent > child { property: value; }
```

例如对于HTML结构

```
<ul id="nav">
  <li>Home</li>
  <li>Services
    <ul>
      <li>Design</li>
      ...
    </ul>
```

```
</li>
...
</ul>
```

如下的样式

```
#nav > li { font-family: Tahoma; }
```

仅选择了id为nav的下的一级, 而不影响嵌套在这些内部的元素。而相邻同胞选择器可用于定位同一个父元素下某个元素之后的元素。语法如下:

```
element + sibling { property: value; }
```

例如对于HTML结构

```
<h1>Top News</h1>
<p>First Paragraph</p>
<p>Second Paragraph</p>
```

如下的样式

```
h1 + p { font-weight: bold; }
```

仅选择了紧跟<h1>元素后的一个<p>元素而不影响之后的段落。

◆ 属性选择器

属性选择器可以根据元素的某个属性是否存在或属性的值来寻找元素。属性的条件写在一对中括号内。主要有三种属性选择方式:

简单属性选择器

```
[attribute] { property: value; }
```

选择定义了某属性(无论属性值是什么)的元素。例如

```
h1[title] { border-bottom: 1px dotted gray; }
```

选择了所有带有title属性的<h1>元素。

精确属性值选择器

```
[attribute="value"] { property: value; }
```

选择定义了某属性且其属性值为指定值的元素。例如

```
a[rel="nofollow"] { color: gray; }
```

选择了所有存在属性rel值为"nofollow"的<a>元素, 如。

部分属性值选择器

```
[attribute~="value"] { property: value; }
```

选择的元素定义了某属性, 且将这个属性值按空格进行分隔后其中有一段恰为指定值。例如

```
a[rel~="friend"] { color: red; }
```

选择了所有存在属性rel, 且将rel属性值按空格分隔后其中恰有一段为"friend"的元素, 如, 而不匹配。

特殊属性选择器

```
[attribute/="value"] { property: value; }
```

选择的元素定义了某属性, 且将这个属性值按"-"进行分隔后其中第一段恰为指定值。例如

```
p[lang|=“en”] { font-family: Verdana; }
```

选择了所有存在属性rel, 且将lang属性值按“-”分隔后其中第一段恰为“en”的元素, 如<p lang=“en-us”>。事实上这个选择器主要就是针对语言属性lang而设计的。

TIPS

如果在某个选择器后直接接上一个或多个类选择器、ID选择器或属性选择器(没有空格), 则表示被选择的元素需要同时满足这几个选择器的条件。例如

“p.right”对应了具有属性class=“right”的<p>元素;

“.main[title]”对应了具有属性class=“main”定义了属性title的元素;

“.bold.right”对应了具有属性class=“bold right”的元素。

对于多条样式, 一般将其写为多行, 以便阅读:

```
p.main {  
  text-align: center;  
  font-weight: bold;  
}
```

对于应用于多个元素的样式中重复的部分, 可将同属性值的选择器组合起来用逗号隔开:

```
h1, h2, h3, h4, h5, h6 { font-family: arial; }
```

对于带单位的属性值, 在数值和单位之间, 不要留有空格。如设定高度为200px时, 应该写为“height: 200px”而不要写成“height: 200 px”。

3 特殊性

即使在不太复杂的样式表中, 也有可能有两个或更多规则寻找到同一元素。CSS通过一个称谓层叠(cascade)的过程处理这种冲突。层叠过程根据选择器的特殊性决定规则的次序。具有更特殊选择器的规则优先于具有比较一般的选择器的规则。如果两个规则的特殊性相同, 那么后定义的规则优先。

为了计算规则的特殊性(specificity), 每种选择器被分配一个数字值。然后, 将规则的每个选择器的值加在一起, 计算出规则的特殊性。不幸的是, 特殊性的计算不是以10为基数的, 而是采用一个更高的未指定的基数。这确保非常特殊的选择器(比如ID选择器)不会被大量比较一般的选择器(如元素选择器)的组合所超越。但是为了简化, 如果在一个特定的选择其中的选择器数量少于10个, 那么可以以10为基数计算特殊性。

选择器的特殊性分成四个成分等级: a、b、c和d。

- ◆ 如果样式是行内样式, 那么a=1。
- ◆ b=ID选择器的总数。
- ◆ c=类、伪类和属性选择器的总数。
- ◆ d=元素选择器的总数。

使用这些规则可以计算任何CSS选择器的特殊性。表3-1给出一系列选择器以及相应的特殊性。

表3-1 特殊性示例

选择器	特殊性(a,b,c,d)	以10为基数的特殊性
style=""	1,0,0,0	1000
#wrapper #content {}	0,2,0,0	200
#content .datePosted {}	0,1,1,0	110
div#content {}	0,1,0,1	101
#content	0,1,0,0	100
p.comment .dateposted {}	0,0,2,1	21
p.comment {}	0,0,1,1	11
div p {}	0,0,0,2	2
p {}	0,0,0,1	1

基本上，用style属性编写的规则总是比其他任何规则特殊(优先)。具有ID选择器的规则比没有ID选择器的规则特殊，具有类选择器的规则比只有类型选择器的规则特殊。最后，如果两个规则的特殊性相同，那么后定义的规则优先。

4 CSS样式

可参考课程PPT或者[这里](#)。

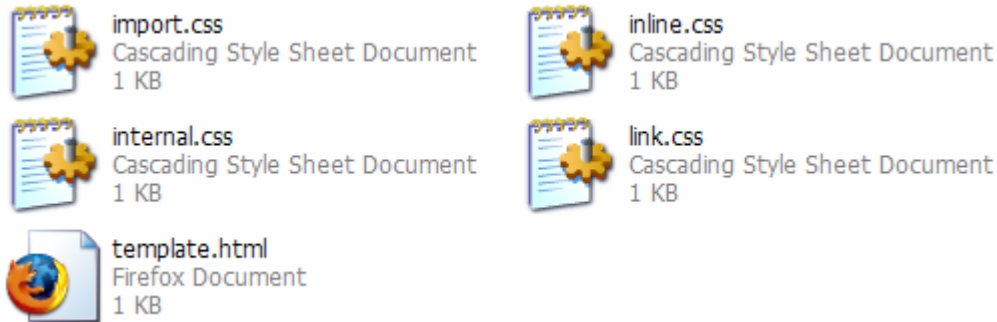
实验一 CSS和HTML的结合方式

实验内容

使用文本编辑器(如记事本、Editplus等，不允许使用Dreamweaver等开发环境)练习CSS与HTML结合的四种方式。

实验任务

task_1文件夹下有如下5个文件：



利用在HTML中应用样式表的四种方式，仅修改template.html，对template.html中的4个<div>分别应用样式。其中，

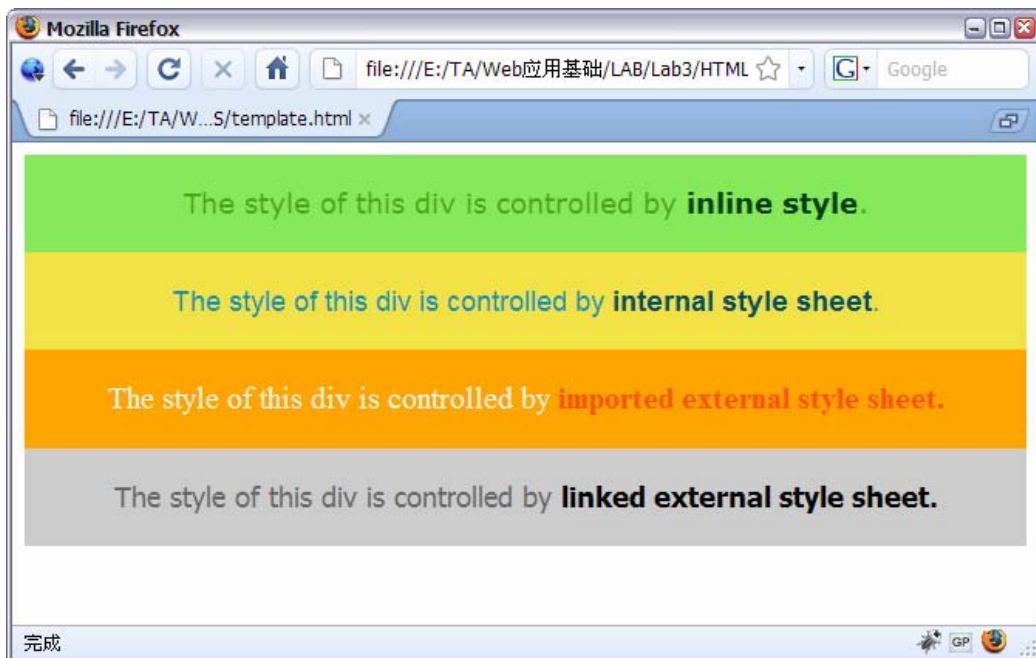
对class="inline"的<div>以内联方式应用inline.css中的样式；

对class="internal"的<div>以内部样式表的方式应用internal.css中的样式；

对class="link"的<div>以链接外部样式表的方式链接link.css中的样式；

对class="import"的<div>以引入外部样式表的方式引入import.css中的内容。

最终完成后的效果如下：



实验提交物

task_1文件夹，包括template.html、import.css以及link.css。

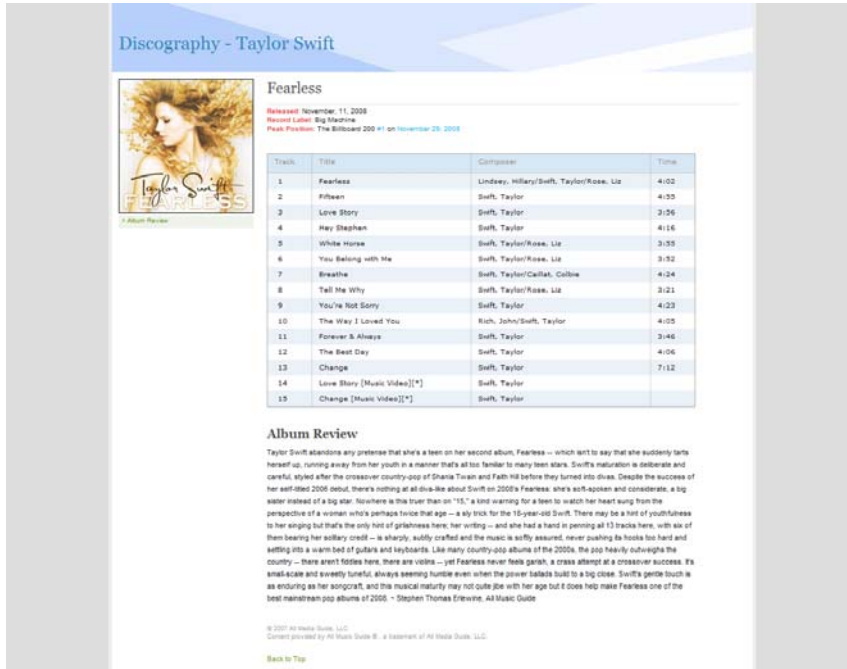
实验二 CSS和HTML的结合方式

实验内容

使用文本编辑器(同实验一)练习利用CSS对HTML页面的样式进行控制。

实验任务

对文件夹task_2下的文件discography.html编写CSS样式，使其尽量接近下图中给出的效果。这是某张唱片的简介页面。(全图见task_2/final.jpg)



顶部banner使用的背景图片为task_2/images/bg.png。不要求完全一样，接近即可。必须做到的有：页面设置底色；顶部应用背景图案；两处锚链接应用伪类选择器使其在4种情况下显示的色彩不同；效果图中颜色与正文不同的文字必须设置颜色，可以自选；对表格设置合适的文字大小和尺寸，并使表头不同于其他行，且为单数行的单元格设置底色。总体的要求为尽量做到页面效果美观；代码规范，在HTML代码中只允许为元素加入id和class属性，样式表代码写在单独的文件中并用link或import的方法应用在HTML上；各个元素的定位尽量接近效果图。

实验提交物

task_2文件夹，包括discography.html、style.css(名字可以自己起)以及images文件夹(不需要进行改动)。

TIPS

在线取色器可尝试[这里](#)。

使用margin和padding控制元素在父元素中的位置以及元素间的间隔。

参考文字大小：顶部标题27px、右侧部分顶端22px、唱片信息与表格文字10px、评论标题18px、评论正文11px。标题字体为Georgia，评论正文为Arial，表格内容为Verdana。

实验三 CSS特殊性的计算

实验内容

计算CSS特殊性(优先级)。

实验任务

对于下列给出的情况计算相应的特殊性，结果为(a,b,c,d)的形式。

1. `div#banner {}`
2. `p.comment .reply {}`
3. `.comment > a[rel~="friend"].backtrack {}`
4. `#news2314 .header + h1 {}`
5. `div ul li a:hover {}`
6. `#main_content .clear {}`
7. `table.tracklist tr:hover {}`
8. `style=""`
9. `form#search input:focus {}`
10. `#content p:first-letter {}`

要求将答案以如下形式写入文本文件

0,2,1,0

1,0,0,0

0,1,0,1

...

不需要写题号。

实验提交物

task_3.txt。