



CSS 基础

主讲人：和凌志

CSS层叠样式表

- CSS的全称是Cascading Style Sheet，中文翻译为“层叠样式表”。
- 实用CSS可以改变HTML的基本特性，从而控制传统网页上的元素，诸如精确的留白、段落缩进、文本的外观和链接的状态等等。这样，就可以实现对网页元素的格式控制。
- CSS不仅允许指定单个网页的外观，还可以被用来为网站提供一致的界面。

CSS 应用

- CSS样式通常保存在外部的 `.css` 文件中。通过仅仅编辑一个简单的 CSS 文档，外部样式表使你有能力同时改变站点中所有页面的布局和外观。这样，创建的文档内容和文档的显示外观就有着明显的相互独立。
- 当然，CSS样式也可以以 `<style>` 标签的形式写在网页文件的 `<head>` 标签中，或是直接将CSS语句书写在某个元素标签的属性部分。

CSS 优先级

疑问：

当同一个 HTML 元素被不止一个样式定义时，会使用哪个样式呢？

一般而言，所有的样式会根据下面的规则层叠于一个新的虚拟样式表中，它们产生作用的优先级别从高到低：

1. 内联样式（在 HTML 元素内部）
2. 内部样式表（位于 <head> 标签内部）
3. 外部样式表（以CSS文件独立存在）

遵循 “就近原则”

CSS 语法

CSS 语法由三部分构成：选择器、属性和值：

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

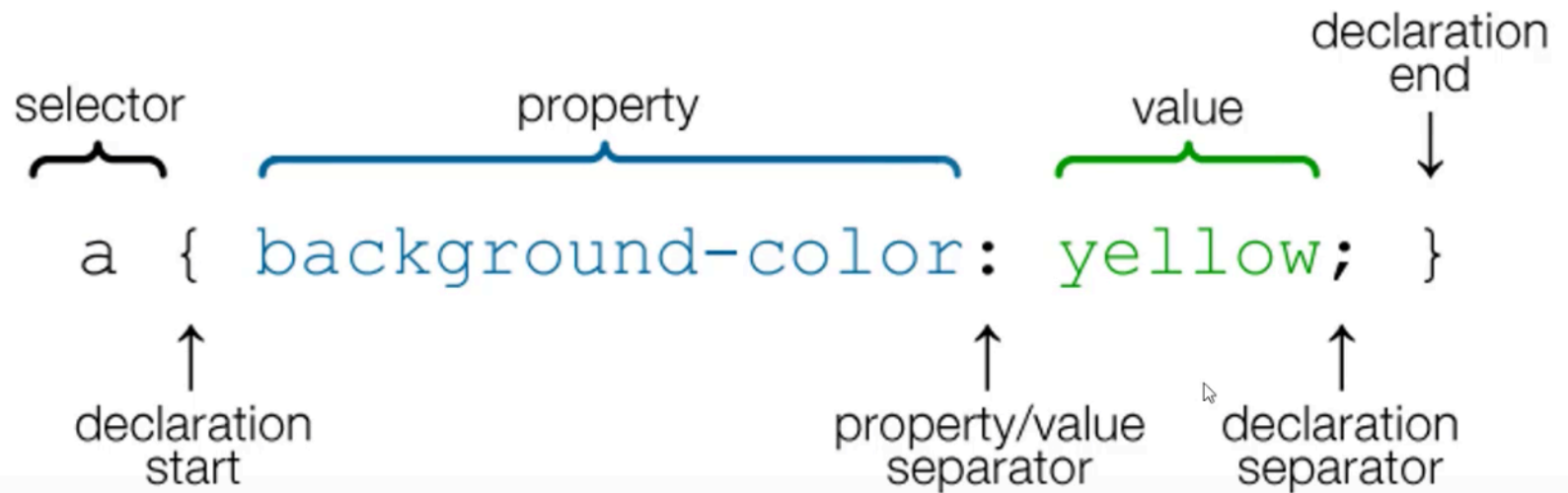
选择器 (selector) 通常是你希望定义的HTML元素或标签，属性 (property)是你希望改变的属性，并且每个属性都有一个值。属性和值被冒号分开，并由花括号包围，这样就组成了一个完整的样式声明 (declaration)，例如：

```
body {color: blue}
```

如果值为若干单词，则要给值加引号：

```
p {font-family: "sans serif";}
```


CSS *Selectors*



定制化背景颜色和字体颜色

```
body {  
    background-color : ■ #f4f4f4;  
    color: ■ #555555  
}
```

Hello World



如果要定义不止一个声明，则需要用分号将每个声明分开。最后一条规则是不需要加分号的，但是一般在每条声明的末尾都会加上分号，包括最后一条规则。这么做的好处是尽可能的减少出错的可能性。

```
p {text-align:center; color:red;}
```

在书写的时候可以每行只描述一个属性，以便使样式定义的可读性更强。

```
p {  
    text-align: center;  
    color: black;  
    font-family: arial;  
}
```


选择器的分组

可以对选择器进行分组，这样，被分组的选择器就可以分享相同的声明，用逗号将需要分组的选择器分开。

```
h1, h2, h2, h3, h5, h6 {  
    color: green;  
}
```

在上面的例子中，对所有的标题标签进行了分组。所有的标题元素都是绿色的。

选择器的类型

■ 派生选择器

通过依据元素在其位置的上下文关系来定义样式，在CSS中，通过这种方式来应用规则的选择器被称为派生选择器。这样可以利用元素的上下文关系来应用或者避免某项规则。

```
li b {  
    font-style: italic;  
    font-weight: normal;  
}
```

■ 类选择器

在 CSS 中，类选择器以一个点号显示：

```
.center {text-align: center}
```

因此，所有拥有 center 类的HTML元素均为居中。

```
<h1 class="center"> This heading will be center-  
aligned </h1>
```

```
<p class="center"> This paragraph will also be  
center-aligned. </p>
```

■ id选择器

id 选择器可以为标有特定id的HTML元素指定特定的样式。id 选择器以 “#” 来定义。下面的两个id选择器，第一个可以定义元素的颜色为红色，第二个定义元素的颜色为绿色：

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

```
<p id="red">这个段落是红色。</p>  
<font id="green">这句文本是绿色。</font>
```

注意：一个值的id属性只能在每个HTML文档中出现一次。

id选择器和派生选择器

id选择器常用于建立派生选择器。

```
#sidebar p {  
    font-style: italic;  
    text-align: right;  
    margin-top: 10px;  
}
```

上面的样式只会应用于出现在id是sidebar的元素内的段落。这个元素很可能是 div 或者是表格单元，即使被标注为sidebar的元素只能在文档中出现一次，这个id选择器作为派生选择器也可以被使用很多次。

和 id 一样，**class** 也可被用作派生选择器：

```
.fancy td { color: #f60; background: #666; }
```

类名为fancy的元素内部的表格单元都会以灰色背景显示橙色文字。

在上面的例子中，类名为fancy的表格单元将是带有灰色背景的橙色。你可以将类fancy分配给任何一个表格元素任意多的次数。那些以 fancy 标注的单元格都会是带有灰色背景的橙色。那些没有被分配名为fancy的类的单元格不会受这条规则的影响。

class 选择器与 **id** 选择器，哪个更好？

Bootstrap 所用的全部是 **class** 选择器

如何创建 CSS

创建CSS样式表的方法有三种：

1 外部样式表

当样式需要应用于很多页面时，外部样式表将是理想的选择。Web 页面使用 `<link>` 标签链接到样式表文件。`<link>` 标签写在文档的头部（`head`）：

```
<head>
```

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

```
</head>
```

提示： 外部文件引用时， 要注意所引用文件的路径

外部样式表可以在任何文本编辑器进行编辑。文件不能包含任何的 html 标签。样式表应该以 .css 扩展名进行保存。

```
hr {  
    color: sienna  
}  
p {  
    margin-left: 20px  
}  
body {  
    background-image: url("images/back40.gif")  
}
```

2 内部样式表

当单个文档需要特殊的样式时，就应该使用内部样式表。

内部样式表是使用 `<style>` 标签写在头部：

```
<head>
<style type="text/css">
    hr {color: sienna}
    p {margin-left: 20px}
    body {background-image:url("images/back40.gif")}
</style>
</head>
```

3 内联样式表

当样式仅需要在一个元素上应用一次时，就需要内联样式表。要使用内联样式，你需要在相关的标签内使用样式（style）属性，Style属性可以包含任何CSS属性。

```
<p style="color: sienna; margin-left: 20px">  
    This is a paragraph  
</p>
```

CSS属性

CSS能够处理的属性有：

背景、文本、字体、边框、内边距、外边距、列表、表格、尺寸、定位、分类、伪类

CSS 背景 (background)

该属性定义元素的背景效果。元素的背景区包括前景之下直到边框边界的所有空间。因此，内容框与内边距都是元素背景的一部分，且边框画在背景上。

CSS 允许应用纯色作为背景，也允许使用背景图像创建相当复杂的效果。

Background 简写属性

background-attachment 背景图像是否固定或者随着页面的其余部分滚动。

background-color 设置元素的背景颜色。

background-image 把图像设置为背景。

background-position 设置背景图像的起始位置。

background-repeat 设置背景图像是否及如何重复。

实例：

背景色为红色

```
<body>
```

```
  <p style="background-color: red"> 背景色为红色 </p>
```

```
</body>
```

```
<body>
```

```
|   <p style="background-color:  red"> 背景色为红色 </p>
```

```
</body>
```

CSS 文本

使用文本属性，你可以改变文本的颜色，增加或减少文本中的字符间距，对齐文本，装饰文本，对文本中的首行进行缩进等。

Color	设置文本颜色
Direction	设置文本方向。
line-height	设置行高。
letter-spacing	设置字符间距。
text-align	对齐元素中的文本。
text-decoration	向文本添加修饰。
text-indent	缩进元素中文本的首行。
text-transform	控制元素中的字母。
word-spacing	设置字间距。

CSS 字体 (font)

CSS 字体属性允许您设置字体系列 (font-family) 和字体加粗 (font-weight), 您还可以设置字体的大小、字体风格 (如斜体) 和字体变形 (如小型大写字母)。

Font	简写属性
font-family	设置字体系列。
font-size	设置字体的尺寸。
font-style	设置字体风格。
font-variant	以小型小写字体或者正常字体显示文本。
font-weight	设置字体的粗细。

CSS 边框(border)

CSS边框属性可以规定元素边框的粗细、样式和颜色。

border	简写属性
border-style	用于设置元素所有边框的样式，或者单独地为各边设置边框样式。
border-width	用于为元素的所有边框设置宽度，或者单独地为各边边框设置宽度。
border-color	设置元素的所有边框中可见部分的颜色，或为 4 个边分别设置颜色。

CSS 外边距(margin)

CSS外边距属性定义元素周围的空间。设置外边距会在元素外创建额外的“空白”。“空白”通常指不能放其它元素的区域，而且在这个区域中可以看到父元素的背景。

```
h1 {margin : 10px 0px 15px 5px;}
```

这些值的顺序是从上外边距（top）开始围着元素顺时针旋转的：

margin: top right bottom left

margin	简写属性
margin-bottom	设置元素的下外边距。
margin-left	设置元素的左外边距。
margin-right	设置元素的右外边距。
margin-top	设置元素的上外边距。

CSS 内边距(padding)

CSS 内边距属性定义元素边框与元素内容之间的空白。

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

Padding	简写属性。
padding-bottom	设置元素的下内边距。
padding-left	设置元素的左内边距。
padding-right	设置元素的右内边距。
padding-top	设置元素的上内边距。

CSS 列表

CSS 列表属性可以放置、改变列表项标志，或者将图像作为列表项标志。

<code>list-style</code>	简写属性。
<code>list-style-image</code>	将图象设置为列表项标志。
<code>list-style-position</code>	设置列表中列表项标志的位置。
<code>list-style-type</code>	设置列表项标志的类型。

CSS 表格

CSS 表格属性允许你设置表格的布局。

<code>border-collapse</code>	设置是否把表格边框合并为单一的边框。
<code>border-spacing</code>	设置分隔单元格边框的距离。（仅用于“separated borders”模型）
<code>empty-cells</code>	设置是否显示表格中的空单元格。（仅用于“separated borders”模型）
<code>caption-side</code>	设置表格标题的位置。
<code>table-layout</code>	设置显示单元、行和列的算法。

CSS 尺寸(Dimension)

CSS 尺寸(Dimension)属性可以控制元素的高度和宽度及行间距。

Height	设置元素的高度。
line-height	设置行高。
max-height	设置元素的最大高度。
max-width	设置元素的最大宽度。
min-height	设置元素的最小高度。
min-width	设置元素的最小宽度。
Width	设置元素的宽度。

CSS 分类(Classification)

CSS 分类属性可以规定如何以及在何处显示元素。

Clear	设置一个元素的侧面是否允许其他的浮动元素。
Cursor	规定当指向某元素之上时显示的指针类型。
Display	设置是否及如何显示元素。
Float	定义元素在哪个方向浮动。
Position	把元素放置到一个静态的、相对的、绝对的、或固定的位置中。
Visibility	设置元素是否可见或不可见。

Display 的属性值

None	此元素不会被显示。
Block	此元素将显示为块级元素，此元素前后会带有换行符。
Inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素。

浮动元素：出现在其他元素中的图形和文本元素称为浮动元素（floating element）。

clear 的属性值

Left	在左侧不允许浮动元素
Right	在右侧不允许浮动元素
Both	在左右两侧均不允许浮动元素
None	默认。允许浮动元素出现在两侧。

Float 的属性值

left	文本或图像会移至父元素中的左侧。
right	文本或图像会移至父元素中的右侧。
none	默认。文本或图像会显示于它在文档中出现的位置。

例题 css-15.html、 css-16.html、 css-17.html

visibility 的属性值

visibility 属性设置元素是否可见或不可见。即使不可见的元素也会占据页面上的空间。如果不可见元素不占据页面空间则用display来控制。

Visible	元素是可见的。
Hidden	元素是不可见的。

CSS 定位(Positioning)

Bottom	定义了定位元素下外边距边界与其包含块下边界之间的偏移。
clip	设置元素的形状。元素被剪入这个形状之中，然后显示出来。
left	定义了定位元素左外边距边界与其包含块左边界之间的偏移。
Overflow	设置当元素的内容溢出其区域时发生的事情。
Position	把元素放置到一个静态的、相对的、绝对的、或固定的位置中。
Right	定义了定位元素右外边距边界与其包含块右边界之间的偏移。
Top	定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移。
vertical-align	设置元素的垂直对齐方式。
z-index	设置元素的堆叠顺序。

CSS 伪类(Pseudo-classes)

CSS 伪类用于向某些选择器添加特殊的效果。

伪类的语法：

```
selector:pseudo-class {property: value}
```

```
a:link {color: #FF0000}
```

```
a:visited {color: #00FF00}
```

```
a:hover {color: #FF00FF}
```

```
a:active {color: #0000FF}
```

提示：在 CSS 定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。a:active 必须被置于 a:hover 之后，才是有效的。

CSS 伪元素(Pseudo-elements)

CSS 伪元素用于将特殊的效果添加到某些选择器。伪元素的语法：

选择器 : 伪元素 { 属性: 值 }

:first-letter	将特殊的样式添加到文本的首字母
:first-line	将特殊的样式添加到文本的首行

编码规范

Html 不区分大小写

在css中:

- id名和class名是区分大小写的
- 选择器中，只有标签选择器不区分大小写
- 属性名和属性值是不区分大小写的

约定：一律小写，中间用短线 -

Q & A