

Web开发一

第四章 CSS语法基础

本讲目标



- ▶ 理解CSS的作用及优势
- ▶ 理解内容和样式分离思想
- ▶ 掌握CSS基本语法
- ▶ 掌握CSS基本应用实例

本节内容



CSS简介



CSS基础语法



CSS基础小结

● CSS概述

- **CSS**是 **Cascading Style Sheet** 的缩写。译作「层叠样式表」。
- CSS是用于(增强)控制网页样式并允许将**样式**与网页**内容分离**的一种标记性语言。

A blue square icon with a white border and a white triangle pointing upwards, containing the text "CSS" in white.

CSS

A blue square icon with a white border and a white triangle pointing upwards, containing the text "HTML" in white.

HTML

为什么使用CSS

```
<ul>
  <li><a href="#"><font color="red" size="4" face="微软雅黑">第一条国际新闻</font></a></li>
  <li><a href="#"><font color="red" size="4" face="微软雅黑">第二条国际新闻</font></a></li>
  <li><a href="#"><font color="red" size="4" face="微软雅黑">第三条国际新闻</font></a></li>
  <li><a href="#"><font color="red" size="4" face="微软雅黑">第四条国际新闻</font></a></li>
  <li><a href="#"><font color="red" size="4" face="微软雅黑">第五条国际新闻</font></a></li>
</ul>
```

- [第一条国际新闻](#)
- [第二条国际新闻](#)
- [第三条国际新闻](#)
- [第四条国际新闻](#)
- [第五条国际新闻](#)

- [第一条国际新闻](#)
- [第二条国际新闻](#)
- [第三条国际新闻](#)
- [第四条国际新闻](#)
- [第五条国际新闻](#)

CSS

```
<ul>
  <li><a href="#">第一条国际新闻</a></li>
  <li><a href="#">第二条国际新闻</a></li>
  <li><a href="#">第三条国际新闻</a></li>
  <li><a href="#">第四条国际新闻</a></li>
  <li><a href="#">第五条国际新闻</a></li>
</ul>
```

```
+ ul li a{
  color:red;
  font-size:16px;
  font-family:"微软雅黑";
}
```

▶ 为什么使用CSS

- 单纯HTML方法控制元素样式的**缺点**：

- 针对元素样式设置的属性太**少**
- 修改元素样式太**麻烦**
- 控制元素样式的代码**冗余**度过高

▶ 更专业的样式
修饰方法

▶ 改一处动全局
的方法

▶ 更简便的样式
修饰方法

▶ 更简约的布局
方法



本节内容



CSS简介



CSS基础语法



CSS基础小结

▶ CSS语法

```
a          { color:#2d374b;text-decoration:none          }
a:hover    { color:#cd0200;text-decoration:underline     }
em         { font-style:normal                          }
li         { list-style:none                             }
img        { border:0;vertical-align:middle              }
table      { border-collapse:collapse;border-spacing:0    }
.undis     { display:none                                 }
.dis       { display:block                                }
.mt8       { margin-top:8px                               }
.mt12      { margin-top:12px                              }
.mt15      { margin-top:15px                              }
.mb10      { margin-bottom:10px                           }
#mymsg a   { color:#2D374B;                               }
#myz a     { color:#2D374B;                               }
#myd a     { color:#2D374B;                               }
#myp a     { color:#2D374B;                               }
#myp a     { color:#0d519c;                               }
```

CSS代码是由一条条语句构成，而每一条语句的结构，都基本相同

选择器 { 属性：属性值；属性：属性值；…… }

● CSS语法：

选择器

{

属性：属性值；属性：属性值；.....

}

1. 选择器

2. 大括号

3. 属性：属性取值（属性之间用；分隔）

p{

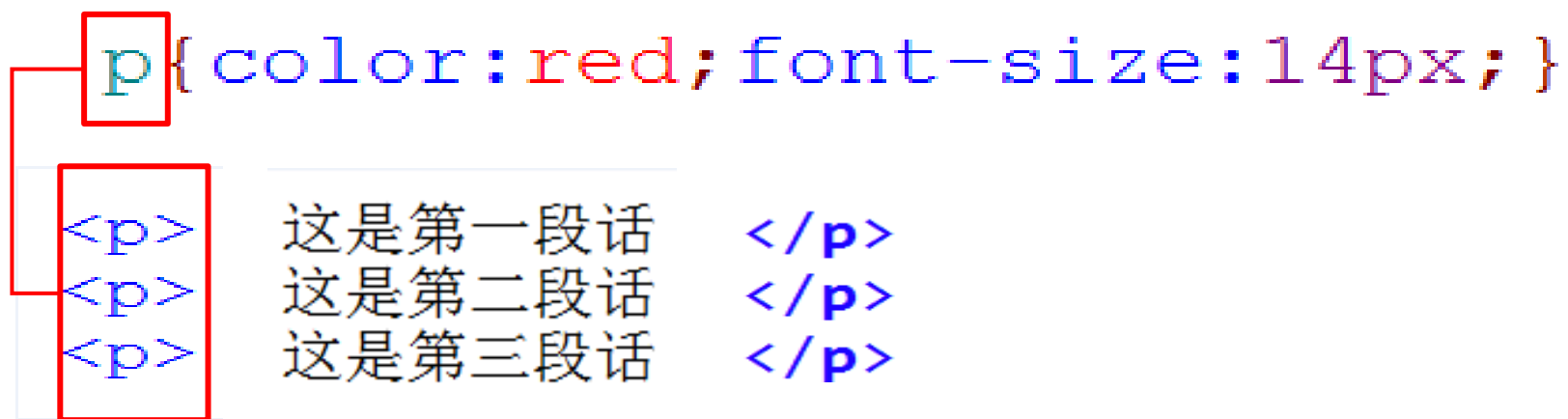
font-size: 20px;

color: red;

}

▶ 选择器

- **选择器**：规定该选择器定义的样式将对哪些元素生效
- **标签选择器** —— 选择器是HTML标签



```
p { color: red; font-size: 14px; }
```

```
<p> 这是第一段话 </p>  
<p> 这是第二段话 </p>  
<p> 这是第三段话 </p>
```

影响整个页面中该种标签的样式。

▶ 选择器

- 类选择器 —— 以 “.” 开头的选择器

```
.red{ color:red; font-size:14px; }
```

```
<p class="red">这是第一段话 <p>  
<p class="red">这是第二段话 <p>  
<h1 class="red">这是一个标签 <h1>
```

影响所有以class属性引用该类的标签样式。

注意：类名的第一个字符不能使用数字！

▶ 选择器

- 元素可以加入多个类。把各个类名放在class属性中，各个类名之间用一个空格分隔，类名的顺序并不重要。

```
<body>
```

```
<p class="bluep redp">我可以有多个类名，通过任何一个类名都可以找到我，  
| 通过不同的类名可以给我添加不同的样式。</p>
```

```
</body>
```

▶ 选择器

- ID选择器 —— 以 “#” 开头的选择器

```
#red{color:red;font-size:14px;}
```

```
<p id="red"> 这是第一段话 </p>
```

影响以id属性引用该选择符的标签样式。

注意：单一页面中，一个id选择器只能使用一次！

▶ 选择器总结

	定义方法	引用方法	与元素对应关系
标签选择器	html标签	直接作用	一对多
类选择器	.自定义	class= ""	多对多
id选择器	#自定义	id= ""	一对一



从内容中抽离出来的CSS样式，
应如何与HTML内容进行整合
使之生效呢？

- 将CSS样式与HTML的内容整合有**三种**方式：
 - 行内样式
 - 页内样式
 - 外部样式

▶ 行内样式

- **行内样式** —— 在元素标签内通过style属性添加样式

```
<body>
```

```
...
```

```
<标签 style="属性:属性值; 属性:属性值; ...">
```

```
...
```

```
</body>
```

```
<h1 style="color:#00FC00">将河北中学首页嵌入到此网页中</h1>
```

用途：指定网页中个别元素的显示效果。不符合样式与内容分离原则，不推荐使用。

▶ 页内样式

- **页内样式** —— 在head部分的style**标签**内添加样式

```
<head>
```

```
<style type="text/css">
```

```
p{color:red; font-size:14px;}
```

```
.green{color:green; font-size:14px;}
```

```
#one{color:yellow; font-size:14px;}
```

```
</style>
```

```
</head>
```

用途：对页面中某些标签或元素设置样式风格。 **控制当前页面样式，维护较困难。**

▶ 外部样式

- **外部样式** —— 引用外部建立的.css文件

```
<head>
```

```
...
```

```
<link rel="stylesheet" type="text/css" href="样式文件路径">
```

```
...
```

```
</head>
```

用途：**可同时控制多个页面**，适用于各类大型网站，**可用性最强**，推荐使用。

▶ CSS样式总结

	优点	缺点	用途
行内样式	使用灵活	只能控制单一标签， 不易维护，代码冗余度高	为网页中某特殊标签进行样式设置时使用
页内样式	同一页面可使用多次	作用范围仅限于定义的页面，其他页面无法使用	对特殊的页面设置单独的样式风格
外部样式	可作用于多个页面， 减少重复工作量， 方便后期维护		适用于各类网站

▶ 练习

● 练习

```
<ul>
```

```
  <li>第一条国际新闻</li>
```

```
  <li>第二条国际新闻</li>
```

```
  <li>第三条国际新闻</li>
```

```
  <li>第四条国际新闻</li>
```

```
  <li>第五条国际新闻</li>
```

```
</ul>
```

```
li {  
    color:red;  
}
```

▶ 练习

● 练习

- 第一条国际新闻
- 第二条国际新闻
- 第三条国际新闻
- 第四条国际新闻
- 第五条国际新闻



```
li {  
  color:red;  
}
```

- 第一条国际新闻
- 第二条国际新闻
- 第三条国际新闻
- 第四条国际新闻
- 第五条国际新闻



```
li {  
  color:green;  
}
```

- 第一条国际新闻
- 第二条国际新闻
- 第三条国际新闻
- 第四条国际新闻
- 第五条国际新闻



```
.red{  
  color:red;  
}  
class = "red"
```

- 第一条国际新闻
- 第二条国际新闻
- 第三条国际新闻
- 第四条国际新闻
- 第五条国际新闻



```
#blue {  
  color:blue;  
}  
id = "blue"
```



如果3种选择器同时作用于某个元素，哪一种选择器起作用呢？

如果某元素同时被设置了行内、页内、外部样式，哪一种样式起作用呢？

▶ 样式优先级

● 样式优先级：

- ID选择器 > 类选择器 > 标签选择器
- 行内样式 > 页内样式 > 外部样式
- 就近原则，距离元素最近的样式优先级最高

- CSS样式的叠加

- 多个样式，在同一内容上共同实现，叫做CSS样式的叠加

- CSS样式的可继承性

- 文档中的某些元素，将沿用为其父元素所设置的样式，这种特点叫做CSS样式的可继承性

- CSS注释方法：

- */*...*/* 多行注释

```
<style>  
    /* #box{  
        color:red;  
    } */  
</style>
```

▶ 练习

● 练习

```
<ul>
```

```
  <li>第一条国际新闻</li>
```

```
  <li>第二条国际新闻</li>
```

```
  <li>第三条国际新闻</li>
```

```
  <li>第四条国际新闻</li>
```

```
</ul>
```

```
<p>第五条国际新闻</p>
```

要求

1. 样式选择页内样式。
2. 为无序列表ul下的文字设置文字大小为12px，文字颜色为蓝色color:blue。
3. 为p标签设置类名，设置其文字颜色为green。

本节内容



CSS简介



CSS基础语法



CSS基础小结

▶ 本节小结

- CSS (层叠样式表)
- 语法：选择器{ 属性：属性值；属性：属性值；..... }
- 选择器：标签选择器、类选择器、ID选择器

标签选择器：

直接使用标签，对所有该标签产生作用，用于控制所有该标签的样式。

类选择器：

用 “.” 定义，在元素中用 class 引用，用于控制所有引用该类的元素的样式。

id选择器：

用 “#” 定义，在元素中用 id 引用，用于控制设置了该id的元素的样式。

单一页面中，一个 id 选择器只能使用一次！

▶ 本节小结

- 在HTML中使用CSS的方法：

- **行内样式**：为某元素添加 style 属性，属性取值为 css 代码。
 - 最灵活，但是不符合样式与内容分离原则，不推荐使用
- **页内样式**：在页面的 head 部分添加 style 标签，将 css 代码写在标签间。
 - 控制单一页面样式，维护较困难，不推荐使用
- **外部样式**：将 css 代码写在外部 .css 文件中，使用时引入。
 - 可用性最强，可同时控制多个页面，推荐使用

▶ 本节小结

- 样式优先级

- ID选择器 > 类选择器 > 标签选择器
- 行内样式 > 页内样式 > 外部样式
- 就近原则，距离元素最近的样式优先级最高

- 参考资料

- <http://www.w3school.com.cn/css/>

谢谢
Thank You

▶ CSS选择器补充

- 通配符选择器 (*****)

- 选择所有元素 ***** { margin: 0; padding: 0; }
- 选择某个元素下的所有元素 **.demo *** {border:1px solid blue;}

- 群组选择器 (**E , F**)

- 将具有**相同样式**的元素分组在一起
- 每个选择器之间使用逗号 “**,**” 隔开 **h1,p,a** {color:red;}

▶ CSS选择器补充

- 后代选择器 (**E F**)

- 选择某元素的**所有**后代元素 `ul li {color: red;}`
- 前面E为祖先元素，F为后代元素，他们之间需要一个**空格**隔开

- 子元素选择器 (**E > F**)

- 选择某元素的**直接**后代元素（第一代） `ul>li {color: red;}`
- 前面E为祖先元素，F为后代元素，他们之间需要一个**大于号**隔开

▶ CSS选择器补充

- 兄弟选择器 (**E + F**)

- E F 两元素具有一个相同的父元素，而且 F 元素在 E 元素后面，而且相邻
- 由E选择F，而不能由F选择E **img+p** img: hover +p

- 属性选择器 (**E [attr]**)

- 选择匹配E元素，且该元素定义了attr属性
- E选择符可以省略，表示选择定义了attr属性的任意类型的元素
- 比如**a[href]**、**img[alt]**

- 在编写CSS样式的时候，会时常碰到书写样式没有生效，尤其是引用一些外部框架的时候，这种情况更容易发生。

- 样式优先级

- ID选择器 > 类选择器 > 标签选择器
- 行内样式 > 页内样式 > 外部样式
- 就近原则，距离元素最近的样式优先级最高

什么是CSS**权重**？



- 权重是CSS样式被浏览器解析生效的规则

- 当多个样式被应用到同一个元素上时，权重决定哪个样式生效。
- 每个选择器都有自己的权重。每条CSS语法，都包含一个权重级别。这个级别是由不同的选择器加权计算的。
- 如果两个选择器同时作用到一个元素上，权重高者生效。

- 加权是如何计算的呢？
- 根据选择器种类的不同可以分为**四类**，也决定了四种不同等级的**权重值**
 - 行内样式 **1000**
 - ID选择器 **100**
 - 类选择器、属性选择器和伪类选择器(:hover) **10**
 - 元素和伪元素(:before :after) **1**

▶ 加权计算

- 将 id 选择器作为 a 组，类选择器作为 b 组，元素名作为 c 组，每组中出现一次，计数一次。

选择器	计算	权重
<code>*{}</code>	<code>a=0 b=0 c=0</code>	0
<code>ul li a{</code>	<code>a=0 b=0 c=3</code>	3
<code>ul#nav li{</code>	<code>a=1 b=0 c=2</code>	102
<code>ul.nav li</code>	<code>a=0 b=1 c=2</code>	12
<code>ul li.active a:hover{</code>	<code>a=0 b=2 c=3</code>	23

如何改变样式优先级？



- **! important**

- 用途：!important允许开发人员**强制**应用某样式，
- 使用方法：写在**该样式的某属性值后，结束分号前**，如：

```
<style>  
    img { width: 150px !important ; }  
</style>
```

- **注意**：不推荐使用!important，建议通过改变选择器类型来改变权重。

▶ 权重

- 通用选择器 (*) < 元素选择器 < 类选择器 = 属性选择器 = 伪类选择器 < ID 选择器 < 行内样式 < !important
- 权重相同，就近原则；权重不同，权重大者有效