

第5章 CSS基础



- CSS引入方式
- 选择符详解

样式的继承

样式的优先级



5.1

CSS引入方式

[点击查看本小节知识架构](#)

5.2

选择符详解

[点击查看本小节知识架构](#)

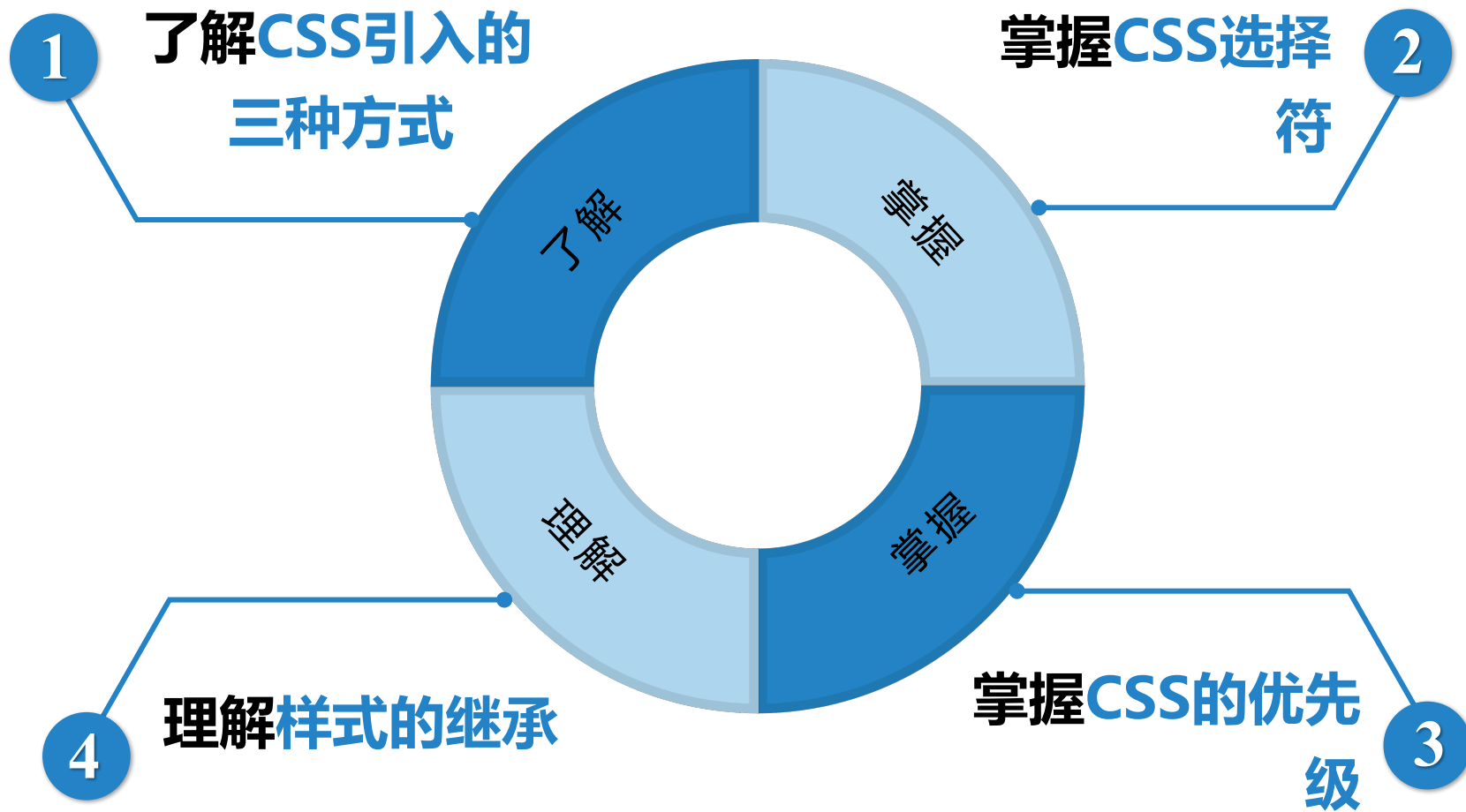
5.3

样式的继承

5.4

样式的优先级







5.1 CSS引入方式



[返回目录](#)

5.1.1

● 内部引入方式

5.1.2

● 外部引入方式

5.1.3

● 三种方式的对比





5.1 CSS引入方式

5.1.1 内部引入方式

- 内部引入方式是通过<style>标签添加CSS样式，它与行间样式引入方式的区别是内部引入是通过<style>标签而不是style属性。<style>标签一般情况下都会添加到<head>标签当中。具体示例如下：

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>内部引入方式</title>
6 <style>
7     /* 这里添加 CSS 的内部样式 */
8 </style>
9 </head>
10 <body>
11     内容
12 </body>
13 </html>
```

- CSS中通过/**/ 的形式添加注释，与HTML中添加注释的方式不一样。/* 作为注释的起始，*/ 作为注释的结束。和HTML中添加注释一样，在Dreamweaver中也可以通过按钮选择添加CSS注释。





5.1 CSS引入方式

5.1.1 内部引入方式

- 内部引入CSS样式，要遵从一定的CSS样式规范，其语法格式如下：

```
选择符{ 属性 1:值 1;属性 2:值 2; }
```

- 选择符是选择指定标签的方法，通过这种方法，可以选择到指定的HTML标签，并且给其添加CSS样式。例如页面中有三个<p>标签，具体示例如下。

```
1 <body>
2     <p>第一个 p 标签</p>
3     <p>第二个 p 标签</p>
4     <p>第三个 p 标签</p>
5 </body>
```

- 如果想让中间的<p>标签添加文字颜色，而其他两个<p>标签不作任何处理，就需要用到选择符，下面先来简单讲解一下id选择符和class选择符。





5.1 CSS引入方式

5.1.1 内部引入方式

- 1.id选择符
- id选择符是可以给指定的标签设置一个id属性和一个id值，然后通过id选择符（即“# + id值”）找到对应的标签为其设置id值，即添加CSS样式。
- 图5.2 id选择符展示效果
- id选择符的特点是，在同一个页面中，不允许出现两个相同的id值，就像每个人的身份证号都是唯一的，id选择符也具有唯一性。
- id选择符的错误使用方式，具体示例如下。

```
1 <body>
2 <p id="elem">第一个 p 标签</p>
3 <p id="elem">第二个 p 标签</p>
4 <p id="elem">第三个 p 标签</p>
5 </body>
```

- 示例中将三个<p>标签的id选择符都设置为elem显然是CSS不允许的，编译会出错。





5.1 CSS引入方式

5.1.1 内部引入方式

- 2.class选择符
- class选择符也叫“类”选择符，可以给指定的标签设置一个class属性和class值，然后通过class选择符（即“.” + class值”）找到对应的标签，为其设置class值，即添加CSS样式。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>内部引入方式</title>
6 <style>
7     .elem{ color:red;}
8 </style>
9 </head>
10 <body>
11 <p>第一个 p 标签</p>
12 <p class="elem">第二个 p 标签</p>
13 <p class="elem">第三个 p 标签</p>
14 </body>
15 </html>
```

- class选择符的特点跟id选择符不同，在同一个页面中，class值可以重复出现，即可以重复利用CSS样式。





5.1 CSS引入方式

5.1.2 外部引入方式

- 外部样式的写法与内部样式的写法完全一样，只是外部引入方式将CSS代码保存在扩展名为.css的样式表中，在HTML文件中引用扩展名为.css的样式表。引入的方式用链接式和导入式两种方式。下面将详细讲解这两种外部引入CSS的方式。

- 1. 链接式

- 链接式是将所有的样式放在一个或多个 “.css” 为扩展名的外部样式表中，通过网页的<head> </head> 标签对中使用<link> 标记将外部样式表文件链接到HTML文档中。其语法格式如下：

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

- 语法中必须指定<link> 标记的三个属性，href是定义链接外部样式表文件的url，可以是相对路径和绝对路径；rel是定义当前文档与被链接文档之间的关系，这里指定为stylesheet，表示被链接的文档是样式表文件；type是定义链接文档的类型，这里类型指定为text/css，表示链接的外部文件为CSS样式表。





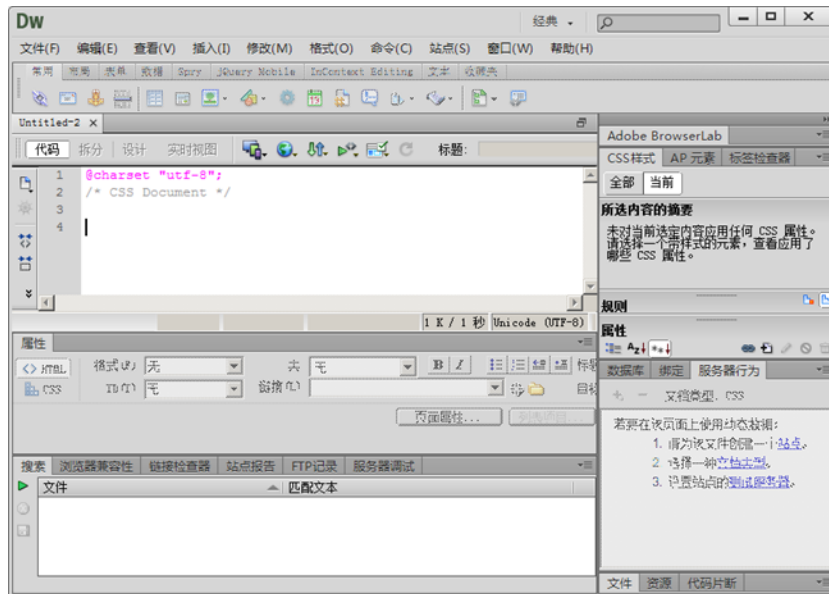
5.1 CSS引入方式

5.1.2 外部引入方式

- 这里面涉及到三个属性，rel= “stylesheet” 规定当前文档与被链接文档之间的关系。type= “text/css” 规定被链接文档的 MIME 类型。href= “mystyle.css” 规定被链接文档的位置。只需要记住这个固定写法即可。
- （1）首先打开DW软件，创建一个HTML文档，并在该文档中添加标题和段落文本。将HTML文档命名为5-4.html，保存。
- （2）创建CSS样式表，在DW选择菜单栏中选择【文件】→【新建】命令，弹出“新建文档”对话框。创建一个CSS样式表，弹出CSS文档编辑窗口，如下图所示。



5.1.2 外部引入方式



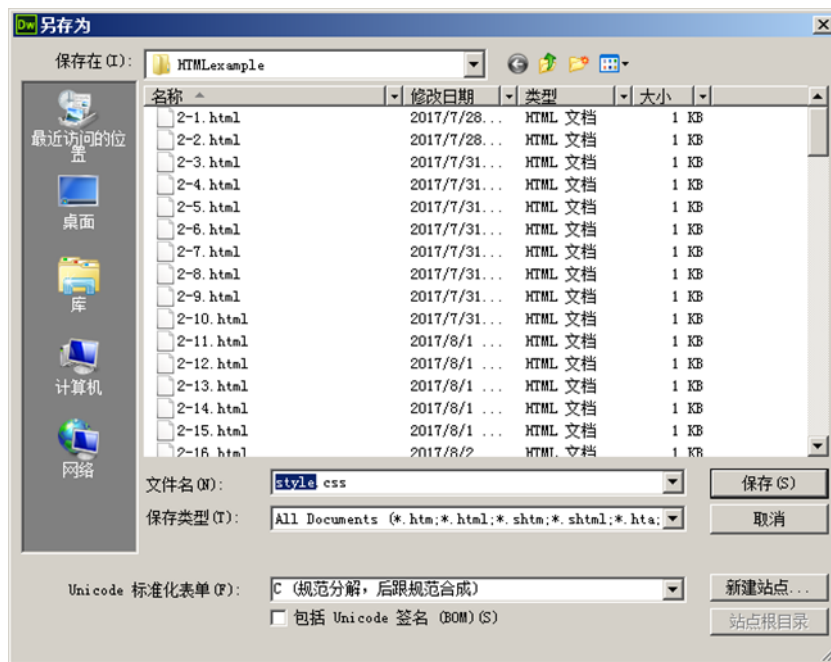
- （3）书写CSS样式，在图中的文档编辑窗口中输入以下代码。

```
p{ font-size:18px; color:red; text-decoration:underline;}  
/*定义文本修饰样式*/
```



5.1.2 外部引入方式

- （4）保存CSS演示表文件，选择[【文件】→【保存】]，弹出“另存为”对话框。



- 在图中，将文件命名为style.css，保存在5-4.html文件所在的文件夹中。





5.1 CSS引入方式

5.1.2 外部引入方式

- （5）链接CSS样式表。在例5-4的<head>头部标签中，添加<link>语句，将style.css外部样式表文件链接到5-4.html文档中，一般<link>标签都会写在<meta>标签和<title>标签之间。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link href="style.css" rel="stylesheet" type="text/css">
6 <title>外部引入方式</title>
7 </head>
8 <body>
9 <h3 align="center">链接式引入方式</h3>
10 <p>通过 link 标签将扩展名为.css 的外部样式表文件链接到 HTML 文档中</p>
11 </body>
12 </html>
```

- 链接式引入同一个CSS样式表可以被不同的HTML文档链接使用，一个HTML文档也可以通过多个<link>标签链接多个CSS样式表，这是其最大的好处。





5.1 CSS引入方式

5.1.2 外部引入方式

- 2. 导入式
- 导入式是将一个独立的.css文件导入HTML文档中，其是在HTML文档<head>标签中应用<style>标签，并在<style>标签中的开头处用@import语句，即可导入外部样式表文件。其基本语法格式如下：

```
<style type="text/css">  
    @import url(CSS 文件路径);或@import "CSS 文件路径"  
    /*此处还可存放其他 CSS 样式*/  
</style>
```

- 语法中，style标签内还可以存放其他的内嵌样式，@import语句需要位于其他内嵌样式的上面。
- 如果对例5-4使用导入式CSS演示，只需将HTML文档中的<link>语句替换成<style>标签即可，具体示例如下。





5.1 CSS引入方式

5.1.2 外部引入方式

- 如果对例5-4使用导入式CSS演示，只需将HTML文档中的<link>语句替换成<style>标签即可，具体示例如下。

```
<style type="text/css">  
    @import "style.css";  
</style>
```

- 或者

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

- 导入式会在整个网页加载完后再加载CSS文件，因此如果网页比较大则会出现先显示无样式的页面，再出现网页的样式的情况。这是导入式固有的一个缺陷。





5.1 CSS引入方式

5.1.2 外部引入方式

- 虽然导入式和链接式功能基本相同，但大多数网站都是采用链接式引入外部样式表。这是因为两者的加载时间和顺序不同。加载页面时，<link>标签引用的CSS样式表将同时被加载，而@import引用的CSS样式表会等整个网页下载结束后再被加载。可能会显示无样式的页面，造成不好的用户体验。因此大多数的网站采用链接式的引入方式。
- 链接式是使用频率最高、最实用的CSS样式表。它可以将HTML代码与CSS代码分离为两个或多个文件，实现类结构和表现的完全分离，使网页的前期制作和后期维护都变得十分方便。





5.1 CSS引入方式

5.1.3 三种方式的对比

- 前面学习了CSS的三种引入方式，在实际开发中，为了提升网站的性能和维护性，一般都是使用外部引入方式，下面通过对比来了解一下。
- 比如要给两个不同的标签添加相同的样式，首先查看使用行间引入方式。

```
1 <body>
2     <div style="color:red;">我是一个 div 标签</div>
3     <p style="color:red;">我是一个 p 标签</p>
4 </body>
```

- 接下来查看内部引入方式。

```
1 <style>
2     .elem{ color : red; }
3 </style>
4 <body>
5     <div class="elem">我是一个 div 标签</div>
6     <p class="elem">我是一个 p 标签</p>
7 </body>
```



5.1.3 三种方式的对比

- 上面示例代码可以看出，使用行间引入方式，直接对每个标签应用样式有它的优点，但是这种引入方式需要逐个地对元素设置样式，而且软件更新时需要逐个地对标签进行样式调整，可维护性差，使用内部引入方式可以通过选择器来改变元素的样式，可以重复使用代码，这样做可以提升性能。如果修改样式，只需修改一次就可以对所有设置此样式的标签同时修改，可维护性好。
- 相比内部样式，外部样式有更好的提升性和维护性，这是因为可以在不同的页面引入同一个CSS样式文件，可以在多个页面之间复用CSS样式。实现结构和表现的完全分离，使得网页的前期开发和后期维护都变得十分方便。





5.2 选择符详解



[返回目录](#)

5.2.1

● id选择符

5.2.2

● class选择符

5.2.3

● tag选择符

5.2.4

● 通配选择符





5.2 选择符详解



[返回目录](#)

5.2.5

● 组选择符

5.2.6

● 包含选择符

5.2.7

● 伪类选择符



5.2.1 id选择符

- 上一节介绍了id选择符的基本使用，下面来了解选择符的命名规范，一般尽量采用语义化的英文单词，不能以数字或特殊字符开头。具体示例如下。

```
1 <style>
2     #123elem{ color : red; }    /*命名错误*/
3     #%@elem{ color : green; }   /*命名错误*/
4     #container{ color : blue; } /*命名正确*/
5 </style>
```

- 对于复杂一些的命名还可以采用驼峰命名方式或下画线方式。

```
1 <style>
2     #containerMailTop{ color : red; }    /*驼峰式命名*/
3     #login_gray_icon{ color : green; }   /*下划线命名*/
4 </style>
```





5.2 选择符详解

5.2.2 class选择符

- 同样上一节中讲解了class选择符的基本使用。下面介绍下class样式的多组值添加方式，多组值通过空格隔开。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>class 选择符</title>
6 <style>
7     .boxSize{ width:200px; height:100px; }
8     .boxColor{ background:red; }
9     .boxBorder{ border:5px black solid; }
10 </style>
11 </head>
12 <body>
13 <div class="boxSize boxColor boxBorder"></div>
14 </body>
15 </html>
```





5.2 选择符详解

5.2.3 tag选择符

- tag选择符也叫标签选择符。可直接通过HTML标签的名字来设置样式。
- 在同一个HTML网页中，相同的标签可能有不同的样式，具体示例如下。

```
1 <style>
2     /* 带有 active 样式的 div 标签添加颜色 */
3 </style>
4 <body>
5     <div class="active">第一个 div 标签</div>
6     <div>第二个 div 标签</div>
7     <p class="active">第一个 p 标签</p>
8     <p>第二个 p 标签</p>
9 </body>
```

- 上面的示例代码中需要将带有active样式的<div>标签添加颜色，但是对同样带有active样式的<p>标签没有影响，这时就需要tag选择符和class选择符组合来设置选择符，使网页能够只选择<div>标签中带active的class样式，而不去选择其他标签中带active的class样式。





5.2 选择符详解

5.2.4 通配选择符

- 通配选择符可以把样式通用在所有的标签当中，通过星号（*）的方式来设置，要慎用通配选择符。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>通配选择符</title>
6 <style>
7     *{ color:red; }
8 </style>
9 </head>
10 <body>
11 <div>这是一个 div 标签</div>
12 <p>这是一个 p 标签</p>
13 <h1>这是一个 h1 标签</h1>
14 </body>
15 </html>
```





5.2 选择符详解

5.2.5 分组选择符

- 分组选择符可以简化相同样式的操作。通过逗号(,)来进行分组设置。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>分组选择符</title>
6 <style>
7     div,p,h1{ color:red; }
8 </style>
9 </head>
10 <body>
11 <div>这是一个 div 标签</div>
12 <p>这是一个 p 标签</p>
13 <h1>这是一个 h1 标签</h1>
14 <span>这是一个 span 标签</span>
15 </body>
16 </html>
```

- 第7行将div、p、h1标签统一设置颜色，和把标签分开设置颜色效果相同，但是这种组合写法提供了可维护性。



5.2.6 包含选择符

- 包含选择符是指被选标签被其他标签所包含，从而通过筛选的方式进行操作，通过空格来进行包含设置。例如只需要ul中的li添加文本颜色，而ol中的li不做任何处理，这时就要用到包含选择符进行选择。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>包含选择符</title>
6 <style>
7     ul li{ color:red; }
8 </style>
9 </head>
10 <body>
11 <ul>
12     <li>千锋教育</li>
13     <li>扣丁课堂</li>
14     <li>千问千知</li>
15 </ul>
16 <ol>
17     <li>千锋教育</li>
18     <li>扣丁课堂</li>
19     <li>千问千知</li>
20 </ol>
21 </body>
22 </html>
```



5.2.7 伪类选择符

- 为了提高用户体验，定义超链接时，经常需要为超链接指定不同的状态，表示超链接在单击前、单击后和鼠标悬停时的不同的样式。在CSS中，通过链接伪类选择符可以实现不同的链接状态。伪类选择符用于向某些HTML标签添加特殊的效果。伪类选择符包含的内容很多，在CSS基础部分先学习link、hover、active和visited四个伪类选择符，其中的link、visited这两个伪类选择符只能用在<a>标签上，hover、active这两个伪类选择符则可以用在所有的标签上。伪类选择符通过冒号(:)的方式来设置。下面将详细介绍这四个伪类选择符。
 - 1.link伪类
 - link伪类是用来设置<a>标签“未访问”时的样式。





5.2 选择符详解

5.2.7 伪类选择符

- 2.hover伪类
- hover伪类是用来设置HTML标签“鼠标划过”时的样式。
- 3.active伪类
- active伪类是用来设置HTML标签“鼠标按下”时的样式。
- 4.visited伪类
- visited伪类是用来设置<a>标签“访问过后”时的样式。



- 样式的继承可以理解为子类的样式从父标签或祖先标签中继承过来。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>样式的继承</title>
6 <style>
7     div{ font-size : 30px; color : red; border : 1px #000 solid; }
8 </style>
9 </head>
10 <body>
11 <div>
12     <p>这是一个段落</p>
13 </div>
14 </body>
15 </html>
```

- 当给<div>标签添加边框样式时，并没有被<p>标签所继承。这里总结下，与元素外观（文字颜色、字体等）相关的样式会被继承；与元素在网页上的布局相关的样式不会被继承。在样式中使用inherit这个特别设立的值可以强行继承，明确指示浏览器在该属性上使用父元素样式中的值。





5.4 样式的优先级

- 在前面的小节中已经学习了很多选择符，可以给一个HTML标签同时作用多个选择符，当出现多样式的时候，就会出现它们的优先级问题，本节将从相同样式优先级、内部样式和外部样式、单一样式优先级、!important、标签+类与单类、分组优先级和包含优先级几种情况分别讲解样式的优先级及其使用。
- 1.相同样式优先级
- 当设置相同样式时，后面的优先级较高，但不建议出现重复设置样式的情况。
- 2.内部样式与外部样式
- 内部样式与外部样式优先级相同，如果都设置了相同样式，那么后写的引入方式优先级高。



- 3.单一样式优先级
- 单一样式的优先级规则如图所示。

style行间样式 > id选择符 > class选择符 > tag选择符 > 默认继承

- 接下来通过分别比较不同选择符的优先级，来验证单一样式的优先级规则。
- （1）tag选择符与默认继承优先级比较，tag选择符的优先级高于默认继承。
- （2）class选择符与tag选择符优先级比较，class选择符的优先级比tag选择符的优先级高。
- （3）id选择符与class选择符优先级比较。id选择符的优先级比class选择符优先级高。
- （4）style行间样式与id选择符优先级比较，style行间样式优先级比id选择符优先级高。





5.4 样式的优先级

- 4. !important
- !important表示“重要的”，用来提升样式优先级。当给指定的样式添加!important时，表示当前样式优先级最高，此时可以不用按照上一小节中的规则。（注意!important对默认继承不起作用）。
- 5. 标签+类与单类
- 标签+类的优先级要大于单类的优先级。
- 6. 分组优先级
- 分组选择符与单一选择符的优先级相同，靠后写的优先级高。
- 7. 包含优先级
- 包含选择符的优先级相对比较复杂，下面通过案例讲解。





5.4 样式的优先级

```
1 <doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>样式的优先级</title>
6 <style>
7     ul #listLi .listDiv .listP{ color : red; }
8     #list .listLi div p{ color : green; }
9 </style>
10 </head>
11 <body>
12 <ul id="list" class="list">
13     <li id="listLi" class="listLi">
14         <div id="listDiv" class="listDiv">
15             <p id="listP" class="listP">这是一个段落</p>
16         </div>
17     </li>
18 </ul>
19 </body>
20 </html>
```

- 显示字体为红色，因此第7行优先级比第8行优先级高，可以利用“约分法”来比较包含选择符的优先级高低。“约分法”的做法是将相同类型的选择符进行约分处理，比如id选择符跟id选择符进行约分，class选择符与class选择符进行约分，然后看最终约分后的结果来比较优先级高低。



- 上例中的约分结果如图所示。

```
ul #listLi .listDiv .listP{ color : red; }  
#list .listLi div p{ color : green; }
```

- 通过“约分法”处理后，第7行的 .listP（class选择符）优先级高于第8行的p（tag选择符）优先级，因此最终文字颜色为红色。





本章小结

- 通过本章的学习，了解CSS样式的三种引入方式，同时掌握CSS的选择符和CSS的优先级，以及理解CSS的继承操作。本章都是CSS中重要的内容，希望读者多实践和记忆这些重要内容。在下一章中将学习CSS进阶部分，来了解更多CSS高级的知识点。



THANK YOU

