CSS高级选择器、继承和层叠

# 目录

[目录 2](#_Toc464474924)

[一、css基础选择器 3](#_Toc464474925)

[1.1 标签选择器 3](#_Toc464474926)

[1.2 id选择器 3](#_Toc464474927)

[1.3 类选择器 4](#_Toc464474928)

[二、css高级选择器 6](#_Toc464474929)

[2.1 后代选择器 6](#_Toc464474930)

[2.2 交集选择器 7](#_Toc464474931)

[2.3 并集选择器（分组选择器） 8](#_Toc464474932)

[2.4 通配符\* 8](#_Toc464474933)

[三、CSS的继承性和层叠性 8](#_Toc464474934)

[3.1 继承性 8](#_Toc464474935)

[3.2 层叠性 9](#_Toc464474936)

# 一、css基础选择器

html负责结构，css负责样式，js负责行为。

css写在head标签里面，容器style标签。

先写选择器，然后写大括号，大括号里面是样式。

|  |
| --- |
| 1. <style type="text/css"> 2. body{ 3. background-color: pink; 4. } 5. </style> |

常见属性：

|  |
| --- |
| 1. h1{ 2. color:blue; 3. font-size: 60px; 4. font-weight: normal; 5. text-decoration: underline; 6. font-style: italic; 7. } |

## 1.1 标签选择器

就是用标签名来当做选择器。

1） 所有标签都能够当做选择器，比如body、h1、dl、ul、span等等

2） 不管这个标签藏的多深，都能够被选择上。

3） 选择的是所有的，而不是某一个。所以是共性，而不是特性。

比如网易，希望页面上所有的超级链接都没有下划线：

|  |
| --- |
| 1. a{ 2. /\*去掉下划线：\*/ 3. text-decoration: none; 4. } |

## 1.2 id选择器

#表示选择id

|  |
| --- |
| 1. #lj1{ 2. font-size: 60px; 3. font-weight: bold; 4. color:black; 5. } |

1）任何的标签都可以有id，id的命名要以字母开头，可以有数字、下划线。大小写严格区别，也就是说mm和MM是两个不同的id。

2）同一个页面内id不能重复，即使不一样的标签，也不能是相同的id。

也就是说，如果有一个p的id叫做haha，这个页面内，其他所有的元素的id都不能叫做haha。

## 1.3 类选择器

**.就是类的符号**。类的英语叫做class。

所谓的类，就是class属性，class属性和id非常相似，**任何的标签都可以携带class属性**。

**class属性可以重复，比如，页面上可能有很多标签都有teshu这个类**：

|  |
| --- |
| 1. <h3>我是一个h3啊</h3> 2. <h3 **class="teshu"**>我是一个h3啊</h3> 3. <h3>我是一个h3啊</h3> 4. <p>我是一个段落啊</p> 5. <p **class="teshu"**>我是一个段落啊</p> 6. <p **class="teshu"**>我是一个段落啊</p> |

css里面用.来表示类：

|  |
| --- |
| 1. **.**teshu{ 2. color: red; 3. } |

**同一个标签，可能同时属于多个类，用空格隔开**：

|  |
| --- |
| 1. <h3 class="**teshu zhongyao**">我是一个h3啊</h3> |

这样，这个h3**就同时属于teshu类，也同时属于zhongyao类**。

初学者常见的错误，就是写成了两个class：

|  |
| --- |
| 1. <h3 **class="teshu" class="zhongyao"**>我是一个h3啊</h3> |

所以要总结两条：

1） class可以重复，也就是说，同一个页面上可能有多个标签同时属于某一个类；

2） 同一个标签可以同时携带多个类。

类的使用，能够决定一个人的css水平。

比如，我们现在要做一个页面：

段落1

段落2

段落3

正确的思路，就是用所谓“公共类”的思路，就是我们类就是提供“公共服务”，比如有绿、大、线，一旦携带这个类名，就有相应的样式变化：

|  |
| --- |
| 1. <style type="text/css"> 2. .lv{ 3. color:green; 4. } 5. .da{ 6. font-size: 60px; 7. } 8. .xian{ 9. text-decoration: underline; 10. } 11. </style> |

每个标签，就去选取自己想要的类：

|  |
| --- |
| 1. <p **class="lv da"**>段落1</p> 2. <p **class="lv xian"**>段落2</p> 3. <p **class="da xian"**>段落3</p> |

也就是说：

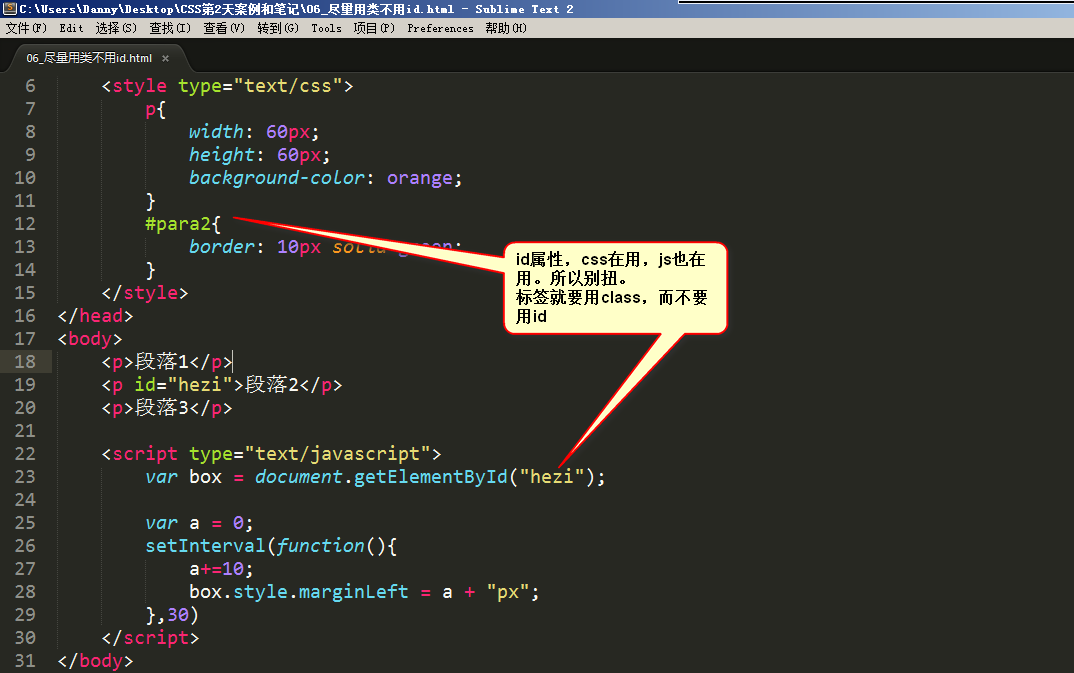
1） 不要去试图用一个类名，把某个标签的所有样式写完。这个标签要多携带几个类，共同造成这个标签的样式。

2） 每一个类要尽可能小，有“公共”的概念，能够让更多的标签使用。

到底用id还是用class？

答案：尽可能的用class，除非极特殊的情况可以用id。

原因：id是js用的。也就是说，js要通过id属性得到标签，所以我们css层面尽量不用id，要不然js就很别扭。另一层面，我们会认为一个有id的元素，有动态效果。



类上样式，id上行为

就是一个标签，可以同时被多种选择器选择，标签选择器、id选择器、类选择器。这些选择器都可以选择上同一个标签，从而影响样式，这就是css的cascading“层叠式”的第一层含义。

# 二、css高级选择器

## 2.1 后代选择器

|  |
| --- |
| 1. <style type="text/css"> 2. **.div1 p**{ 3. color:red; 4. } 5. </style> |

**空格就表示后代**，.div1 p 就是.div1的后代所有的p。

强调一下，选择的是后代，不一定是儿子。

比如：

|  |
| --- |
| 1. <div class="div1"> 2. <ul> 3. <li> 4. <p>段落</p> 5. <p>段落</p> 6. <p>段落</p> 7. </li> 8. </ul> 9. </div> |

能够被下面的选择器选择上：

|  |
| --- |
| 1. .div1 p{ 2. color:red; 3. } |

所以，看见这个选择器要知道是后代，而不是儿子。选择的是所有.div1“中的”p，就是后代p。

空格可以多次出现。

|  |
| --- |
| 1. .div1 .li2 p{ 2. color:red; 3. } |

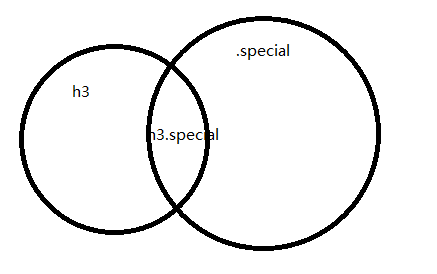
就是.div1里面的后代.li2里面的p。

后代选择器，就是一种平衡：共性、特性的平衡。当要把某一个部分的所有的什么，进行样式改变，就要想到后代选择器。

后代选择器，描述的是祖先结构。



## 2.2 交集选择器



|  |
| --- |
| 1. h3.special{ 2. color:red; 3. } |

选择的元素是同时满足两个条件：必须是h3标签，然后必须是special标签。

**交集选择器没有空格。**

所以有没有空格

|  |
| --- |
| 1. div.red |

和

|  |
| --- |
| 1. div .red |

不是一个意思。

交集选择器可以连续交（一般不要这么写）

|  |
| --- |
| 1. h3.special.zhongyao{ 2. color:red; 3. } |

**交集选择器，我们一般都是以标签名开头，比如div.haha 比如p.special。**

## 2.3 并集选择器（分组选择器）

|  |
| --- |
| 1. h3,li{ 2. color:red; 3. } |

用逗号就表示并集。

## 2.4 通配符\*

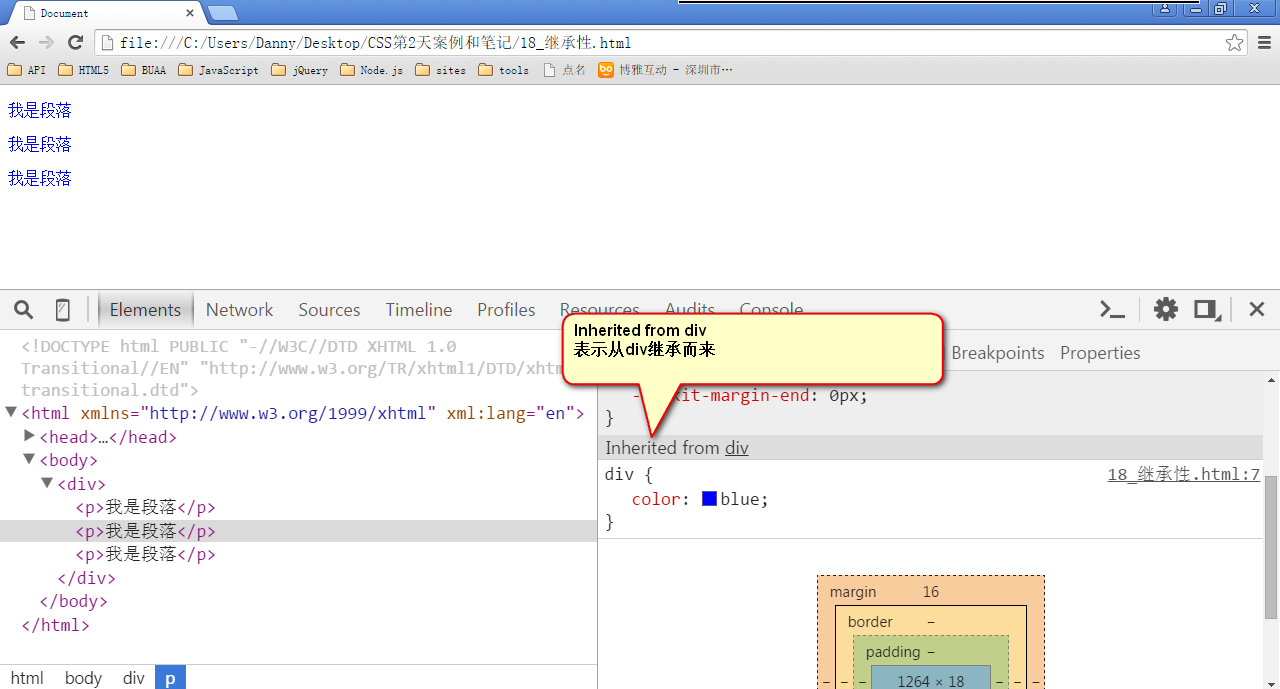
\*就表示所有元素。

|  |
| --- |
| 1. \*{ 2. color:red; 3. } |

效率不高，如果页面上的标签越多，效率越低，所以页面上不能出现这个选择器。

# 三、CSS的继承性和层叠性

## 3.1 继承性



有一些属性，当给自己设置的时候，自己的后代都继承上了，这个就是继承性。

哪些属性能继承？

color、 text-开头的、line-开头的、font-开头的。

**这些关于文字样式的，都能够继承； 所有关于盒子的、定位的、布局的属性都不能继承。**

所以，如果我们的页面的文字，都是灰色，都是14px。那么就可以利用继承性：

|  |
| --- |
| 1. body{ 2. color:gray; 3. font-size:14px; 4. } |

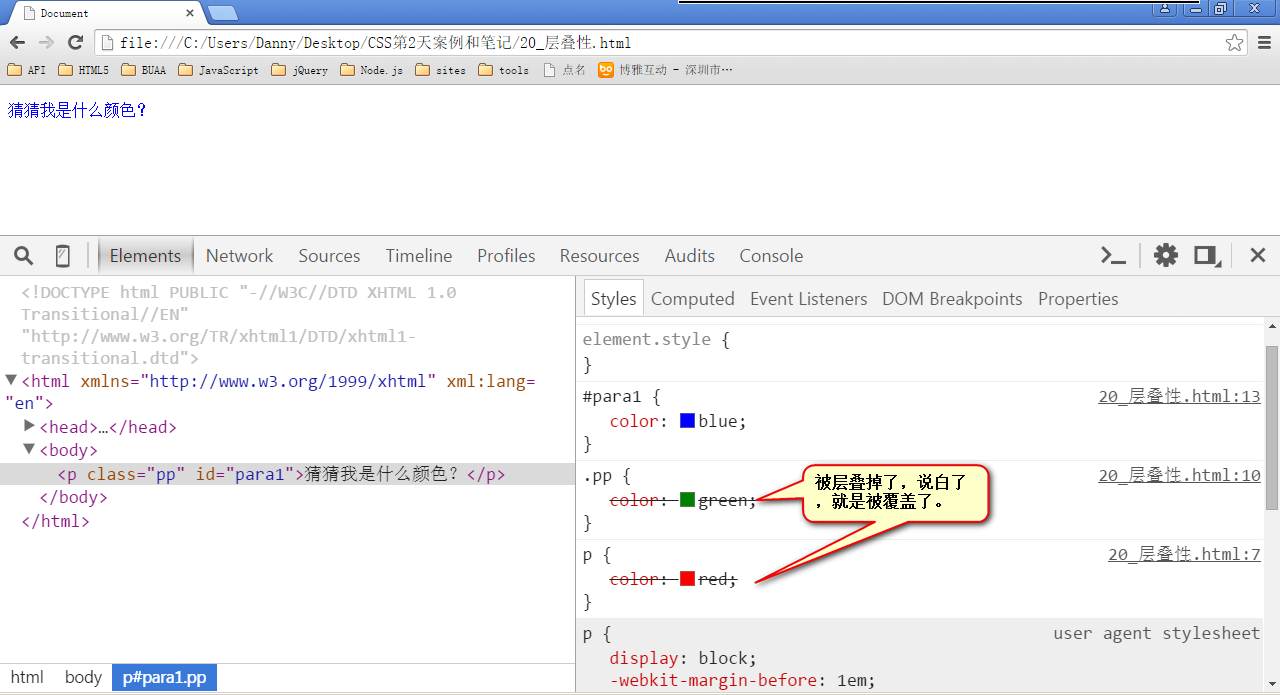
继承性是从自己开始，直到最小的元素。

## 3.2 层叠性

很多公司如果要笔试，那么一定会考层叠性。

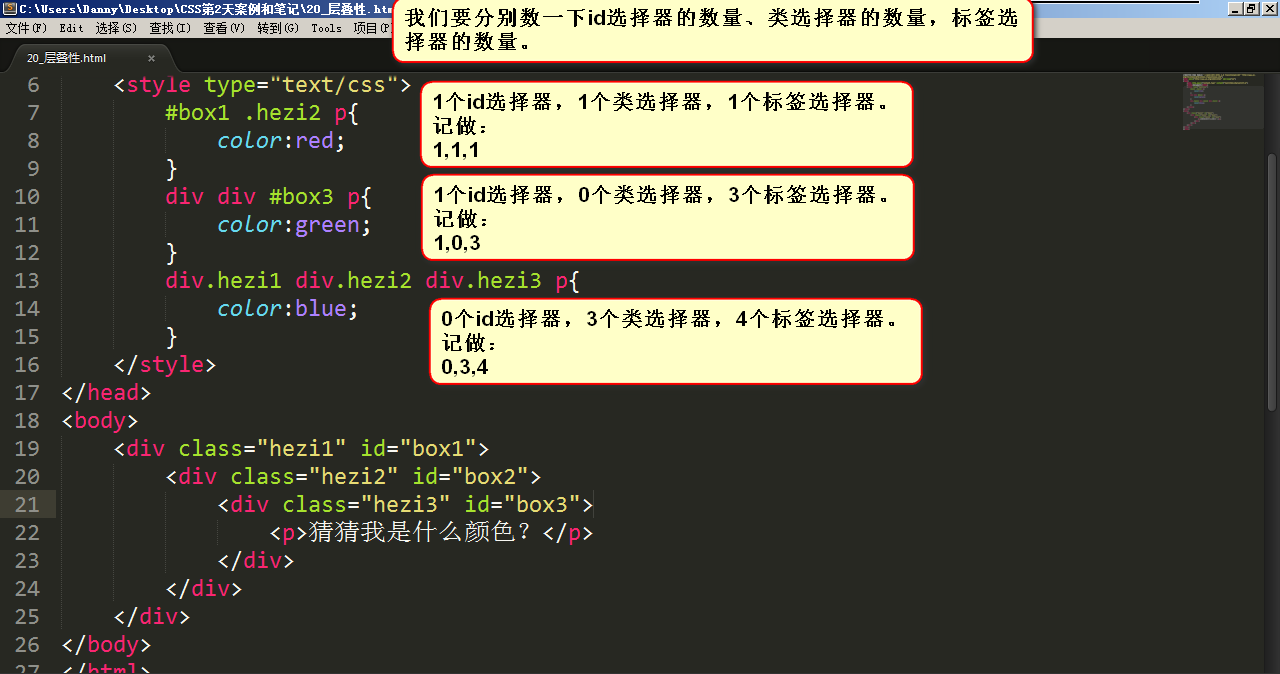
**层叠性：就是css处理冲突的能力。 所有的权重计算，没有任何兼容问题！**

**CSS像艺术家一样优雅，像工程师一样严谨。**

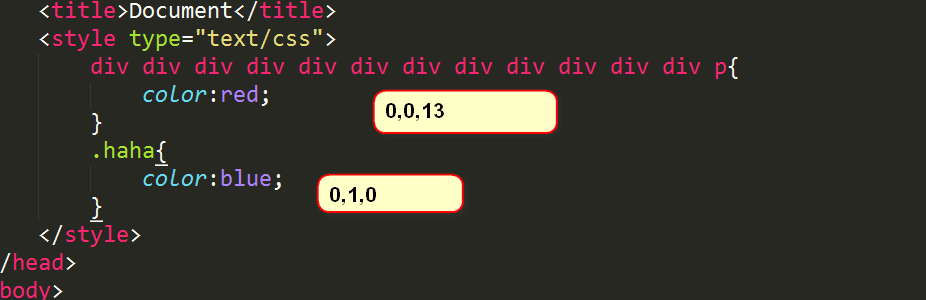


当选择器，选择上了某个元素的时候，那么要这么统计权重：

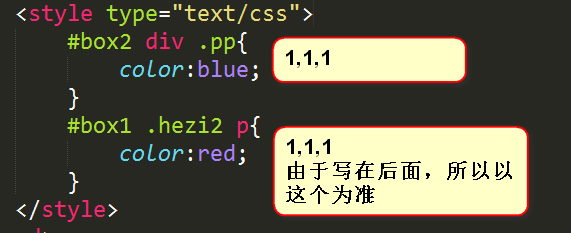
id的数量，类的数量，标签的数量



不进位，实际上能进位（奇淫知识点：255个标签，等于1个类名）但是没有实战意义！



如果权重一样，那么以后出现的为准：



================================我是华丽丽的分界线=================================

如果不能直接选中某个元素，通过继承性影响的话，那么权重是0。



**如果大家都是0，那么有一个就近原则：谁描述的近，听谁的。**



权重问题大总结：

1） 先看有没有选中，如果选中了，那么以（id数，类数，标签数）来计权重。谁大听谁的。如果都一样，听后写的为准。

2） 如果没有选中，那么权重是0。如果大家都是0，就近原则。

题目演示：

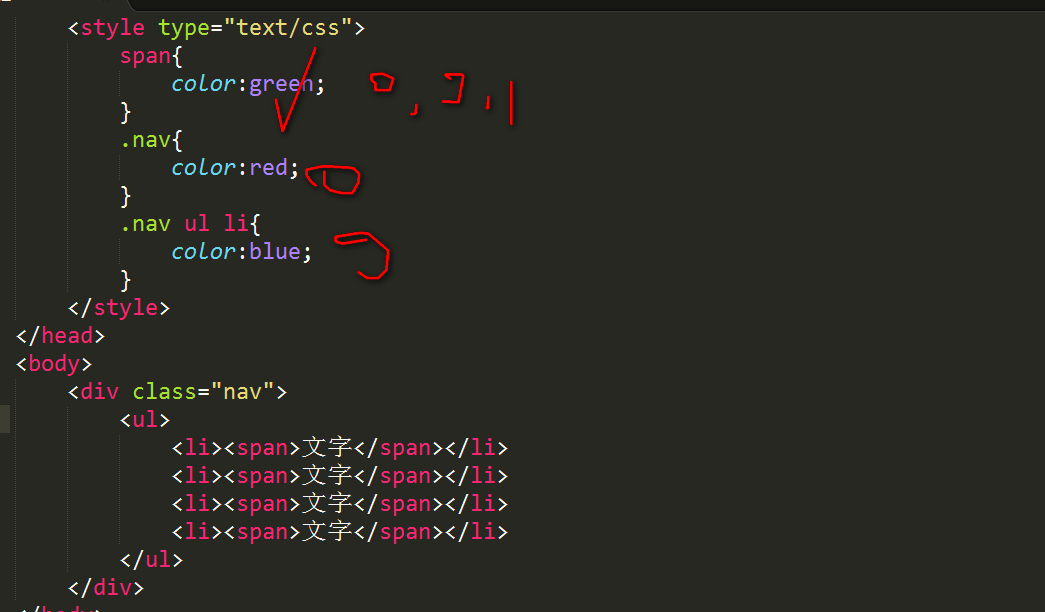
第1题：



第2题：



第3题：



第4题：



继承性：好的事儿。继承从上到下，哪些能？哪些不能？

层叠性：冲突，多个选择器描述了同一个属性，听谁的？