CSS基础入门

第1天讲义（本课程共5天）

# 

# CSS

现在的互联网前端三层：

HTML 超文本标记语言 从**语义**的角度描述页面**结构**。

CSS 层叠式样式表 从**审美**的角度负责页面**样式**。

JS JavaScript 从**交互**的角度描述页面**行为**。

我们现在要开始学习CSS了，就是样式。

## CSS整体感知

css是**c**ascading **s**tyle **s**heet 层叠式样式表的简写。“层叠式”的意思，我们将慢慢的去理解。

|  |
| --- |
| 1. <style type="text/css"> 2. p{ 3. color:red; 4. font-size: 30px; 5. text-decoration: underline; 6. font-weight: bold; 7. text-align: center; 8. font-style: italic; 9. } 10. h1{ 11. color:blue; 12. font-size: 50px; 13. font-weight: bold; 14. background-color: pink; 15. } 16. </style> |

我们写css的地方是style标签，就是“样式”的意思，写在head里面。

type表示“类型” ， text就是“纯文本”。css也是纯文本的。

sublime这里如果想自动生成，那么要输入：



<st 然后按tab键。

**css对换行不敏感，对空格也不敏感。但是一定要有标准的语法。冒号，分号都不能省略。**

语法：

|  |
| --- |
| 1. 选择器{ 2. k:v; 3. k:v; 4. k:v; 5. k:v; 6. } |

## 一些常见的属性

字符颜色：

|  |
| --- |
| 1. color:red; |

color属性的值，可以是英语单词，比如red、blue、yellow等等；也可以是rgb、十六进制，不要着急，后几天讲。

sublime中的快捷键是c，然后tab

字号大小：

|  |
| --- |
| 1. font-size:40px; |

font就是“字体”，size就是“尺寸”。px是“像素”。

单位必须加，不加不行。

sublime中的快捷键是fos，然后tab

背景颜色：

|  |
| --- |
| 1. background-color: blue; |

background就是“背景”。

sublime中的快捷键是bgc，然后tab

加粗：

|  |
| --- |
| 1. font-weight: bold; |

font是“字体” weight是“重量”的意思，bold粗。

sublime中的快捷键是fwb，然后tab

不加粗：

|  |
| --- |
| 1. font-weight: normal; |

normal就是正常的意思

sublime中的快捷键是fwn，然后tab

斜体：

|  |
| --- |
| 1. font-style: italic; |

italic就是“斜体”

sublime中的快捷键是fsi，然后tab

不斜体：

|  |
| --- |
| 1. font-style: normal; |

sublime中的快捷键是fsn，然后tab

下划线：

|  |
| --- |
| 1. text-decoration: underline; |

decoration就是“装饰”的意思。

sublime中的快捷键是tdu，然后tab

没有下划线：

|  |
| --- |
| 1. text-decoration:none; |

sublime中的快捷键是tdn，然后tab

css没啥难的，就是要把属性给记忆准确。

# 基础选择器

css怎么学？很简单有两个知识部分：

1） 选择器，怎么选；

2） 属性，样式是什么

## 标签选择器

就是标签的名字。

|  |
| --- |
| 1. <h1>北京贝沃汇力</h1> |

css:

|  |
| --- |
| 1. <style type="text/css"> 2. **h1**{ 3. color:red; 4. } 5. </style> |

注意的是：

1） 所有的标签，都可以是选择器。比如ul、li、label、dt、dl、input

2） 无论这个标签藏的多深，一定能够被选择上：

3） 选择所有的标签，而不是一个。

标签选择器，选择的是页面上所有这种类型的标签，所以经常描述“共性”，无法描述某一个元素的“个性”的。

## id选择器

|  |
| --- |
| 1. <p>我是段落1</p> 2. <p **id="para2"**>我是段落2</p> 3. <p>我是段落3</p> |

css：

|  |
| --- |
| 1. <style type="text/css"> 2. **#**para2{ 3. color:red; 4. } 5. </style> |

id选择器的选择符是“#”。

任何的HTML标签都可以有id属性。表示这个标签的名字。

这个标签的名字，可以任取，但是：

1） 只能有字母、数字、下划线

2） 必须以字母开头

3） 不能和标签同名。比如id不能叫做body、img、a

一个HTML页面，不能出现相同的id，哪怕他们不是一个类型。比如页面上有一个id为pp的p，一个id为pp的div，是非法的！



一个标签可以被多个css选择器选择，共同作用，这就是“层叠式”的第一层含义。第二层含义，明天介绍。

## 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。

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。

# css高级选择器

## 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。

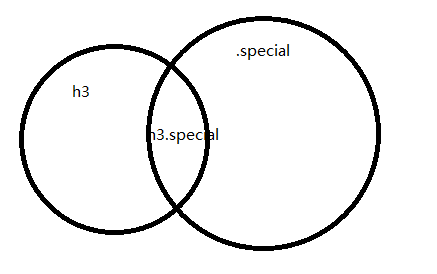
|  |
| --- |
| 1. .div1 .li2 p{ 2. color:red; 3. } |

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。

## 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。

## 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。

## 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。

# 一些CSS3选择器

## 兼容问题介绍

我们现在给大家介绍一下浏览器：

IE： 微软的浏览器，随着操作系统安装的。所以每个windows都有IE浏览器。

windows xp 操作系统安装的IE6

windows vista 操作系统安装的IE7

windows 7 操作系统安装的IE8

windows 8 操作系统安装的IE9

windows10 操作系统安装的edge

浏览器兼容问题，要出，就基本上就是出在IE6、7身上，这两个浏览器是非常低级的浏览器。

http://www1.pconline.com.cn/pcedu/specialtopic/css3-doraemon/

|  |  |  |  |
| --- | --- | --- | --- |
| Chrome浏览器： | Firefox浏览器： | IE8： | IE6： |

浏览器的市场占有率： <http://tongji.baidu.com/data/>

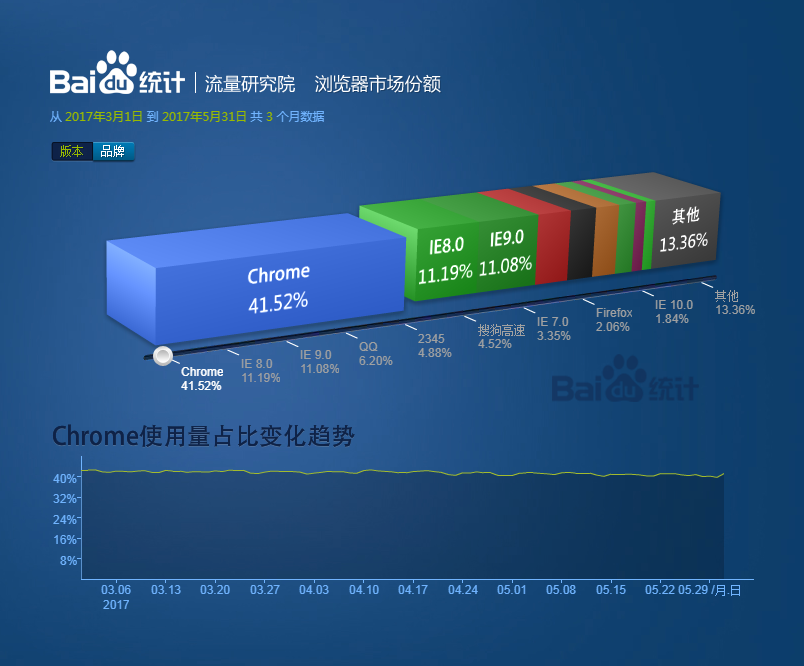
今天：

IE9 5.94%

IE8 21.19%

IE7 4.79%

IE6 4.11%



## 儿子选择器>

**IE7开始兼容，IE6不兼容。**

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

div的儿子p。和div的后代p的截然不同。

能够选择：

|  |
| --- |
| 1. <div> 2. <p>我是div的儿子</p> 3. </div> |

不能选择：

|  |
| --- |
| 1. <div> 2. <ul> 3. <li> 4. <p>我是div的重孙子</p> 5. </li> 6. </ul> 7. </div> |

## 序选择器

**IE8开始兼容；IE6、7都不兼容**

选择第1个li：

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

选择最后一个1i：

|  |
| --- |
| 1. ul li**:last-child**{ 2. color:blue; 3. } |

由于浏览器的更新需要过程，所以现在如果公司还要求兼容IE6、7，那么就要自己写类名：

|  |
| --- |
| 1. <ul> 2. <li **class="first"**>项目</li> 3. <li>项目</li> 4. <li>项目</li> 5. <li>项目</li> 6. <li>项目</li> 7. <li>项目</li> 8. <li>项目</li> 9. <li>项目</li> 10. <li>项目</li> 11. <li **class="last"**>项目</li> 12. </ul> |

用类选择器来选择第一个或者最后一个：

|  |
| --- |
| 1. ul li**.first**{ 2. color:red; 3. } 4. ul li**.last**{ 5. color:blue; 6. } |

## 下一个兄弟选择器

**IE7开始兼容，IE6不兼容。**

+表示选择下一个兄弟

|  |
| --- |
| 1. <style type="text/css"> 2. h3+p{ 3. color:red; 4. } 5. </style> |

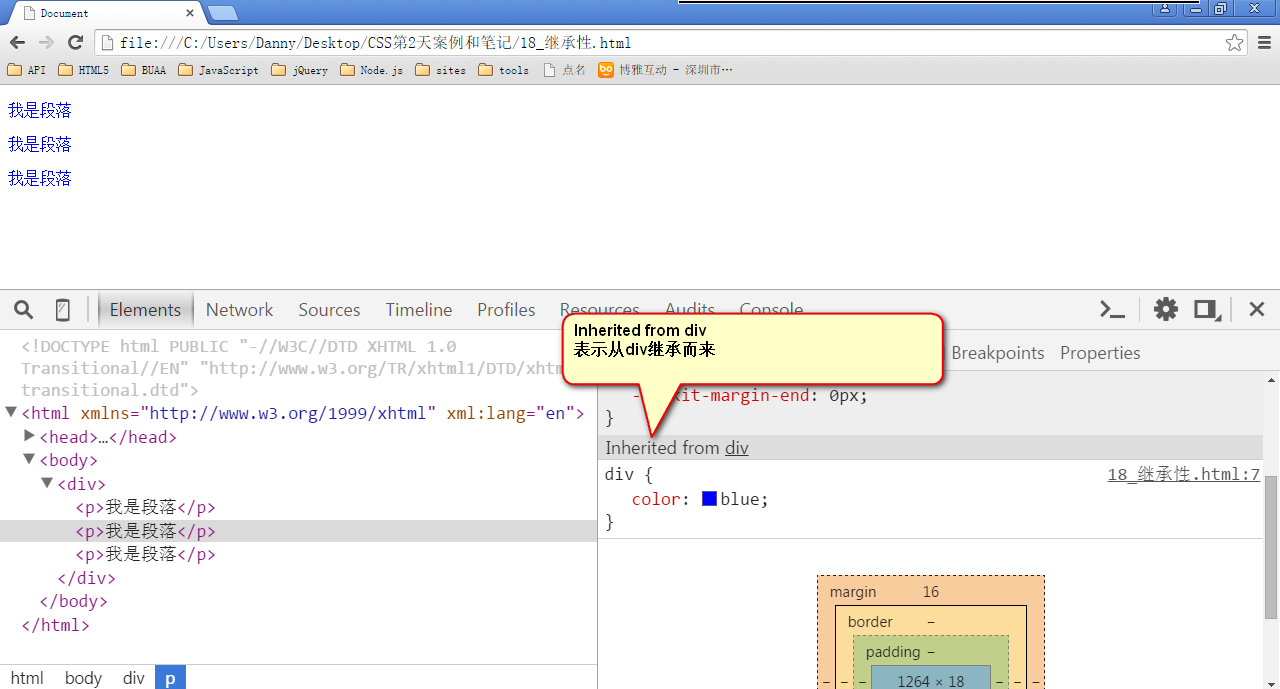
选择上的是h3元素后面紧挨着的第一个兄弟。

|  |
| --- |
| 1. <h3>我是一个标题</h3> 2. **<p>我是一个段落</p>** 3. <p>我是一个段落</p> 4. <p>我是一个段落</p> 5. <h3>我是一个标题</h3> 6. **<p>我是一个段落</p>** 7. <p>我是一个段落</p> 8. <p>我是一个段落</p> 9. <h3>我是一个标题</h3> 10. **<p>我是一个段落</p>** 11. <p>我是一个段落</p> 12. <p>我是一个段落</p> 13. <h3>我是一个标题</h3> |

所有的css3选择器，我们放在HTML5和CSS3课上介绍。

# CSS的继承性和层叠性

## 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. } |

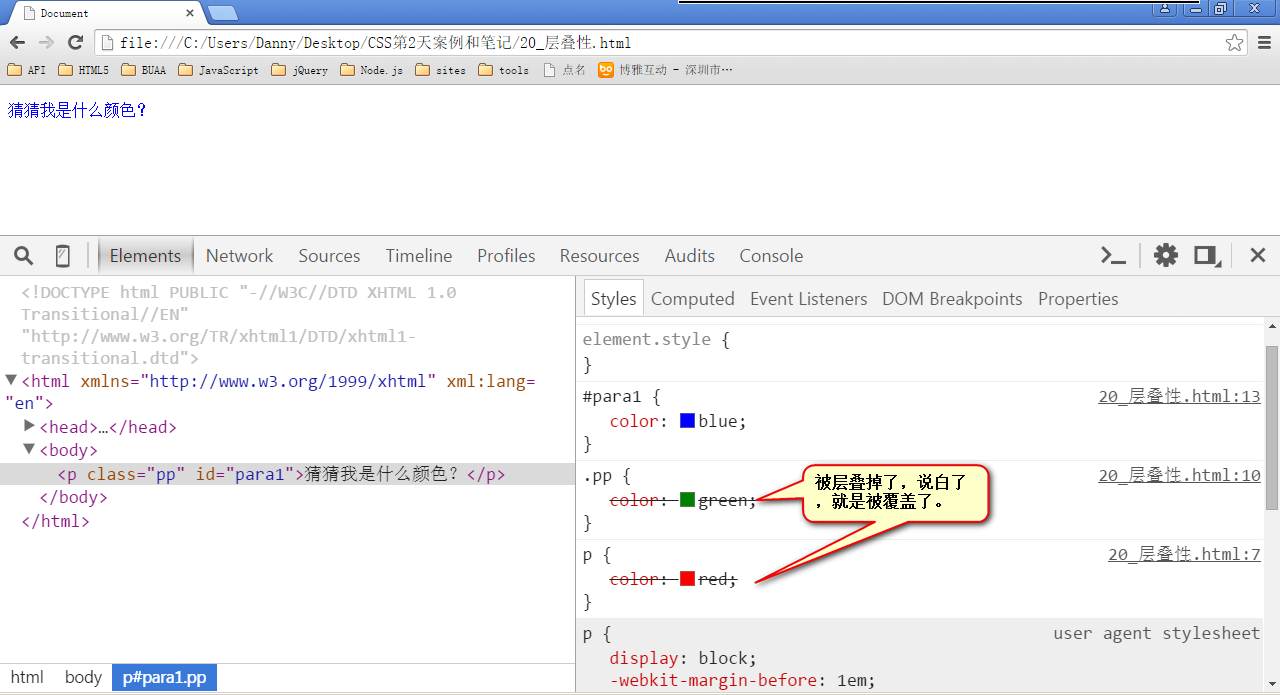
继承性是从自己开始，直到最小的元素。

## 层叠性

很多公司如果要笔试，那么一定会考层叠性。

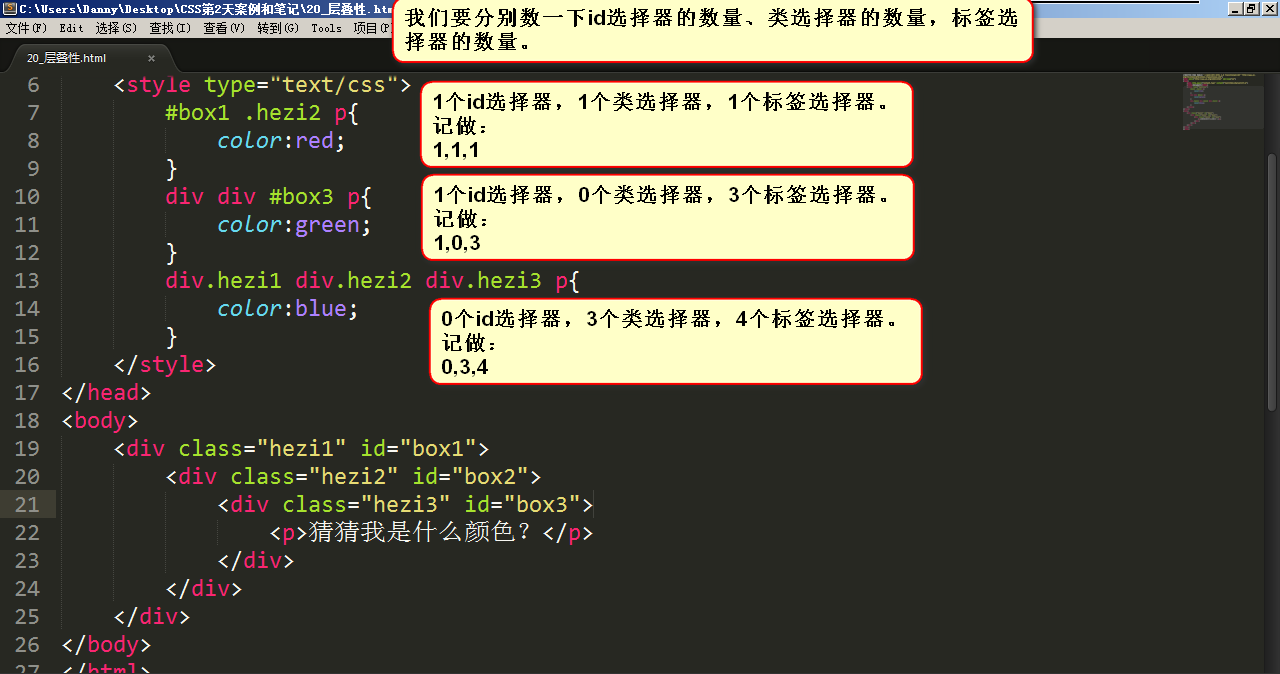
**层叠性：就是css处理冲突的能力。 所有的权重计算，没有任何兼容问题！**

**CSS像艺术家一样优雅，像工程师一样严谨。**

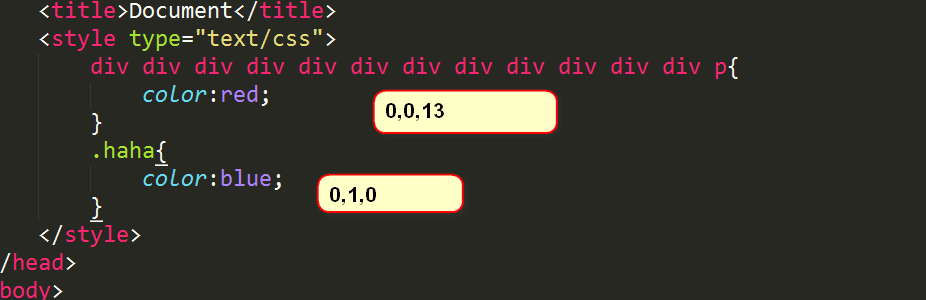


当选择器，选择上了某个元素的时候，那么要这么统计权重：

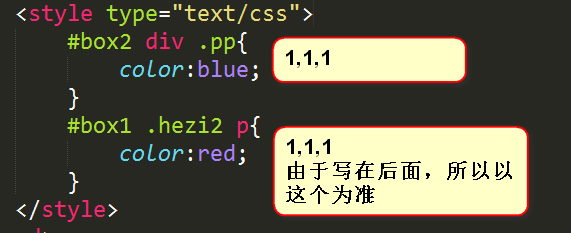
id的数量，类的数量，标签的数量



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。

复习一下：

下列都是IE6兼容的选择器：

p

#box

.spec

div.box

div .box

div , .box

\*

下列都是IE7开始兼容：

div>p

div+p

下列都是IE8开始兼容：

div p:first-child

div p:last-child

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？