# CSS入门和高级技巧

第1天课堂笔记（本课程共8天）

# 目录

[CSS入门和高级技巧 1](#_Toc16605)

[目录 2](#_Toc307)

[一、上节课知识的复习 3](#_Toc16566)

[二、表格 5](#_Toc10313)

[2.1 表格的基本使用 5](#_Toc18472)

[2.2 表格单元格的合并 5](#_Toc18932)

[三、div和span标签 9](#_Toc12485)

[3.1 div标签 9](#_Toc18023)

[3.2 span 10](#_Toc15155)

[四、HTML杂项 11](#_Toc16357)

[4.1 注释 11](#_Toc15358)

[4.2 字符实体 11](#_Toc20185)

[五、废弃标签介绍 13](#_Toc20419)

[六、CSS整体感知 14](#_Toc29637)

[6.1 简介 14](#_Toc23026)

[6.2 整体感知 15](#_Toc7553)

[6.3 style标签——css的舞台 16](#_Toc27265)

[6.4 css对换行不敏感、缩进不敏感 16](#_Toc3749)

[6.5 分号 17](#_Toc19006)

[6.6 先学几个属性 17](#_Toc787)

[七、基本选择器 18](#_Toc10914)

[7.1 标签选择器 18](#_Toc16122)

[7.2 id选择器 19](#_Toc7510)

[7.3 类选择器 20](#_Toc15050)

[八、高级选择器 23](#_Toc22727)

[8.1 后代选择器 23](#_Toc31729)

[8.2 交集选择器 24](#_Toc14977)

[8.3 并集选择器 24](#_Toc26714)

[8.4 通配符\* 25](#_Toc15650)

# 一、上节课知识的复习

● 一定要会用Chrome浏览器审查别人的网页，还有自己的项目，排错的功能，**快捷键是F12**。

2008年诞生的谷歌浏览器，一诞生就解决了原来前端工程师调试难的问题。

● 插入图片，页面中能够插入的图片类型：jpg、jpeg、bmp、png、gif；不能的psd、fw。

语法：

|  |
| --- |
| 1. <img src=”路径” alt=”替代文字” /> |

● 相对路径、绝对路径：

相对路径就是从html页面出发，找到图片：

|  |
| --- |
| 1. <img src=”a/b/1.jpg” /> |

等价于

|  |
| --- |
| 1. <img src=”**./**a/b/1.jpg” /> |

图片位于html文件较浅的文件夹中：

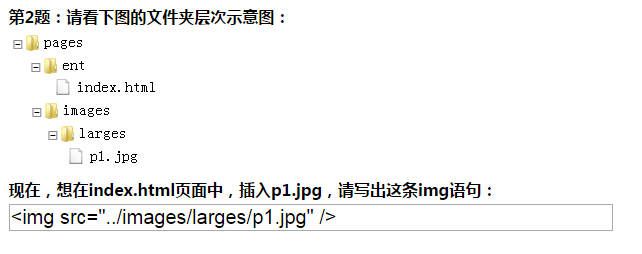
|  |
| --- |
| 1. <img src=”**../**../../1.jpg” /> |

绝对路径：直接的门牌号。以http://开头，或者以/开头的都是绝对路径。

|  |
| --- |
| 1. <img src=”**http://**www.163.com/images/logo.png” /> |

自己的服务器根目录：

|  |
| --- |
| 1. <img src=”**/**images/logo.png” /> |



● 超级链接

a标签，语法：

|  |
| --- |
| 1. <a href=”地址” title=”悬停文字” target=”\_blank”>链接的文字</a> |

超级链接还能做页面锚点

|  |
| --- |
| 1. <a name=”info”></a> |

此时就能用1.html#info来直接看见这个内容

● 表单

form标签里面，这个标签我们在Ajax课程上详细介绍。

input类型：

text、password、radio、checkbox、submit、button、reset

下拉框：

select、option

多行文本框：

textarea

单选框如果想互斥，必须要有相同的name属性。如果想点击文字，就能勾选，就要有label

|  |
| --- |
| 1. <input type=”radio” name=”sex” id=”male” /> 2. <label for=”male”>男</label> |

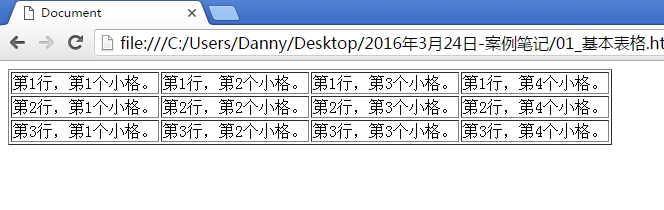
● 无序列表、有序列表、定义列表

注意嵌套的写法，ul的儿子只能是li，ul的孙子是谁无所谓：

|  |
| --- |
| 1. <ul> 2. <li> 3. <h3></h3> 4. <ul> 5. <li></li> 6. <li></li> 7. <li></li> 8. <li></li> 9. </ul> 10. </li> 11. <li></li> 12. <li></li> 13. <li></li> 14. </ul> |

# 二、表格

## 2.1 表格的基本使用



|  |
| --- |
| 1. <table border="1"> 2. <tr> 3. <td>第1行，第1个小格。</td> 4. <td>第1行，第2个小格。</td> 5. <td>第1行，第3个小格。</td> 6. <td>第1行，第4个小格。</td> 7. </tr> 8. <tr> 9. <td>第2行，第1个小格。</td> 10. <td>第2行，第2个小格。</td> 11. <td>第2行，第3个小格。</td> 12. <td>第2行，第4个小格。</td> 13. </tr> 14. <tr> 15. <td>第3行，第1个小格。</td> 16. <td>第3行，第2个小格。</td> 17. <td>第3行，第3个小格。</td> 18. <td>第3行，第4个小格。</td> 19. </tr> 20. </table> |

如果有表头的语义，那么应该书写th替换td。也就是说td和th都是小格格。

## 2.2 表格单元格的合并

单元格的合并，单元格的合并属性，colspan、rowspan。**一定要写在td或者th上，tr是没有这两个属性的**。

colspan我们叫做列跨度，rowspan行跨度。**这两个属性，都是小格格的属性，而不是“行”的属性。**

题目1：

|  |  |  |
| --- | --- | --- |
| 1 | 2 | |
| 3 | 4 | 5 |
| 6 | 7 |

|  |
| --- |
| 1. <table border="1"> 2. <tr> 3. <td>1</td> 4. <td colspan="2">2</td> 5. </tr> 6. <tr> 7. <td>3</td> 8. <td rowspan="2">4</td> 9. <td>5</td> 10. </tr> 11. <tr> 12. <td>6</td> 13. <td>7</td> 14. </tr> 15. </table> |

题目2：

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **2** | **3** | **4** |
| **5** | **6** | **7** | |
| **8** | **9** | |
| **10** |

题目3:

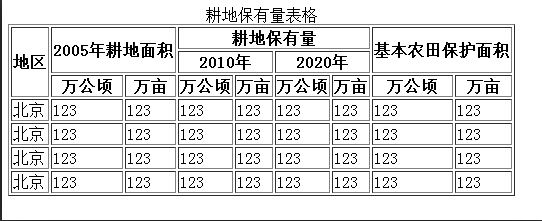
请看下面代码：

|  |
| --- |
| 1. <table> 2. <tr> 3. <td>1</td> 4. <td rowspan="2">2</td> 5. </tr> 6. <tr> 7. <td>3</td> 8. </tr> 9. </table> |

对应的图形是：选A

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A:   |  |  | | --- | --- | | 1 | 2 | | 3 | | B：   |  |  | | --- | --- | | 1 | 2 | | 3 | | C：   |  |  | | --- | --- | | 1 | | | 2 | 3 | | D：   |  |  | | --- | --- | | 1 | 2 | | 3 | | |

留的作业：



|  |
| --- |
| 1. <table border="1"> 2. <tr> 3. <th rowspan="3">地区</th> 4. <th rowspan="2" colspan="2">2005年耕地面积</th> 5. <th colspan="4">耕地保有量</th> 6. <th rowspan="2" colspan="2">基本农田保护面积</th> 7. </tr> 8. <tr> 9. <th colspan="2">2010年</th> 10. <th colspan="2">2020年</th> 11. </tr> 12. <tr> 13. <th>万公顷</th> 14. <th>万亩</th> 15. <th>万公顷</th> 16. <th>万亩</th> 17. <th>万公顷</th> 18. <th>万亩</th> 19. <th>万公顷</th> 20. <th>万亩</th> 21. </tr> 22. <tr> 23. <td>北京</td> 24. <td>123</td> 25. <td>123</td> 26. <td>123</td> 27. <td>123</td> 28. <td>123</td> 29. <td>123</td> 30. <td>123</td> 31. <td>123</td> 32. </tr> 33. <tr> 34. <td>北京</td> 35. <td>123</td> 36. <td>123</td> 37. <td>123</td> 38. <td>123</td> 39. <td>123</td> 40. <td>123</td> 41. <td>123</td> 42. <td>123</td> 43. </tr> 44. <tr> 45. <td>北京</td> 46. <td>123</td> 47. <td>123</td> 48. <td>123</td> 49. <td>123</td> 50. <td>123</td> 51. <td>123</td> 52. <td>123</td> 53. <td>123</td> 54. </tr> 55. <tr> 56. <td>北京</td> 57. <td>123</td> 58. <td>123</td> 59. <td>123</td> 60. <td>123</td> 61. <td>123</td> 62. <td>123</td> 63. <td>123</td> 64. <td>123</td> 65. </tr> 66. </table> |

一个完整的表格，是有caption、thead、tbody三个部件的：

thead就是表格头部体的意思，tbody就是表格内容体的意思，caption就是表格标题

|  |
| --- |
| 1. <table> 2. <caption>我是标题</caption> 3. <thead> 4. <tr> 5. <th></th> 6. <th></th> 7. <th></th> 8. </tr> 9. <tr> 10. <th></th> 11. <th></th> 12. <th></th> 13. </tr> 14. </thead> 16. <tbody> 17. <tr> 18. <td></td> 19. <td></td> 20. <td></td> 21. </tr> 22. <tr> 23. <td></td> 24. <td></td> 25. <td></td> 26. </tr> 27. </tbody> 28. </table> |

# 三、div和span标签

## 3.1 div标签

div是division分割的意思，在页面上表示容器，将相同语义的一组元素放在同一个div里面。

div是典型的容器级标签，可以包裹任何东西。

语义上来说，所有相同语义、相关的语义的元素，都要放到一个div里面。

|  |
| --- |
| 1. <div> 2. <h3>中国主要城市</h3> 3. <ul> 4. <li>北京</li> 5. <li>哈尔滨</li> 6. <li>吉林</li> 7. <li>长春</li> 8. </ul> 9. </div> 11. <div> 12. <h3>中国著名男演员</h3> 13. <ul> 14. <li>郭德纲</li> 15. <li>刘德华</li> 16. <li>邓超</li> 17. <li>胡歌</li> 18. </ul> 19. </div> |

div标签，浏览器没有任何的默认样式。

div这个东西能够把层次分的很清楚。

div是网页布局使用的最常用的标签，老师来进行一个演示。div+css制作网页。

**div负责描述页面的结构，css负责描述页面的样式。**

现在我们第一次提到三层的结构：

HTML 结构层 从语义的角度描述页面的结构

CSS 样式层 从审美的角度装饰页面

JS 行为层 从交互的角度增加运动效果

## 3.2 span

span表示跨度，就是文本级的div。

就是把一些语义相近、功能相同的文本标签，都包裹在一起。注意，直觉上觉得span比a大，比p小。

span也没有任何的默认样式。span单独使用没有什么意义，都是配合样式表使用的。

|  |
| --- |
| 1. <p> 2. 今天是CSS的第1天，每天一定要按时<span>完成作业</span>，要按时吃饭睡觉，要<span>多吃蔬菜</span>少喝酒，上课不要迟到。 3. </p> |

也就是说，div和span现在还看不出有什么用，感觉浏览器没有任何变化，必须等到css之后，你才能理解。

# 四、HTML杂项

## 4.1 注释

任何的语言都有注释，注释是**不会当做程序执行的，是给程序员自己看的**。

HTML中，注释的语法：

|  |
| --- |
| 1. **<!--**这是网页的头部**-->** |

换行是合法的：

|  |
| --- |
| 1. **<!--** 2. 这是网页的内容 3. 制作人杜承玖 4. **-->** |

很明显，写一些注释，对代码的清晰度、可读性有很大的提升。

webstrom中，按ctrl+/ ， 能够快速输入一个注释。

为了方便调试代码，经常会把一些代码注释掉，集中精力看其余的代码。

## 4.2 字符实体

也叫作转义字符。

|  |
| --- |
| 1. <div> 2. 好高兴啊，今天老师教会我们**&lt;**h1>是主标题的意思，我真开心呀！ 3. </div> |

也就是说，在写HTML页面的时候，经常会有误会，比如<就正常出现。那么为了防止HTML页面认为这是一个标签，所以就应该写成转义字符。

所有的转义字符，都是这样的格式：

|  |
| --- |
| 1. &字母; |

&lt; 就是< , lt就是英语less than小于

&gt; 就是> , gt就是英语greater than大于

&copy; 就是© 版权符号

&nbsp; 就是空格

手册



有一个非常重要的转义字符，就是nbsp。 **n**on-**b**reaking **sp**ace，无换行空格。

看代码：

|  |
| --- |
| 1. <ul> 2. <li>邓 超</li> 3. <li>孙 俪</li> 4. <li>王宝强</li> 5. </ul> |

两个空格，折叠成了一个空格



所以要用&nbsp;来表示一个中文的空格：

|  |
| --- |
| 1. <ul> 2. <li>邓**&nbsp;**超</li> 3. <li>孙**&nbsp;**俪</li> 4. <li>王宝强</li> 5. </ul> |

# 五、废弃标签介绍

你会发现，现在的HTML标签只负责语义、结构。不负责样式的。但是在2008之前，HTML也是负责样式的。

但是HTML中，有一些标签，具有浓郁的样式色彩。就是这个标签，不是负责语义的，而是负责样式的，2008年开始，坚决的废弃掉了这些标签，用CSS来替代他们的功能。

这些标签：

|  |
| --- |
| 1. <b></b> 2. <u></u> 3. <i></i> 4. <del></del> |

丝毫看不出语义，只有样式，所以是被废弃了！

这些标签，**我们还有点用，当做CSS钩子**，什么是钩子，以后再说！

还有一些标签，是擦边球，就是自己也有一些语义，但是也有一些样式色彩，这些标签是可以合理使用的：

|  |
| --- |
| 1. <p><strong>我是一个文字</strong></p> 2. <p><em>我是一个文字</em></p> |

strong是强壮的意思，表示要强调的语义；

em是emphasize的意思，也是强调的意思。

strong、em要比b、i好。

HTML在现在的社会，就是负责页面语义和结构的，所以标签极大的简化。页面上常用的标签实际上就几个：

容器： div、ul（li）、ol、dl、table

放文字的： h系列、p、span、a

文本流： img、input

当做css钩子： b、u、i

最后最后强调一下，HTML放什么标签，不是看什么样子，要看语义！

# 六、CSS整体感知

## 6.1 简介

前端开发3层：

HTML 结构层 语义、结构

CSS 样式层 样式

JS 行为层 交互、行为、动画

开始进军CSS了，非常好用，很有意思，很有成就感，负责页面的样式。

CSS的全程是**c**ascading **s**tyle **s**heet ， 层叠式样式表，“样式表”比较好理解，“层叠式”比较难理解，所以我们后面的课程，慢慢告诉你什么是层叠式。

CSS的维护者也是w3c，**最新版本是CSS3，但是浏览器没有那么的兼容，所以我们学习的是CSS2.1**。

我们后面也有专门的CSS3的课程，下图就是在Chrome和IE8中看机器猫：

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

|  |  |
| --- | --- |
|  |  |

## 6.2 整体感知

css分为两个部分，选择谁、设置什么样式。

|  |
| --- |
| 1. <style type="text/css"> 2. **h1{** 3. color:red; 4. background-color: blue; 5. font-size: 20px; 6. font-style: italic; 7. **}** 8. **p{** 9. color:green; 10. font-size: 40px; 11. text-shadow:2px 2px 2px black; 12. font-family: "微软雅黑"; 13. transform:rotate(30deg); 14. width: 300px; 15. **}** 16. </style> |

上面的h1、p就是选择器。表示选择哪些HTML标签给他们加样式，就是说你要给谁加样式。

然后大括号里面，就是k:v;的结构，把所有的样式列出来，拥有的样式就写上，没有的样式不写，有多少写多少：



|  |
| --- |
| 1. 选择器{ 2. k:v; 3. k:v; 4. k:v; 5. k:v; 6. k:v; 7. } 8. 选择器{ 9. k:v; 10. k:v; 11. k:v; 12. k:v; 13. k:v; 14. } |

所以我们的课程，就是这么进行，今天我们解决选择的问题，明天开始我们要学习大量的CSS属性。

css就这么简单，没有与或非、没有逻辑、没有加减乘除、没有函数、没有选择循环判断……

只有一条一条的罗列下来。

## 6.3 style标签——css的舞台

所有的css样式，都要写在style标签中（当然今后的课程你会看见其他形式的css，到时候再说）。

style标签写在**<head>**里面。

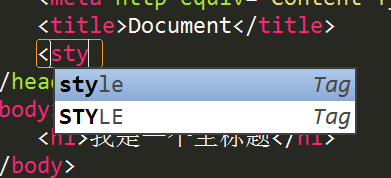
|  |
| --- |
| 1. <style type="text/css"> 2. </style> |

style就是样式的意思，表示这个标签里面写的是样式。

type是类型；

text/css表示纯文本的css。

傻瓜版的sublime能快速输入，只需要输入<sty 按tab就行了：



然后就在style标签里面，书写样式：

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

## 6.4 css对换行不敏感、缩进不敏感

|  |
| --- |
| 1. h1{ 2. color:red;background-color: blue;font-size:22px; 3. } |

等价于

|  |
| --- |
| 1. h1{ 2. color:red; 3. background-color: blue; 4. font-size:22px; 5. } |

等价于

|  |
| --- |
| 1. h1{color:red;background-color: blue;font-size:22px;} |

写成一行，css代码尺寸小了，这是最后压缩的事情，写的时候，还是一行一行写。

## 6.5 分号

每个选择器最后一项的属性值可以没有分号，其他都要有。

|  |
| --- |
| 1. h1{ 2. color:red**;** 3. background-color: blue**;** 4. font-size:22px 5. } |

我们一般也会把最后一项写上分号。

## 6.6 先学几个属性

今天下午主要讲解选择器，但是不讲解几个属性，咱们没法做试验，先草草学几个属性。

● 文字颜色

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

color就是文字颜色的意思，后面的值我们使用英语描述。我们后面的课程将告诉你，还能用rgb、十六进制来描述颜色，现在先不着急。

sublime中快捷键是c tab

● 背景颜色

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

background就是背景的意思，中间有个短横和color连接。我们css中多个词组都是短横分开的。

sublime中快捷键是bgc tab

● 字号

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

font就是字体，size就是尺寸。px是英语pixel像素的意思。

● 边框

|  |
| --- |
| 1. border:1px solid green; |

border就是边框的意思，**它的值很有意思，是三个部分，用空格隔开**。

这三个部分，可以顺序交换。

1px 就是粗细，可以任意调整；

solid就是实线，如果是虚线写dashed；

green就是颜色，可以任意设置。

# 七、基本选择器

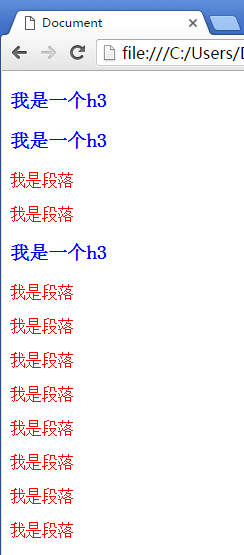
选择器就是怎么选择元素的

## 7.1 标签选择器

标签选择器是最最简单的选择器，就是选择页面上所有的同种标签。

要注意的是，选择的是页面上的所有这个标签，而不是一个。

|  |
| --- |
| 1. <style type="text/css"> 2. h3{ 3. color:blue; 4. } 5. p{ 6. color:red; 7. } 8. </style> |



即使它藏的很深，也能被选择器上。

|  |
| --- |
| 1. <ul> 2. <li> 3. <ul> 4. <li> 5. <p>我是很深的p</p> 6. …… |

body就是页面，如果要设置整个页面的背景颜色，要设置body。

|  |
| --- |
| 1. body{ 2. background-color:yellow; 3. } |

标签选择器有点不好用，要选择就都选择了。所以主要的用途就是设置一些标签的默认情况。

当我们给元素设置样式之后，我们应该会通过Chrome审查元素来查看：



## 7.2 id选择器

就是对某一个（注意就一个）元素，进行特别的样式设定的时候用。**就是给元素一个特别的名字，然后通过这个名字来选择它。**

就是给元素加上id属性，然后在选择器用#开头进行选择。

|  |
| --- |
| 1. <p **id="pp"**>我是一个段落</p> |

选择的时候写#：

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

首先，任何元素都可以设置id属性，**合法的id属性是英语字母开头（大小写敏感，也就是说aa和AA是两个不同的合法id）、可以有数字、下划线、短横。**

**同一个页面中，id绝对不能相同**。哪怕是不同的元素，id也不能相同，**我们必须保证，id在页面上是唯一的**。

下面这个例子是错误的，初学者常见的错误：

|  |
| --- |
| 1. <p id="para">我是一个段落</p> 2. <h3 id="para">我是一个h3</h3> |

页面上已经有元素的id是para了，所以不能有元素仍然叫para。即使之前叫做para的是一个p，后面叫做para的是一个h3，也不行！是不合法的！

你会发现，我们可以综合利用标签选择器、id选择器一起做效果。比如，我们让所有的p的字号都是20px，让其中某一个的颜色是红色。**这就是层叠式样式表的第一层含义，同一个标签可以有多个选择器选择它。**



最后强调一下，选择器写的时候，#一定要紧紧跟在id前面：

|  |
| --- |
| 1. # para1{  **← 错误的写法！！应该是#para1， 没有空格** 2. color:red; 3. } |

## 7.3 类选择器

id有点不好用，因为是唯一的，如果页面上有两个、三个元素，想设置为同样的样式，id就无能为力了。

所以就是用类选择器，**就是给某一类元素，设置相同的类名，然后通过这个类名来选择**。

class叫做类名。

|  |
| --- |
| 1. <p **class="warning"**>我是段落</p> 2. <p **class="warning"**>我是段落</p> |

选择的时候,用点：

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

类名的命名和id是一样的，英语字母开头（aa和AA是不一样的类名）、数字、下划线、短横。

可以有多个标签携带同一个类名：

|  |
| --- |
| 1. <p **class="warning"**>我是段落</p> 2. <p **class="warning"**>我是段落</p> 3. <p>我是段落</p> 4. <p>我是段落</p> 5. <h3>我是一个h3</h3> 6. <h3 **class="warning"**>我是一个h3</h3> 7. <p>我是段落</p> 8. <p>我是段落</p> 9. <h3>我是一个h3</h3> 10. <h3 **class="warning"**>我是一个h3</h3> |

**同一个标签，可以携带多个类名，用空格隔开**。

|  |
| --- |
| 1. <p class="**warning xian**">我是一个段落</p> |

这个p标签同时携带了两个类，所以.warning选择器和.xian选择器能同时选择上他们。

|  |
| --- |
| 1. .warning{ 2. color:red; 3. } 4. .xian{ 5. text-decoration: underline; 6. } |

注意，使用空格隔开，不是两个class

|  |
| --- |
| 1. <p ~~class=”warning” class=”xian”~~>我是一个错误的示范</p> |

原子类的意思： 比如来制作这个效果：

我是段落

我是段落

我是段落

用原子类是最方便的，所谓的原子类，就是一些简单的属性做成一个类，然后元素自行选择：

|  |
| --- |
| 1. <style type="text/css"> 2. .hong{ 3. color:red; 4. } 5. .da{ 6. font-size: 50px; 7. } 8. .xian{ 9. text-decoration: underline; 10. } 11. </style> |

各取所需：

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

类名的使用的时候，尽量用有语义的单词来表示：warning、important、ad、main。

不要用一些具体的单词：red、hong、left-part、right-part。

写页面到底用id还是class呢？答案非常肯定：class。哪怕页面上只有某一个要改样式，那么我尽量给这一个元素一个单独的class，也不给他id。

这是因为id是js常用的标签，如果你在css就把id用掉了，js程序就有耦合性的问题。

**class上样式，id上行为。**

# 八、高级选择器

## 8.1 后代选择器

后代选择器用空格表示：

|  |
| --- |
| 1. <div class="zhengzhi"> 2. <p>我是一些政治新闻</p> 3. <p>我是一些政治新闻</p> 4. <p>我是一些政治新闻</p> 5. <p>我是一些政治新闻</p> 6. </div> |

选择这些p，可以用后代选择器：

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

我们现在要详细介绍一下：

● **选择的是后代的元素，不一定是儿子，如果是孙子、曾孙子、重孙子……都行**

|  |
| --- |
| 1. <div class="div1"> 2. <div class="div2"> 3. <div class="div3"> 4. <p>我是一个小p</p> 5. </div> 6. </div> 7. </div> |

选择的是div1这个div中的后代p：

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

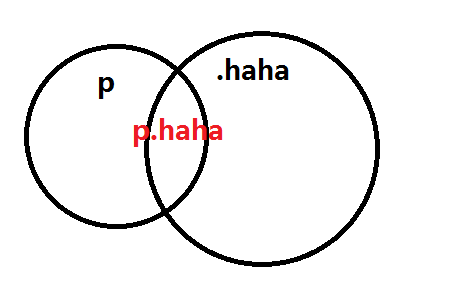
● 可以多重描述祖先顺序，但是顺序必须真实存在：

|  |
| --- |
| 1. <style type="text/css"> 2. **.div1 .div3 p**{ 3. color:red; 4. } 5. </style> |

现在反应要快，看见空格了，就是后代。

## 8.2 交集选择器

选择又是p标签，同时又有haha类的元素：



|  |
| --- |
| 1. p.haha |

一般来说，语法是：

|  |
| --- |
| 1. 标签名.类名 |

别的交集选择器也合法，但是没人这么写：

|  |
| --- |
| 1. #box.haha |

|  |
| --- |
| 1. p.haha.xixi |

注意

|  |
| --- |
| 1. div.box |

和

|  |
| --- |
| 1. div .box |

## 8.3 并集选择器

逗号表示并集，“都”

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

等价于：

|  |
| --- |
| 1. p.haha{ 2. color:red; 3. } 4. div.xixi{ 5. color:red; 6. } |

## 8.4 通配符\*

选择所有元素，用\*

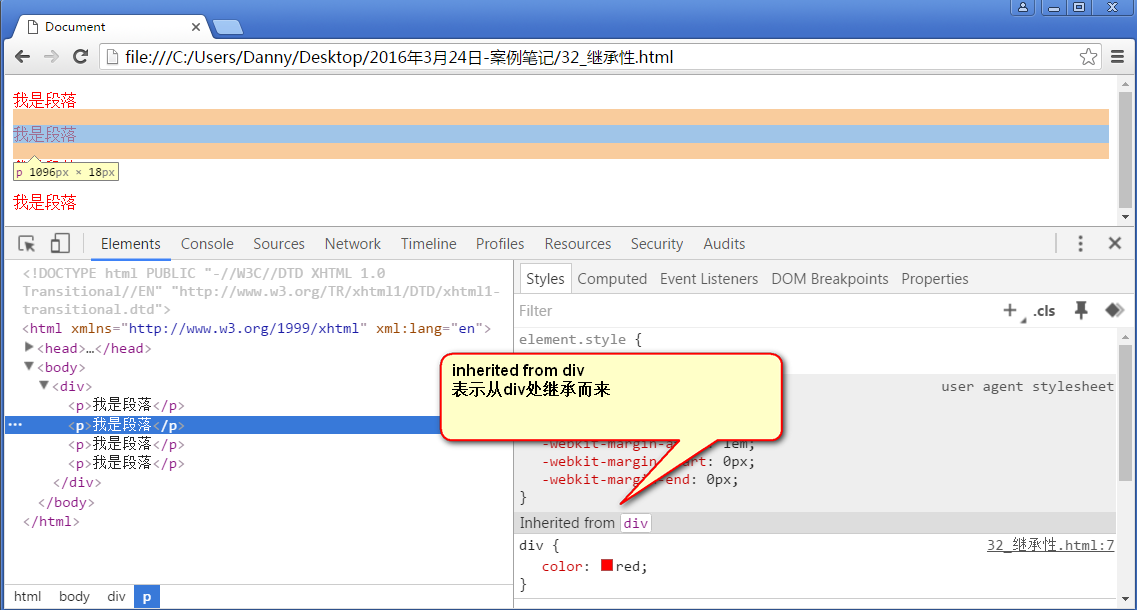
|  |
| --- |
| 1. <style type="text/css"> 2. \*{ 3. color:red; 4. } 5. </style> |

# 九、继承性和层叠性

这里的面试题相当多，相当有逻辑性和挑战性。

## 9.1 继承性

**css规定，有一些属性，给某一个元素设置了，它的后代元素都同样拥有它的这个属性了。**



哪些属性能够继承呢？

color

text-系列 ： 比如text-decoration:underline;

font-系列： 比如font-size:30px;

line-系列

background-color不能继承！border不能继承！等等

继承性是好事儿还是坏事儿？是天大的好事儿！！省事儿！

能够把一些初始的、基本的、普遍的设置，写在body里面。

|  |
| --- |
| 1. body{ 2. font-size: 15px; 3. } |

“层叠式”样式表，cascading的第二层含义，就是继承性。

## 9.2 层叠性

今晚学习吃力，先不看，明天重新讲。

**层叠性是什么？就是处理冲突的能力。就是当多个选择器都选择上了同一个标签，听谁的？**

同一个标签：

|  |
| --- |
| 1. <p class="haha" id="para">我是段落</p> |

现在有三个选择器都能够选择上它：

|  |
| --- |
| 1. <style type="text/css"> 2. p{ 3. color:red; 4. } 5. #para{ 6. color:green; 7. } 8. .haha{ 9. color:blue; 10. } 11. </style> |

听谁的？此时，就是所谓的层叠性发挥作用了，CSS有着一套非常严密的法律，此时听id的。

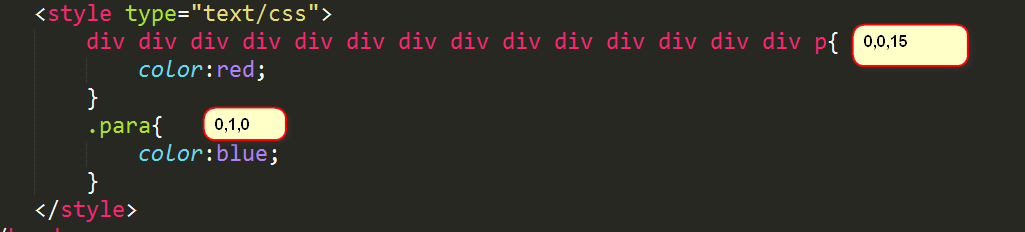


**选择器的一个基本权重： id的权重 > 类选择器 > 标签选择器**

比较复杂的权重计算：



10个便签，也干不过1个类。



只有选中到文字所在的标签，才能计算权重！！如果是继承而来的，权重是0！



如果权重相同怎么办？

● 如果是都选择上了，以style列表中后出现的选择器为准



● 都是通过继承而来的呢：就近原则，谁描述的近，听谁的。



一样近，听谁的？

