# 复习

列表：

无序列表：ul>li

有序列表：ol>li

自定义列表：dl>dt+dd

表格：

table>caption +( thead >tr >th)+ (tbody>tr>td)

合并单元格：td(th)

rowspan: 跨行合并

colspan: 跨列合并

属性值：数字，不用书写单位

表单元素：

form：功能性标签 action method属性

input标签 一个type属性更改属性值，设置不同类型的表单元素

单行文本框：text

密码框：password

单选框：radio 同一组单元框必须设置相同的name属性值

复选框：checkbox 同一组复选框必须设置相同的name属性值

普通按钮：button

重置按钮：reset

提交按钮：submit

label:对input表单进行标注

textarea: 文本域

select>option下拉列表

# css基础知识

## css概述

css:层叠式样式表（cascading style sheet）关键词：层叠式，样式

web前端三层：

结构层：从html角度描述页面结构

样式层：从美化角度设置样式

行为层：从交互角度描述页面的行为

html，css维护，更改由W3c组织。

目前学习css2.1 最新版本是css3.

## css作用

css组成：选择器（设置样式的标签），样式属性（给选择器设置样式）。

样式属性：逐条书写

|  |
| --- |
| 1. div { 2. /\*样式属性\*/ 3. width: 200px; 4. height: 300px; 5. background-color: lightblue; 6. } |

css:作用

给盒子设置样式

给文字设置样式

|  |
| --- |
| 1. div { 2. /\*逐条书写\*/ 3. /\*盒子样式属性\*/ 4. width: 200px; 5. height: 300px; 6. background-color: lightblue; 7. /\*文字属性\*/ 8. color: red; 9. } |

## 一些属性

盒子样式属性：

width: 盒子宽度，属性值数值+px(不能省略)

height: 盒子高度

background:背景，复合属性，其中每一个单一属性可以单独设置，使用短横连接

background-color:背景色

文字样式属性：

color: 文字颜色

font复合属性 单一属性font-weight: 设置文字是否加粗

font-weight: bold;文字加粗

font-weight: normal;文字不加粗

font-size:字号 属性值数值+px

## css书写位置

行内式：直接书写在标签一个style属性值中，该属性值包含了css所有的样式属性。

|  |
| --- |
| 1. <div style="color: red;width: 300px;">盒子</div> |

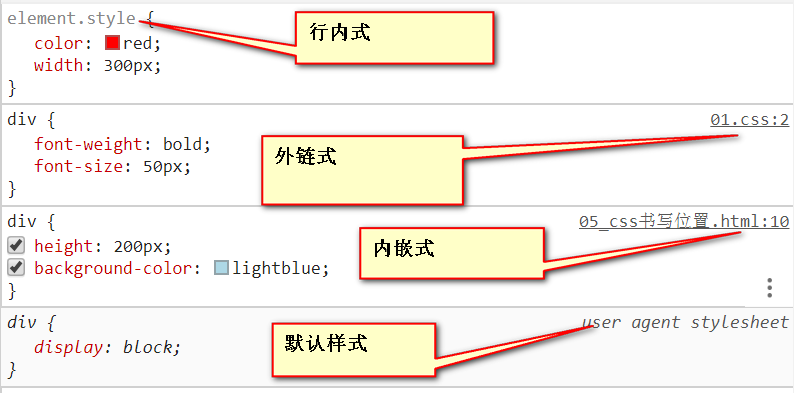
内嵌式：书写在head标签中一对style标签内部

|  |
| --- |
| 1. <head> 2. <meta http-equiv="Content-Type" content="text/html;charset=UTF-8"> 3. <title>Document</title> 4. <!-- type属性 表示纯文本css文件 --> 5. <style type="text/css"> 6. /\*选择器和样式属性\*/ 7. /\*内嵌式\*/ 8. div { 9. height: 200px; 10. background-color: lightblue; 11. } 12. </style> 13. </head> |

外链式：在head标签内部通过link标签引入一个外部的css文件

css文件的扩展名.css 选择器和样式属性直接书写在css文件中，不用再次书写style标签

|  |
| --- |
| 1. <!-- 外链式 通过link标签引入css文件 --> 2. <!-- rel:表示css和html关系 type属性 href表示引入css路径：相对路径表示从html出发查找css--> 3. <link rel="stylesheet" type="text/css" href="css/01.css"> |



## css杂项

注释：ctrl + /

|  |
| --- |
| 1. /\* 注释语句 \*/ |

键值对：k属性 v属性值 k: v

css分号重要性：每一条语句结束必须添加分号，表示语句结束。最后一条样式属性也要书写分号

css同样对空格，缩进，换行都不敏感。

# 基础选择器

## 2.1 标签选择器

通过标签名进行选中

写法： 直接书写标签名

特点：能够选中页面上所有的同种标签，不管标签嵌套多深都可以选中

|  |
| --- |
| 1. p {} |

应用：清除标签默认样式

|  |
| --- |
| 1. /\*标签选择器应用： 清除默认样式\*/ 2. ul { 3. /\*去掉小圆点\*/ 4. list-style: none; 5. } 6. a { 7. /\*去掉下划线\*/ 8. text-decoration: none; 9. color: #000000; 10. } |

## 2.2 id选择器

通过标签id属性进行选中。

写法：#id属性值

命名id属性值命名规则：开头必须是字母（严格区域大小写），后面可以有数字，-短横，下划线\_

id具有唯一性，id属性值只能出现一次，即使是不同的标签也不能重名。

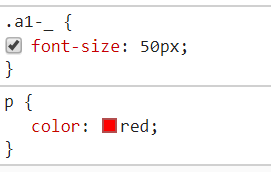
id选择器一般不设置样式，留给后面js使用。

|  |
| --- |
| 1. #para { 2. color: red; 3. } 4. #teshu1-\_ { 5. font-size: 50px; 6. } |

## 2.3 类选择器

通过标签class属性选中页面上所有同种类名元素。

写法： .类名



对于p元素，同时被标签选择器选中，设置颜色；也被类名选中设置不同属性字号，这两个样式属性都可以同时渲染在元素上。这就是css层叠性

层叠性体现1：同一个元素被不同选择器选中，设置不同的样式属性，都可以渲染在元素上。

类名在使用有一个特殊应用，原子类。一般原子类只一个样式属性便于设置。

|  |
| --- |
| 1. .red {color:red;} 2. .green {color:green;} 3. .blue {color:blue;} 4. .pink {color:pink;} |

同一个元素可以设置多个类名，多个类名用空格隔开

|  |
| --- |
| 1. class=”xiaoming xiaogang” |

## 2.4 通配符

能够选中页面上所有的元素，包括Body

\* {}

应用：快速清除默认样式

|  |
| --- |
| 1. \* { 2. font-size: 14px; 3. color: red; 4. } 5. /\*使用快速清除默认样式\*/ 6. \* { 7. padding: 0; 8. margin: 0; 9. } 10. .box { 11. margin-bottom: 20px; 12. } |

# 高级选择器

## 后代选择器

通过有嵌套关系的标签选择器或者类选择器确定我们选择范围，在大范围内再次选择

每一层选择器用空格隔开。

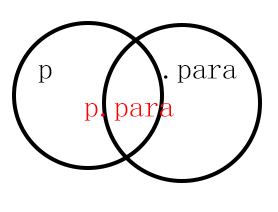
后代选择器不但可以选中儿子，或者是孙子等所有的后代元素都可以。

|  |
| --- |
| 1. div .box p { 2. background-color: lightblue; //该选择器不能选中下面结构 3. } 4. <div class="box"> 5. <div> 6. <p>我是box孙子</p> 7. </div> 8. </div> |

## 交集选择器

同一个元素具有多个选择器。

写法：连续书写选择器，之间不要有任何空格。



|  |
| --- |
| 1. /\*同一个元素同时具有多个选择器 一定要后代区分\*/ 2. /\*对于p 既有p标签又叫teshu\*/ 3. p.teshu { 4. background-color: lightblue; 5. } 6. <p>p元素</p> 7. <p class="teshu">p元素</p> 8. <p>p元素</p> 9. <h3>元素</h3> 10. <h3 class="teshu">元素</h3> 11. <h3>元素</h3> |

交集选择器可以连续交集

|  |
| --- |
| 1. p.para.teshu { 2. width: 300px; 3. height: 300px; 4. } |

|  |
| --- |
| 1. .para.teshu {} //合法，可以不书写标签名 2. #box.teshu {} //不合法，id具有唯一性 |

## 并集选择器

同时选中多个选择器，设置相同的样式。

写法：每个选择器使用逗号隔开

|  |
| --- |
| 1. /\*简化书写 并集选择器\*/ 2. div, h3, p { 3. color: red; 4. } |

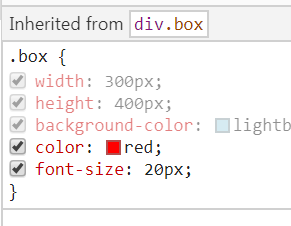
并集选择器也可以使用类名

|  |
| --- |
| 1. /\*类名也可以使用并集\*/ 2. .para, .box, .teshu { 3. width: 300px; 4. height: 300px; 5. background-color: lightblue; 6. } |

# 四、继承性和层叠性

## 4.1 继承性

有些属性给祖先设置，该祖先的后代元素都将继承这些样式属性。



文字属性可以从祖先继承（color, font, line, text等）

盒子属性不能从祖先继承（width,height,background,border,float,position等）

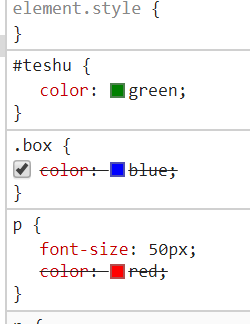
继承性也体现层叠性。

继承性是css比较好的性质，可以将文字属性给祖先设置，后代都可以继承，简化代码。

|  |
| --- |
| 1. /\*设置body文字属性\*/ 2. body { 3. color: lightblue; 4. font-size: 14px; 5. font-family: "Microsoft Yahei"; 6. } |

## 4.2 层叠性

同一个元素被不同选择器选中，设置相同的样式属性，最终听谁？



层叠性体现3： 同一个元素被不同选择器选中，同时设置相同的样式属性，权重大层叠掉权重小的。

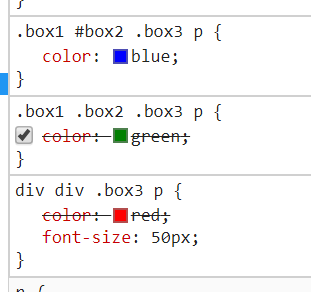
权重：选择器针对性（针对性越强权重越大）

id选择器 >> 类选择器 >>标签选择器

选中元素权重比较：

依次比较id选择器，类选择器，标签选择器数量；

先比较id选择器的数量，id数量多的权重最大；如果id数量相同，比较类的数量，类的数量多的权重就大；如果类的数量还相同就比较标签的数量，标签数量多的权重最大；



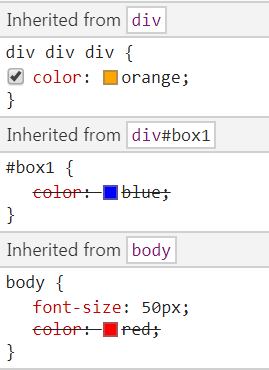
如果标签选择器数量也相同，选择器书写后面的层叠掉前面的相同样式属性。

|  |
| --- |
| 1. #box1 div .box3 p { 2. color: orange; 3. } 4. .box1 div #box3 p { 5. color: red; 6. font-size: 50px; 7. } 8. div .box2 #box3 p { 9. color: blue; 10. } 11. /\* 12. p元素被选中，比较权重 13. （1,1,2） 14. （1,1,2） 15. （1,1,2） 16. 样式书写在后面的层叠掉前面的。 17. 蓝色书写在最后权重最大。 18. \*/ |

元素没有被选中，样式看继承性。

1 元素没有被选中，直接看继承性。看选择器选中元素距离我们样式的元素之间的距离（html结构的远近），直接继承选择器距离最近的样式（就近原则：html结构）

|  |
| --- |
| 1. <div id="box1"> 2. <div class="box2"> 3. <div> 4. <p>颜色？？？</p> 5. </div> 6. </div> 7. </div> |



2 选择器距离相同，看选择器的权重。权重大的继承。

3.距离相同，权重还相同，看css书写顺序。谁写在后面继承谁的属性。

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |