

目录

目录

关于直播问题

昨日回顾

学习目标

CSS复合选择器

后代选择器(常用,重点)

子元素选择器

兄弟选择器(相邻元素选择器)

并集选择器(分组选择器)

交集选择器(了解)

复合选择器总结

复合选择器练习,按已有的HTML结构使用符合选择器设置样式

伪类选择器 重点

链接伪类选择器

超链接的美化 重点

练习-超级连接美化

结构伪类选择器 CSS3 重点 掌握

列表样式属性

元素的显示方式

定义

元素显示方式的分类

元素显示方式的转换(display属性) 要记

练习-元素的显示方式

练习-简单导航案例

元素的显示方式小结

CSS三大特性

CSS层叠性

CSS继承性

CSS优先级

CSS三大特性小结

!important属性

什么是important

!important在CSS中的作用

!important语法格式

使用!important一定要注意以下几点

CSS背景属性(background) 重点

背景颜色(background-color)

背景图片(background-image)

背景平铺(background-repeat)

背景定位(background-position) 重点,难点

背景定位的应用-精灵图(雪碧图)

今日总结

今日作业

使用xmind总结当天知识点

关于直播问题

课时 7 : 《CSS》	直播结束
课时 8 : 《CSS》	直播结束
课时 9 : CSS复合选择器	02-10 09:00
课时 10 : CSS三大特性	02-10 14:00
课时 11 : CSS边框,内填充	02-11 09:00
课时 12 : CSS外边距	02-11 14:00
课时 13 : CSS浮动(上)	02-12 09:00
课时 14 : CSS浮动(下)	02-12 14:00
课时 15 : CSS定位(上)	02-13 09:00
课时 16 : CSS定位(下)	02-13 14:00
课时 17 : CSS高级技巧(上)	02-14 09:00
课时 18 : CSS高级技巧(下)	02-14 14:00
课时 19 : CSS项目实战(上)	02-15 09:00

找到当天上课时间的课时

3/show

昨日回顾

CSS第一天知识总结



学习目标

- 能够写出后代选择器的语法并解释其作用
- 能够写出子元素选择器的语法并解释其作用

- 能够写出相邻元素选择器的语法并解释其作用
- 能够写出分组选择器的语法并解释其作用
- 能够理解并使用伪类选择器
- 能够使用列表属性
- 能够理解CSS属性的层叠性、继承性和优先级
- 能够灵活运用背景属性

学习目标

- 能够写出后代选择器的语法并解释其作用
- 能够写出子元素选择器的语法并解释其作用
- 能够写出相邻元素选择器的语法并解释其作用
- 能够写出分组选择器的语法并解释其作用
- 能够理解并使用伪类选择器
- 能够使用列表属性
- 能够理解CSS属性的层叠性、继承性和优先级
- 能够灵活运用背景属性

上次课我们学习的是基本选择器，
基本选择器:标签选择器,类选择器,
多类名选择器,id选择器,通配符选择器

今天我们来学习其他选择器

这些后代,子元素,相邻,分组等选择器
都属于后代选择器

伪类选择器我们主要用于设置a链接的美化

列表属性可以列表标签ul或者li设置 list-style-type

list-style-position list-style-image...

CSS三大特性

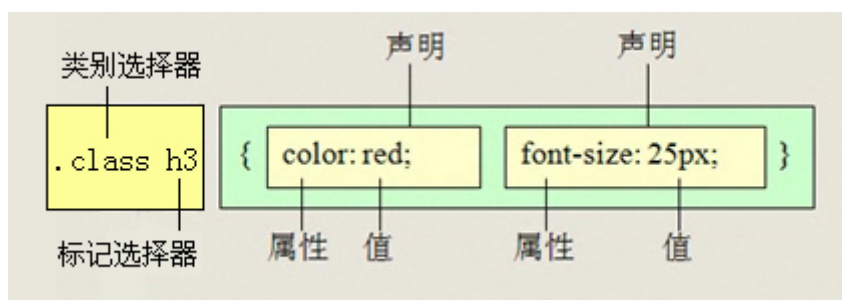
背景颜色,背景图片,背景平铺,背景定位...

CSS复合选择器

复合选择器是由两个或多个**基础选择器**，通过不同的方式组合而成的，目的是为了可以选择更准确更精细的目标元素标签。

后代选择器(常用,重点)

后代选择器又称为包含(嵌套)选择器，用来选择元素或元素组的后代，其写法就是把外层标签写在前面，内层标签写在后面，中间用**空格**分隔。当标签发生嵌套时，内层标签就成为外层标签的后代。



子孙后代都可以这么选择。或者说，它能选择任何包含在内的标签。

选择器名:

```
1 选择器1 选择器2 选择器3...{
2  属性1:属性值1;
3  属性2:属性值2;
4  属性3:属性值3;
5  ...
6 }
```

用法:

找到选择器1下的选择器2下的选择器3...对应的标签!

举例:多个无序列表与多个a标签在不同的父元素下面

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*后代选择器用于选中子孙后代的所有元素*/
8
9     /*
10      后代选择器名,选择器之间使用"空格"隔开
11
12      选择器1 选择器2{
13
14      }
15
16      选中选择器1中所有的选择器2对应的标签
17      */
18
19      /*标签选择器*/
20      /* li{
21        color:red;
22      } */
23
24      /*后代选择器,我们选中ul下面所有li标签*/
25      ul li{
26        color:blue;
27      }
28
```

```
29  /*后代选择器选中的是子孙后代,也就是满足条件所有的后代元素,也就是你只要是它的后
    代,都会被选中*/
30  /* div p{
31  color:orange;
32  } */
33
34  div span{
35  color:green;
36  }
37
38  /* div a{
39  color:orange;
40  } */
41
42
43  /*选中.lishushu里面所有a标签*/
44  .lishushu a{
45  color:skyblue;
46  }
47  /*选中.laowang里面所有a标签*/
48  .laowang a{
49  color:pink;
50  }
51  </style>
52  </head>
53  <body>
54  <ul>
55  <li>苹果</li>
56  <li>香蕉</li>
57  <li>葡萄</li>
58  </ul>
59
60  <ul>
61  <li>百度</li>
62  <li>京东</li>
63  <li>腾讯</li>
64  </ul>
65
66  <ol>
67  <li>腾讯视频</li>
68  <li>爱奇艺</li>
```

```
69 <li>芒果TV</li>
70 </ol>
71
72 <div>
73 <p>div中的p标签</p>
74 <span>div中的span标签</span>
75 </div>
76
77 <div>
78 <h1>
79 <p>
80 div中h1中的p标签
81 <span>我是孙子的儿子</span>
82 </p>
83 </h1>
84 </div>
85
86 <div class="lishushu">
87 <a href="#">张三</a>
88 <a href="#">李四</a>
89 <a href="#">王五</a>
90 </div>
91
92 <div class="laowang">
93 <a href="#">张三</a>
94 <a href="#">李四</a>
95 <a href="#">王五</a>
96 </div>
97 </body>
98 </html>
```


- 苹果
- 香蕉
- 葡萄

- 百度
- 京东
- 腾讯

1. 腾讯视频
2. 爱奇艺
3. 芒果TV

div中的p标签

div中的span标签

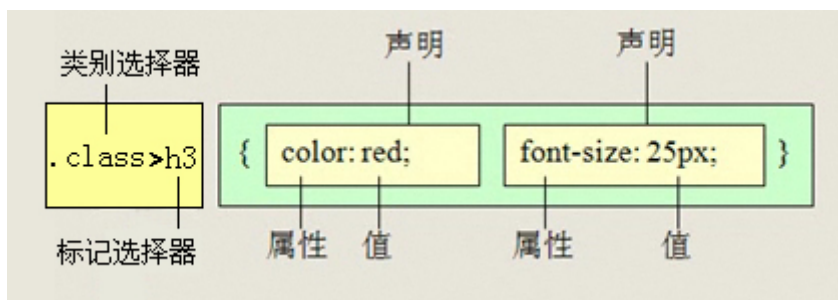
div中h1中的p标签 我是孙子的儿子

张三 李四 王五
张三 李四 王五

小结:后代选择器选择子孙后代!!!

子元素选择器

子元素选择器只能选择作为**某元素子元素**的元素。其写法就是把父级标签写在前，子级标签写在后面，中间跟一个 **>** 进行连接，**注意，符号左右两侧可以各保留一个空格,也可以不要空格**



注意: 这里的子 指的是 亲儿子 不包含孙子 重孙子之类。

选择器名:

```
1 选择器1 > 选择器2{
2  属性1:属性值1;
3  属性2:属性值2;
4  属性3:属性值3;
5  ...
6 }
7
8 选择器1>选择器2{
9  属性1:属性值1;
10 属性2:属性值2;
11 属性3:属性值3;
12 ...
13 }
```

用法:

找到选择器1下的选择器2对应的亲儿子标签!

举例: 只能选中div下的strong标签,而且是要第一代,也就是儿子元素

```
<div>
  <strong>儿子</strong>
  <strong>儿子</strong>
</div>

<div>
  <p>
    <strong>孙子</strong>
    <strong>孙子</strong>
  </p>
</div>
```

代码:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*子元素选择器只选择儿子代元素*/
8
9     /*
```

```
10 子元素选择器名
11
12 选择器1>选择器2{
13
14 }
15
16 能选择器1中选择器2对应的标签,但是这个选择器2必须是选择器1的亲儿子
17 */
18
19 /*后代选择器,选中所有的子孙后代*/
20 /* div strong{
21 color:red;
22 } */
23
24 /*子元素选择器 只选中亲儿子*/
25 div>strong{
26 color:blue;
27 }
28 </style>
29 </head>
30 <body>
31 <div>
32 <strong>儿子</strong>
33 <strong>儿子</strong>
34 </div>
35
36 <div>
37 <p>
38 <strong>孙子</strong>
39 <strong>孙子</strong>
40 </p>
41 </div>
42 </body>
43 </html>
```

儿子 儿子

孙子 孙子

小结:子选择器只选亲儿子!!!

兄弟选择器(相邻元素选择器)

选择器名:

```
1 选择器1+选择器2+.....{  
2  属性1:属性值1;  
3  属性2:属性值2;  
4  属性3:属性值3;  
5  ...  
6 }
```

用法:

兄弟选择器只能选择选择器1相邻下一个的选择器2兄弟标签

举例:

```
<p>p段落</p>  
<p>p段落</p>  
<div>div内容</div>  
<p>p段落</p>  
<p>p段落</p>
```

```
1 <!DOCTYPE html>  
2 <html lang="en">
```

```
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*兄弟选择器(相邻元素选择器)*/
8
9     /* 选择器名:
10
11     选择器1+选择器2+.....{
12     属性1:属性值1;
13     属性2:属性值2;
14     属性3:属性值3;
15     ...
16     }
17     */
18
19     /*用法:兄弟选择器只能选择选择器1"相邻下一个"的选择器2兄弟标签*/
20     div+p{
21       color:red;
22     }
23
24     div+p+p{
25       color:blue;
26     }
27   </style>
28 </head>
29 <body>
30   <p>p段落</p>
31   <p>p段落</p>
32   <div>div内容</div>
33   <p>p段落</p>
34   <p>p段落</p>
35 </body>
36 </html>
```

p段落

p段落

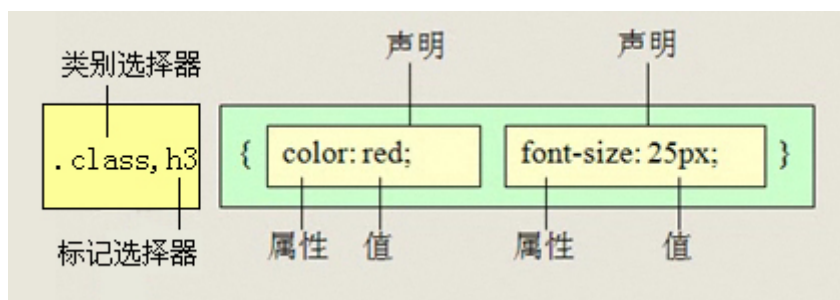
div内容

p段落

p段落

并集选择器(分组选择器)

并集选择器（分组选择器）是各个选择器通过**逗号**连接而成的，任何形式的选择器（包括标签选择器、class类选择器id选择器等），都可以作为并集选择器的一部分。如果某些选择器定义的样式完全相同，或部分相同，就可以利用并集选择器为它们定义相同的CSS样式。



选择器名:

```
1 选择器1,选择器2,选择器3...{  
2  属性1:属性值1;  
3  属性2:属性值2;  
4  属性3:属性值3;  
5  ...  
6 }
```

用法:

满足选择器1,选择器2,选择器3其中一个都会使用上这个样式!

举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*并集选择器也可以叫分组选择器,主要用于提取公共样式*/
8
9
10    /*选择器名
11     选择器1,选择器2,选择器3...{
12
13    }*/
14
15    /*只要满足选择器1,选择器2,选择器3...其中一个选择器,都是应用上对应的样式*/
16    div{
17      /*color:red;*/
18    }
19    p{
20      /*color:red;*/
21      font-style: italic;
22    }
23    .redFont{
24      /*color:red;*/
25      font-size:50px;
26    }
27
28    /*分组选择器 一般用于提取公共样式*/
29    div,p,.redFont{
30      color:blue;
31    }
32  </style>
33 </head>
34 <body>
35   <div>div的内容</div>
```

```
36 <p>p的内容</p>
37 <h1>标题内容</h1>
38 <div class="redFont">div的内容</div>
39 </body>
40 </html>
```

Document

文件 | F:/GZ20H5直播6班/04-CSS2/4-代码/0...

div的内容

p的内容

标题内容

div的内容

记忆技巧:

并集选择器 和 的意思，就是说，只要逗号隔开的，所有选择器都会执行后面样式。

比如 .one, p, #test {color: #F00;}

表示 .one 和 p 和 #test 这三个选择器都会把字体颜色设置为红色。

交集选择器(了解)

交集选择器由**两个**选择器构成，**即满足选择器1的标签又满足选择器2的标签**

选择器名:

```
1 选择器1选择器2{
2  属性1:属性值1;
3  属性2:属性值2;
```



```
4  属性3:属性值3;
5  ...
6  }
```

注意点:

- (1)选择器之间没有任何的连接符号
- (2)选择器可以是标签选择器,也可以是id选择器、类选择器
- (3)交集选择器仅仅是需要了解,平时用的比较少

用法:

找到即满足选择器1的标签又满足选择器2的标签

比如: p.one 选择的是p标签并且类名是.one的标签

p#two 选择的是p标签并且id名是.two的标签

举例:

需求:即是p标签,又是.red类的字体颜色设置为粉色

```
<p class="red">红色</p>
<p class="red">红色</p>
<p class="red">红色</p>
<div class="red">红色</div>
<div class="red">红色</div>
<div class="red">红色</div>
<div>div的内容</div>
<div>div的内容</div>
<div>div的内容</div>
<p>蓝色</p>
<p>蓝色</p>
<p>蓝色</p>
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Document</title>
6    <style>
7      /*交集选择器由两个选择器构成, 即满足选择器1的标签又满足选择器2的标签*/
8
9      /*两个选择器名之间没有任何的符号连接*/
10
11     /* 有以下两种:
12
13     标签选择器类选择器{
14
15     }
```

```
16
17 或者
18
19 标签选择器id选择器{
20
21 }
22 */
23
24 p{
25 color:blue;
26 }
27 .red{
28 color:red;
29 }
30
31 /*交集选择器 同时满足选择器1的标签,又满足选择器2的标签*/
32 p.red{
33 color:pink;
34 }
35 </style>
36 </head>
37 <body>
38 <!-- 需求:即是p标签,又是.red类的字体颜色设置为粉色 -->
39 <p class="red">红色</p>
40 <p class="red">红色</p>
41 <p class="red">红色</p>
42 <div class="red">红色</div>
43 <div class="red">红色</div>
44 <div class="red">红色</div>
45 <div>div的内容</div>
46 <div>div的内容</div>
47 <div>div的内容</div>
48 <p>蓝色</p>
49 <p>蓝色</p>
50 <p>蓝色</p>
51 </body>
52 </html>
```

红色

红色

红色

红色

红色

红色

div的内容

div的内容

div的内容

蓝色

蓝色

蓝色

复合选择器总结

选择器	作用	特征	使用情况	用法
后代选择器	用来选择元素或元素组的后代	子孙后代都能选	最多	div p { color: red;}
子代选择器	选择作为某元素子元素的元素	只选亲儿子	较少	.nav>p { color: red; }
兄弟选择器	只能选择选择器1相邻下一个的选择器2兄弟标签	相邻兄弟	较少	div+p{ color:red; }
并集选择器	选择某些相同样式的选择器	可以用于集体	较多	.nav, .header {color: red;}

		声明		
交集选择器	选择两个标签交集的部分	既...又有	较少	p.red {color: red;}

复合选择器练习,按已有的HTML结构使用符合选择器设置样式

- [公司首页](#)
- [公司简介](#)
- [公司产品](#)
- [联系我们](#)
 - [公司邮箱](#)
 - [公司电话](#)

左侧侧导航栏

[登录](#)

已有HTML结构

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Document</title>
```

```
  <style type="text/css">
```

```
    /*
```

在不修改HTML结构代码的前提下，完成以下任务：

链接 登录 的颜色为红色,同时主导航栏里面的所有的链接改为橙色 (简单)

主导航栏和侧导航栏里面文字都是20像素并且是宋体。（中等）

主导航栏里面的一级菜单链接文字颜色为绿色。（难）

```
    */
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <!-- 主导航栏 -->
```

```
  <div class="nav">
```

```
    <ul>
```

```
      <li> <a href="#">公司首页</a> </li>
```

```
      <li> <a href="#">公司简介</a> </li>
```

```
      <li> <a href="#">公司产品</a> </li>
```

```
      <li>
```

```
        <a href="#">联系我们</a>
```

```
        <ul>
```

```
          <li> <a href="#">公司邮箱</a> </li>
```

```
          <li> <a href="#">公司电话</a> </li>
```

```
        </ul>
```

```
      </li>
```

```
    </ul>
```

```
  </div>
```

```
<!-- 侧导航栏 -->
<div class="sitenav">
  <div class="site-l">左侧侧导航栏</div>
  <div class="site-r"><a href="#">登录</a></div>
</div>
</body>
</html>
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style type="text/css">
7     /* 在不修改HTML结构代码的前提下，完成以下任务： */
8     /*链接 登录 的颜色为红色,同时主导航栏里面的所有的链接改为橙色（简单）*/
9     .sitenav .site-r a{
10      color:red;
11    }
12    .nav a{
13      color:orange;
14    }
15    /*主导航栏和侧导航栏里面文字都是20像素并且是宋体。（中等）*/
16
17    /*分组选择器提取公共样式*/
18    .nav,.sitenav{
19      font-size:20px;
20      font-family: "宋体";
21    }
22
23    /*主导航栏里面的一级菜单链接文字颜色为绿色。（难）*/
24
25    /*子选择器*/
26    .nav>ul>li>a{
27      color:green;
28    }
29  </style>
30 </head>
31 <body>
32   <!-- 主导航栏 -->
```

```
33 <div class="nav">
34 <ul>
35 <li><a href="#">公司首页</a></li>
36 <li><a href="#">公司简介</a></li>
37 <li><a href="#">公司产品</a></li>
38 <li>
39 <a href="#">联系我们</a>
40 <ul>
41 <li><a href="#">公司邮箱</a></li>
42 <li><a href="#">公司电话</a></li>
43 </ul>
44 </li>
45 </ul>
46 </div>
47
48 <!-- 侧导航栏 -->
49 <div class="sitenav">
50 <div class="site-l">左侧侧导航栏</div>
51 <div class="site-r"><a href="#">登录</a></div>
52 </div>
53 </body>
54 </html>
```

伪类选择器 重点

什么是伪类选择器？

类选择器:给页面上的指定的类名的元素设置样式

伪类选择器:给页面上的元素设置样式，**必须满足一定的条件，我们的伪类选择器设置的样式才会显示**。比如:鼠标移上a链接以后,显示的样式就是伪类选择器

伪类选择器用于向某些选择器添加特殊的效果。比如给链接添加特殊效果，比如可以选择 第1个，第n个元素。

为了和我们刚才学的类选择器相区别，类选择器是一个点 比如 .demo {} 而我们的伪类 用 2个点 就是 冒号 比如 :link

总结：与类选择器相似，但是又有所不同。

链接伪类选择器

通过超级链接的不同状态，对其设置不同的样式！

选择器名	描述
:link	未访问的状态 超级链接没有被访问
:visited	访问过后状态 超级链接已经被访问
:hover	鼠标放上状态 鼠标放在超级链接上面，但是并没有将鼠标的左键按下去
:active	激活状态 鼠标左键已经按下去了，但是并没有将鼠标的左键弹出

伪类选择器都是带有冒号“:”

注意：

- 超级链接的不同状态它其实是有顺序。也就是说锚伪类选择器设置其实是有顺序。
- 如果不按照锚伪类选择器的顺序，那么样式就会失效。顺序： :link :visited :hover :active
- 记忆：要遵守“爱恨准则”要先有爱，才有恨。“Love Hate”(也可以通过lv是一个hao包包来记忆)

另外：

- :link和:visited只对a标签有效，对其他元素无效。
- :hover和:active对所有的元素都有效果

举例1：

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*链接伪类选择器*/
8     /*通过超级链接的不同状态，对其设置不同的样式！*/
9     /*:link 未访问的状态 超级链接没有被访问*/
```

```

10
11
12
13  a:link{
14  color:orange;
15  }
16
17  /*:visited 访问过后状态 超级链接已经被访问*/
18  a:visited{
19  color:pink;
20  }
21
22  /*:hover 鼠标放上状态 鼠标放在超级链接上面，但是并没有将鼠标的左键按下去*/
23  a:hover{
24  color:red;
25  }
26
27  a:active{
28  color:green;
29  }
30  /*:active 激活状态 鼠标左键已经按下去了，但是并没有将鼠标的左键弹出,也就是按住不放*/
31
32
33  /*注意：*/
34  /*超级链接的不同状态它其实是有顺序。也就是说锚伪类选择器设置其实是有顺序。*/
35
36  /*如果不按照锚伪类选择器的顺序，那么样式就会失效。顺序：:link :visited :hover :active*/
37
38  /*记忆：要遵守“爱恨准则”要先有爱，才有恨。“Love Hate”(也可以通过 lv是一个hao包包来记忆)*/
39  </style>
40  </head>
41  <body>
42  <a href="https://www.baidu.com">百度</a>
43  </body>
44  </html>

```

举例2:

```

1  <!DOCTYPE html>
2  <html lang="en">

```



```
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*另外:*/
8     /*:link和:visited只对a标签有效, 对其他元素无效。*/
9     a:link{
10      color:orange;
11    }
12    a:visited{
13      color:pink;
14    }
15
16    div:link{
17      color:green;
18    }
19    div:visited{
20      color:blue;
21    }
22    /*:hover和:active对所有的元素都有效果*/
23    a:hover{
24      color:red;
25    }
26    a:active{
27      color:purple;
28    }
29
30    div:hover{
31      /*设置背景颜色*/
32      background-color:skyblue;
33    }
34    div:active{
35      background-color: gold;
36    }
37    /*我们的: hover用的最多, 不管元素, 都可以设置: hover*/
38    span:hover{
39      color:red;
40    }
41  </style>
42 </head>
```

```
43 <body>
44 <a href="https://www.baidu.com">百度</a>
45 <div>div的内容</div>
46 <span>span的内容</span>
47 </body>
48 </html>
```

超链接的美化 重点

我们通常会去掉超级链接的下划线，然后同时会给超级链接设置一个颜色。

一般情况下：

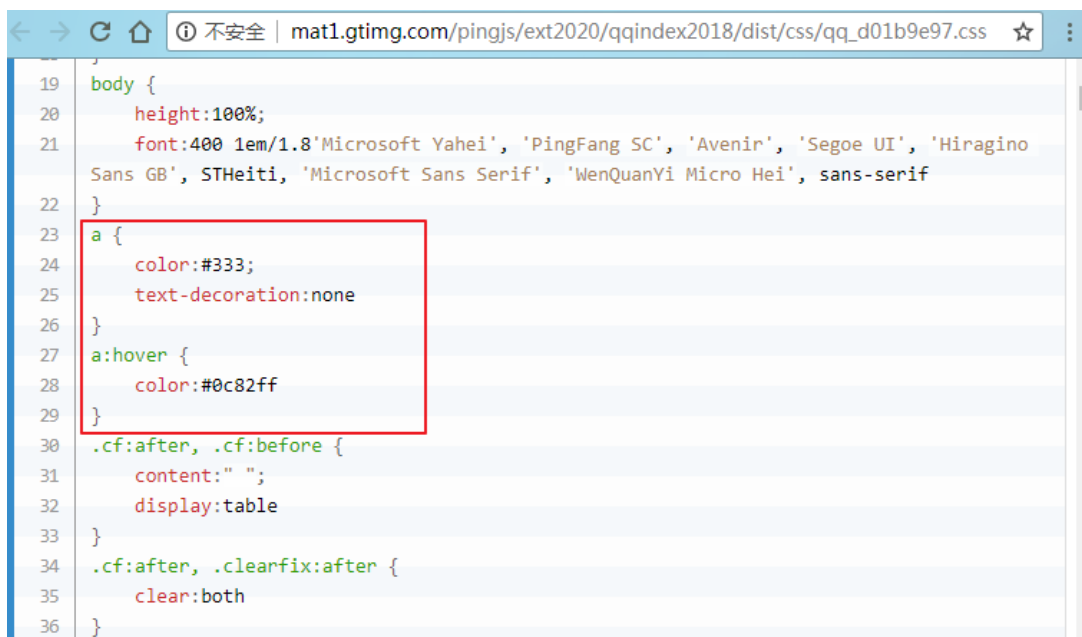
- 正常状态与访问过后的状态的样式设置为一样
- 当鼠标放上的时候给其设置另外一个颜色 激活状态一般不会设置 因为激活状态的时间太短

第一种对超级链接进行美化的设置方法：

```
1 a:link,a:visited{ 去掉超级链接的下划线;设置一个颜色;}
2 a:hover { 设置另外一种颜色;或者增加下划线;}
```

第二种对超级链接进行美化的设置方法： 必须掌握

我们先来观察www.qq.com这个网站是如何对超级链接进行美化？



```
1 a{color:#172c45;text-decoration:none}
2 /*上面的这种设置的方式 是集合了a:link与a:visited的功能*/
```

```
3 a:hover{color:#cd0200;text-decoration:underline}
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 /*我们会通常会去掉超级链接的下划线，然后同时会给超级链接设置一个颜色。*/
8
9 /*一般情况下：*/
10 /*正常未访问状态与访问过后的状态的样式设置为一样*/
11
12 /*当鼠标放上的时候给其设置另外一个颜色*/
13 /*激活状态一般不会设置 因为激活状态的时间太短 */
14
15
16 /*第一种对超级链接进行美化的设置方法： 了解
17
18 a:link,a:visited{去掉超级链接的下划线;设置一个颜色;}
19
20 a:hover {设置另外一种颜色;或者增加下划线;}*/
21
22 /*分组选择器,提取公共样式*/
23 /* a:link,a:visited{
24 color:#3c3c3c;
25 text-decoration: none;
26 }
27 a:hover{
28 color:orange;
29 text-decoration: underline;
30 } */
31
32 /*第二种对超级链接进行美化的设置方法： 必须掌握
33 a{color:#172c45;text-decoration:none}
34
35 a:hover{color:#cd0200;text-decoration:underline}
36 */
37
38 a{
```

```
39  color:#3c3c3c;
40  text-decoration: none;
41  }
42
43  a:hover{
44  color:orange;
45  text-decoration: underline;
46  }
47
48
49  </style>
50 </head>
51 <body>
52  <a href="#">连衣裙</a>
53  <a href="#">半身裙</a>
54  <a href="#">男士外套</a>
55 </body>
56 </html>
```

连衣裙 半身裙 男士外套



鼠标移上以后

练习-超级连接美化

要求使用无序列表以及a链接完成

- [习近平时间|用体育拉近中国与世界的距离一带一路](#)， ← 鼠标移上以后
- 合作共赢的可持续发展之路 积极进取，引领中国经济行稳致远
- 环球时报社评：唱衰中国的人和力量将是历史笑柄形成合力，
- 推动绿富同兴内蒙古库布其沙漠治理经验
- 风雨无阻创造美好生活
- 给孩子清朗的网络空间
- 保障未成年人健康上网

提供文字:

习近平时间|用体育拉近中国与世界的距离一带一路，
合作共赢的可持续发展之路 积极进取，引领中国经济行稳致远
环球时报社评：唱衰中国的人和力量将是历史笑柄形成合力，
推动绿富同兴内蒙古库布其沙漠治理经验
风雨无阻创造美好生活
给孩子清朗的网络空间
保障未成年人健康上网

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     a{
8       /*去掉下划线*/
9       text-decoration: none;
10      color:#646464;
11    }
12    a:hover{
13      /*加上下划线*/
14      text-decoration: underline;
15      color:red;
16    }
17  </style>
18 </head>
19 <body>
20   <!-- ul>li*7>a[#] -->
21
22   <ul>
23     <li><a href="#">习近平时间|用体育拉近中国与世界的距离一带一路，</a></li>
```

```

24 <li><a href="#">合作共赢的可持续发展之路 积极进取，引领中国经济行稳致远</a>
</li>
25 <li><a href="#">环球时报社评：唱衰中国的人和力量将是历史笑柄形成合力，</a></
li>
26 <li><a href="#">推动绿富同兴内蒙古库布其沙漠治理经验</a></li>
27 <li><a href="#">风雨无阻创造美好生活</a></li>
28 <li><a href="#">给孩子清朗的网络空间</a></li>
29 <li><a href="#">保障未成年人健康上网</a></li>
30 </ul>
31 </body>
32 </html>

```

结构伪类选择器 CSS3 重点 掌握

- **E:first-child** :选取属于其父元素的首个子元素的指定选择器
- **E:last-child** :选取属于其父元素的最后一个子元素的指定选择器
- **E:nth-child(n)** : 匹配属于其父元素的第 n 个子元素，**不论元素的类型,只关心元素在父元素中排第几**
- **E:nth-last-child(n)** : 从最后一个子元素开始计数,选择器匹配属于其元素的第 n 个子元素的每个元素，**不论元素的类型 ,只关心元素在父元素中排第几**

注意:

- n 可以是数字、关键词或公式
- n如果等于odd代表匹配奇数
- n如果等于even代表匹配偶数 其他关于n的说明可以参考网
址:https://blog.csdn.net/forest_fire/article/details/79279537

举例1:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 /*结构伪类选择器 CSS3*/
8 /*这里E=>element元素的意思*/

```

```

9  ul li{
10  color:red;
11  }
12  /*E:first-child找到第一个子元素*/
13  ul li:first-child{
14  color:pink;
15  }
16  /*E:last-child找到最后一个子元素*/
17  ul li:last-child{
18  color:green;
19  }
20  /*E:nth-child(n)找到第n个子元素,n从1开始,比如n=1,就是找到第一个子元素*/
21  ul li:nth-child(3){
22  color:purple;
23  }
24
25  /*E:nth-last-child(n)找到倒数第n个子元素,n从1开始,比如n=1,就是找到倒数第一个
子元素*/
26  ul li:nth-last-child(3){
27  color:orange;
28  }
29  </style>
30  </head>
31  <body>
32  <ul>
33  <li>香蕉</li>
34  <li>苹果</li>
35  <li>葡萄</li>
36  <li>奇异果</li>
37  <li>火龙果</li>
38  <li>桔子</li>
39  <li>橙子</li>
40  <li>木瓜</li>
41  <li>哈密瓜</li>
42  <li>西瓜</li>
43  </ul>
44  </body>
45  </html>

```

举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*E:nth-child(n)找到第n子元素,特别需要注意的是,"不关心是什么元素,只关心元素排
      第几"*/
8
9     .father p:nth-child(3){
10      color:red;
11    }
12
13    /*找到最后一个p标签*/
14    /* .father p:last-child{
15      color:blue;
16    } */
17
18    .father p:nth-child(6){
19      color:blue;
20    }
21
22    /*p:nth-type-of选择器只关心同类型排第几,不关心在父元素中排第几个,同学们自己可
      以去了解一下*/
23  </style>
24 </head>
25 <body>
26   <div class="father">
27     <p>p1p1p1p1p1</p>
28     <span>哈哈,我就在这里</span>
29     <p>p2p2p2p2p2</p>
30     <b>加粗标签</b>
31     <i>倾斜标签</i>
32     <p>p3p3p3p3p3</p>
33   </div>
34 </body>
35 </html>
```

举例2:实现隔行变色效果

- 00001
- 00002
- 00003
- 00004
- 00005
- 00006
- 00007
- 00008

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /* ul li:nth-child(1){
8       background-color: skyblue;
9     }
10    ul li:nth-child(2){
11      background-color: pink;
12    }
13    ul li:nth-child(3){
14      background-color: skyblue;
15    }
16    ul li:nth-child(4){
17      background-color: pink;
18    }
19    ul li:nth-child(5){
20      background-color: skyblue;
21    }
22    ul li:nth-child(6){
23      background-color: pink;
24    }
25    ul li:nth-child(7){
26      background-color: skyblue;
27    }
28    ul li:nth-child(8){
29      background-color: pink;
30    } */
31
32
33    /*E:nth-child(n)*/
34    /*注意：*/
35    /*n 可以是数字、关键词或公式 */
```

```
36  /*n如果等于odd代表匹配奇数 */
37  /*n如果等于even代表匹配偶数 其他关于n的说明可以参考网址:https://blog.csdn.net/forest_fire/article/details/79279537*/
38
39  ul li:nth-child(odd){
40      background-color: skyblue;
41  }
42  ul li:nth-child(even){
43      background-color: pink;
44  }
45  ul li:hover{
46      background-color: gray;
47  }
48  </style>
49  </head>
50  <body>
51      <ul>
52          <li>00001</li>
53          <li>00002</li>
54          <li>00003</li>
55          <li>00004</li>
56          <li>00005</li>
57          <li>00006</li>
58          <li>00007</li>
59          <li>00008</li>
60          <li>00009</li>
61          <li>00010</li>
62      </ul>
63  </body>
64  </html>
```

列表样式属性

这里的列表指的是：无序列表和有序列表

因为在整个网页布局中**无序列表**是使用的最多的。

属性	值	含义
list-style-type	none(将列表前面的项目符号 去掉),disc(实心圆),square(实心小方块),circle(空心圆)	设置列表前项目符号的类型。
list-style-position	Inside(在里面), outside (在外面)	设置列表项标记的放置位置。
list-style-image	<u>url</u> (图像路径)	将图象设置为列表项标记。
list-style	square inside url(arrow.gif)	在一个声明中设置所有列表属性

list-style-image:将列表前面的项目符号设置为一张图片 它的值是：url(图片的地址)

凡是在CSS中跟图片地址有关的 都需要将图片的地址放置在url(图片的地址)

list-style: 这个属性是一个简写属性 它集成了上面三个属性的功能。可以同时设置上面的三个属性。每一个属性值之前使用空格分隔！ 它的属性值可以有一个也可以有两个也可以三个。其属性值个数不定，位置也不定！

举例:

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Document</title>
6    <style>
7      /*列表样式属性,可以给ul或者ol或者li标签设置*/
8
9      ul li{
10         /*
11          list-style-type设置列表项目符号类型
12          取值:
13          disc实心圆点,默认值
14          square正方形
15          circle空心圆点
16          none这个取值用的最多,去掉列表项目符号,要记
17          */
18          /*list-style-type: disc;*/
19          /*list-style-type: square;*/
20          /*list-style-type: circle;*/
21          /*list-style-type: none;*/
22

```

```
23  /*list-style-position设置列表项目符号的位置
24  取值:inside里面 outside外面(默认值)
25  */
26  border:1px solid red;
27  /*list-style-position: inside;*/
28  /*list-style-position: outside;*/
29
30  /*list-style-image使用图片代替列表项目符号,取值url(图片所在路径)
31  */
32
33  /*list-style-image: url(images/sina.png);*/
34
35  /*list-style简写属性,可以同时设置list-style-type,list-style-position,list
  -style-image,属性值之间使用"空格"隔开,非常好用,可以省略任意一个或者多个,没有顺序
  要求*/
36  /*list-style:circle inside url(images/sina.png);*/
37  /*list-style:circle inside;*/
38  /*list-style:circle;*/
39
40  /*list-style:inside circle url(images/sina.png);*/
41
42
43  /*列表样式属性需要大家记忆就两个,都是用于去掉项目符号
44  list-style-type: none;
45  或者
46  list-style:none;
47  */
48
49  list-style:none;
50  }
51  </style>
52  </head>
53  <body>
54  <ul>
55  <li>张三</li>
56  <li>李四</li>
57  <li>王五</li>
58  <li>赵六</li>
59  <li>孙七</li>
60  </ul>
61  </body>
```

需要大家要记的:以下两种方法都可以**去掉列表前面的项目符号**

- `list-style-type:none;`
- `list-style:none;`

练习:结合列表样式属性和伪类选择器对超级链接进行美化

- ▶ 习近平时间|用体育拉近中国与世界的距离一带一路，
- ▶ 合作共赢的可持续发展之路 积极进取，引领中国经济行稳致远
- ▶ 环球时报社评：唱衰中国的人和力量将是历史笑柄形成合力，
- ▶ 推动绿富同兴内蒙古库布其沙漠治理经验
- ▶ 风雨无阻创造美好生活
- ▶ 给孩子清朗的网络空间
- ▶ 保障未成年人健康上网

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     a{
8       /*去掉下划线*/
9       text-decoration: none;
10      color:#646464;
11    }
12    a:hover{
13      /*加上下划线*/
14      text-decoration: underline;
15      color:red;
16    }
17
18    /*列表样式属性名可以加到ul上也可以加在li标签*/
19    ul li{
20      list-style: url(images/li01.jpg);
21    }

```

```
22 </style>
23 </head>
24 <body>
25 <!-- ul>li*7>a[#] -->
26
27 <ul>
28 <li><a href="#">习近平时间|用体育拉近中国与世界的距离一带一路, </a></li>
29 <li><a href="#">合作共赢的可持续发展之路 积极进取, 引领中国经济行稳致远</a>
</li>
30 <li><a href="#">环球时报社评: 唱衰中国的人和力量将是历史笑柄形成合力, </a></
li>
31 <li><a href="#">推动绿富同兴内蒙古库布其沙漠治理经验</a></li>
32 <li><a href="#">风雨无阻创造美好生活</a></li>
33 <li><a href="#">给孩子清朗的网络空间</a></li>
34 <li><a href="#">保障未成年人健康上网</a></li>
35 </ul>
36 </body>
37 </html>
```

元素的显示方式

定义

HTML标签默认情况下在展示的时候会出现一些独占一行, 或者多个HTML标签都在一行上显示的现象, 因此我们根据元素的显示方式不同, 将元素分为块级元素, 行内元素, 行内块元素。

元素显示方式的分类

1. 块级元素: 多个同名元素默认情况都会独占一行, 且宽高属性可以起作用, 宽度默认是父元素的100%

```
1 块级元素显示方式:    display:block;
2 特点:
3 1.独占一行
4 2.能够设置有效宽高
5 3.宽度默认是父元素的100%
6
7 常见块级元素:
8 div h1 h2 h3 h4 h5 h6 p ul li ol
```

2. 行内元素：多个同名元素默认情况下会在一行显示，且宽高属性不起作用,默认宽度由内容决定。

```
1 行内元素的显示方式: display:inline
2 特点:
3 1.一行可以显示多个
4 2.默认宽度就是它本身内容的宽度,不能设置有效宽高
5
6 常见的行内元素:
7 span i s del b strong em ins a
```

3. 行内块级元素：多个元素默认会显示在一行，且宽高属性起作用。

```
1 行内块级元素: display: inline-block;
2 特点:
3 1.一行显示多个
4 2.能设置有效宽高
5 3.默认宽度就是它本身内容的宽度
6
7 常见的行内块级元素:
8 input img
```

举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>
7 /*HTML标签默认情况下在显示的时候会出现一些独占一行,或者多个HTML标签都在一行上
  显示的现象,因此我们根据元素的显示方式不同,将元素分为块级元素,行内元素,行内块元素。
  */
8
9 .div1{
10 width: 100px;
11 height: 100px;
12 background-color:skyblue;
13 }
14 .div2{
15 background-color: pink;
16 }
17 ul li{
```

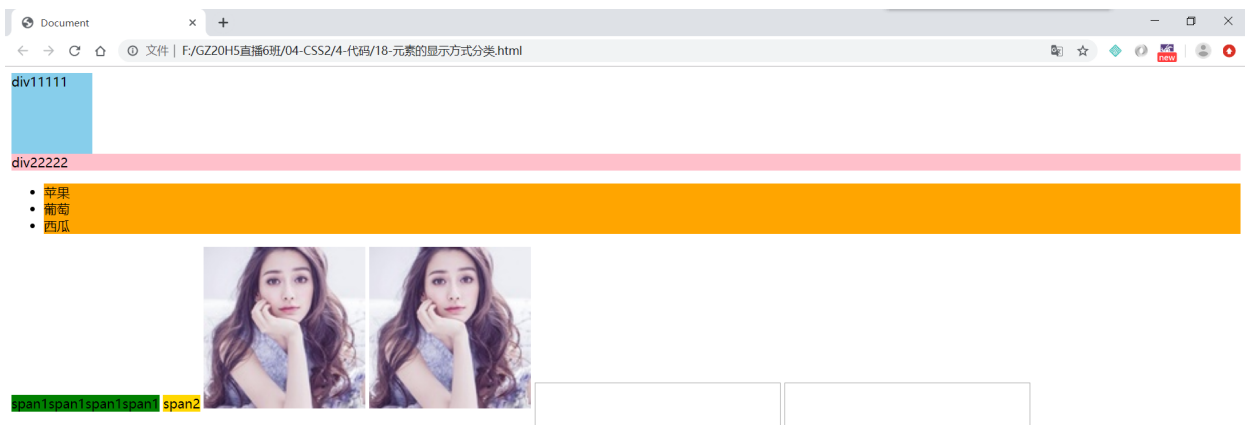
```
18 background-color:orange;
19 }
20
21 .span1{
22 background-color:green;
23 width:100px;
24 height: 100px;
25 }
26 .span2{
27 background-color:gold;
28 width: 200px;
29 height: 200px;
30 }
31
32 img{
33 width: 200px;
34 height: 200px;
35 }
36
37 input{
38 width: 300px;
39 height: 50px;
40 }
41 </style>
42 </head>
43 <body>
44 <!-- "每种元素的特点需要同学们记忆" -->
45 <!-- 块级元素特点:独占一行,可以设置宽度高度,默认宽度是父元素的100% -->
46 <!-- 常见块级元素有div,h1~h6,p,ul,li,ol... -->
47 <div class="div1">div11111</div>
48 <div class="div2">div22222</div>
49 <ul>
50 <li>苹果</li>
51 <li>葡萄</li>
52 <li>西瓜</li>
53 </ul>
54
55 <!-- 行内元素特点:不独占一行,一行可以多个,直到放不下才会自动换行,"设置宽度高度无效",默认宽度由"内容"决定 -->
56 <!-- 常见行内元素span,a,b,strong,u,ins,del,s,em,i,font等等 -->
57 <span class="span1">span1span1span1span1</span>
```



```

58 <span class="span2">span2</span>
59
60 <!-- 行内块元素特点:具有行内元素部分特点,又具有块级元素的部分特点;可以一行显示,又可以设置宽度高度,默认宽度由内容决定 -->
61 <!-- 常见行内块元素 img input -->
62 
63 
64
65 <input type="text"/>
66 <input type="text"/>
67 </body>
68 </html>

```



元素显示方式的转换(display属性) 要记

块级元素,行内元素,行内块元素三种元素,他们之间都可以相互转换,换成哪种元素,就有具有哪有元素的特点

1. 其它元素转块级元素: display: block;
2. 其它元素转行内元素: display:inline;
3. 其它元素转行内块级元素: display: inline-block;

举例:

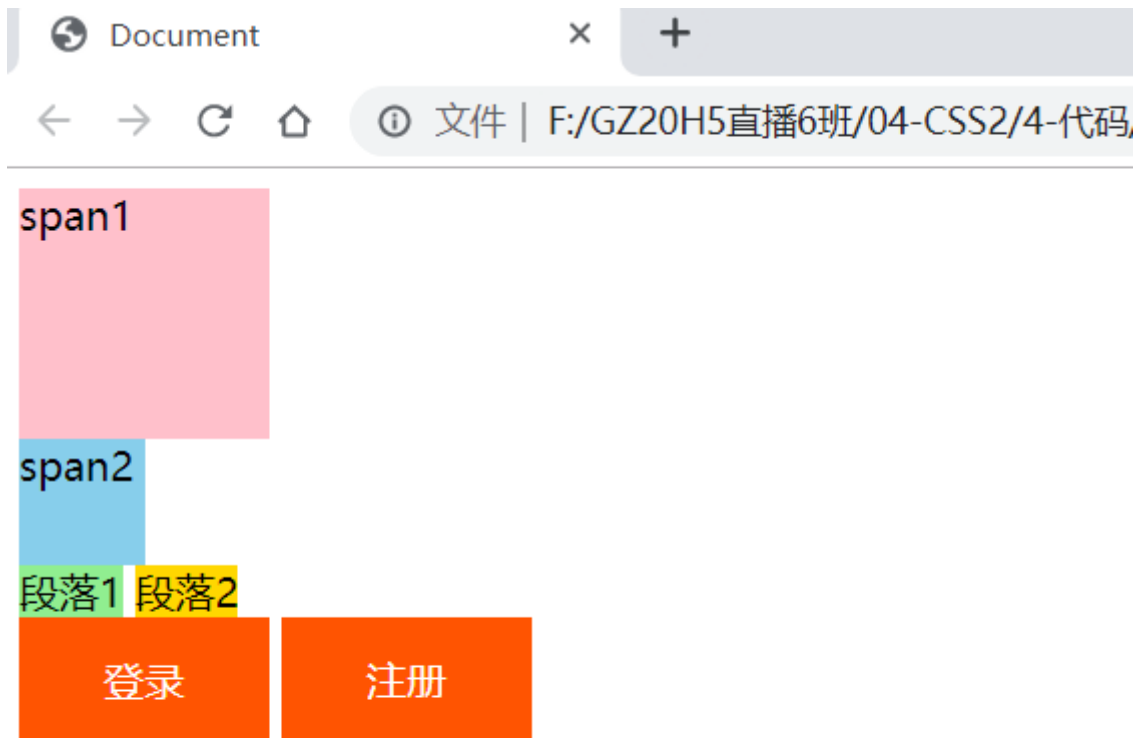
```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Document</title>
6 <style>

```

```
7  /*块元素,行内元素,行内块元素之间可以相互转换,一句话:"转换了什么元素,就有具有什  
   么元素的特点" ; 比如转成了块元素,那么就会具有块元素的特点*/  
8  
9  /*使用的属性是display*/  
10  
11  .span1{  
12      background-color:pink;  
13      width: 100px;  
14      height: 100px;  
15      /*display:block;其他元素转成块元素*/  
16      display: block;  
17  }  
18  .span2{  
19      background-color:skyblue;  
20      width: 50px;  
21      height: 50px;  
22      /*display:block;其他元素转成块元素*/  
23      display: block;  
24  }  
25  
26  .p1{  
27      background-color:lightgreen;  
28      width: 100px;  
29      height: 100px;  
30      /*display:inline;其他元素转成行内元素*/  
31      display: inline;  
32  }  
33  .p2{  
34      background-color:gold;  
35      width: 100px;  
36      height: 100px;  
37      /*display:inline;其他元素转成行内元素*/  
38      display: inline;  
39  }  
40  
41  
42  /*因为a默认是行内元素,宽度高度设置无效*/  
43  a{  
44      background-color:#ff5401;  
45      width: 100px;  
46      height: 50px;
```

```
47  color:white;
48  text-decoration: none;
49  /*display:inline-block;其他元素转成行内块元素*/
50  display: inline-block;
51  text-align: center;
52  /*设置行高的值等于高度的值,实现的垂直具有*/
53  line-height: 50px;
54  }
55  </style>
56  </head>
57  <body>
58  <span class="span1">span1</span>
59  <span class="span2">span2</span>
60
61  <p class="p1">段落1</p>
62  <p class="p2">段落2</p>
63
64  <br/>
65
66  <a href="#">登录</a>
67  <a href="#">注册</a>
68  </body>
69  </html>
```

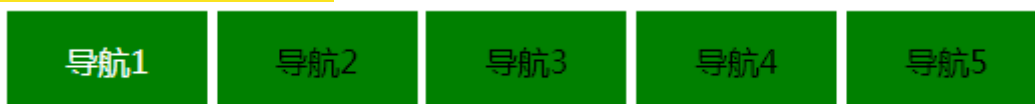


练习-元素的显示方式

备注:设置背景颜色的CSS属性是background-color:颜色值; 或者 background: 颜色值;

1. 三个 div 设置 100 * 100 的红色盒子
2. 三个 span 设置 150 * 150 绿色盒子,不同行显示这三个盒子
3. 三个 a 链接 80 * 20 蓝色 盒子 要求 在一行显示 这三个盒子

练习-简单导航案例



元素的显示方式小结

元素模式	元素排列	设置宽度高度	默认宽度	其他元素转换
块级元素	一行只能放一个块级元素	可以设置宽度高度	容器(父元素)的100%	display:block

行内元素	一行可以放多个行内元素	不可以直接设置宽度高度	它本身内容的宽度	display:inline
行内块元素	一行放多个行内块元素	可以设置宽度和高度	它本身内容的宽度	display:inline-block

CSS三大特性

CSS的三大特性分别是 **层叠性**、**继承性**、**优先级**

CSS层叠性

所谓层叠性是指多种**CSS样式的叠加**。

是浏览器处理冲突的一个能力,如果一个属性通过两个相同选择器设置到同一个元素上,那么这个时候一个属性就会将另一个属性层叠掉

比如先给某个标签指定了内部文字颜色为红色,接着又指定了颜色为蓝色,此时出现一个标签指定了相同样式不同值的情况,这就是样式冲突。

一般情况下

样式(css属性)冲突,遵循的原则是就近原则。那个样式离着结构近,就执行那个样式。

样式(css属性)不冲突,不会层叠

css的层叠性: 同一个元素的同一个属性,后面的样式会将起那么的样式覆盖,优先级高会将优先级低覆盖。

CSS最后的执行口诀: **长江后浪推前浪,前浪死在沙滩上。**



举例:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Document</title>
6    <style>
7      /*CSS的三大特性分别是 层叠性 、继承性 、优先级*/
8
9      /*所谓层叠性是指"多种CSS样式的叠加"*/
10
11
12     div{
13       color:red;
14       font-size:50px;
15     }
16
17     div{
18       color:blue;
19     }
20
21     /*一般情况下*/
22     /*样式(css属性)冲突，遵循的原则是就近原则。 那个样式离着结构近，就执行那个样
式。后面的样式会覆盖前面的样式;*/
23
24     /*样式(css属性)不冲突，不会层叠*/
25
```

```
26  /*css的层叠性：同一个元素的同一个属性，后面的样式会将起那么的样式覆盖，优先级
    高会将优先级低覆盖。*/
27
28  /*CSS最后的执行口诀： 长江后浪推前浪，前浪死在沙滩上。*/
29
30  /*也可以理解为使用油漆刷墙，先刷粉色，再刷天空蓝，最后你能看到的应该就是天空蓝色
    */
31  </style>
32  </head>
33  <body>
34    <div>div的内容</div>
35  </body>
36  </html>
```

Document

← → ↻ 🏠 ⓘ 文件 | F:/GZ20H5直播6班/04-CSS2/4-代码/

div的内容

CSS继承性

所谓继承性是指书写CSS样式表时，子标签会继承父标签的某些样式，如文本颜色和字号。想要设置一个可继承的属性，只需将它应用于父元素即可。

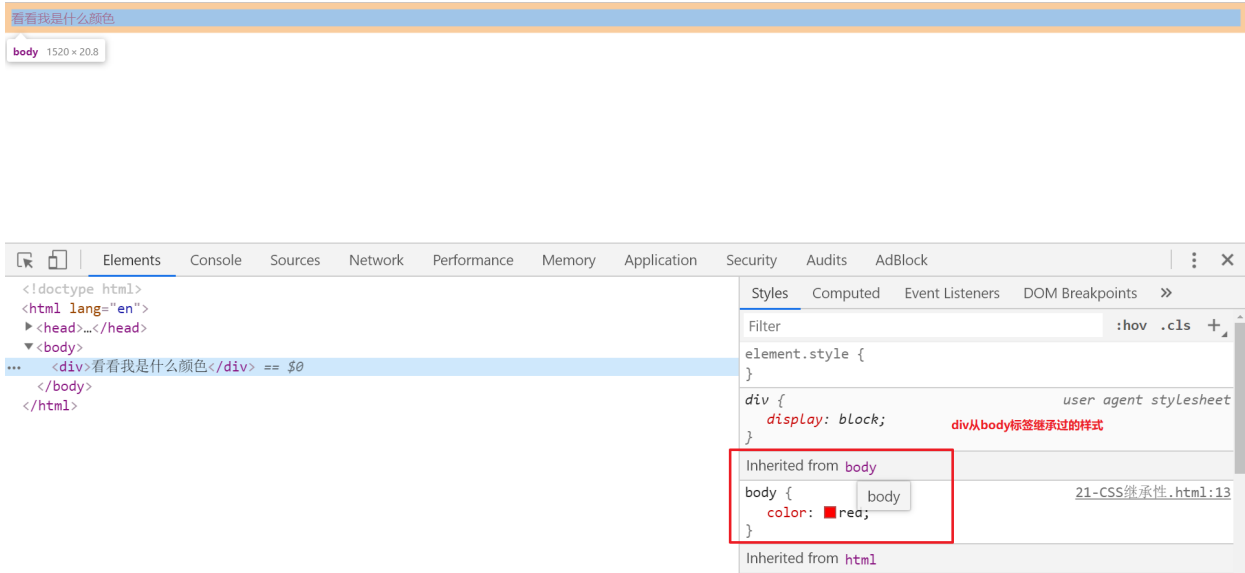
简单的理解就是：**子承父业**。

CSS最后的执行口诀： 龙生龙，凤生凤，老鼠生的孩子会打洞。



举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*所谓继承性是指书写CSS样式表时，子标签会继承父标签的某些样式，如文本颜色和字号
8     等;想要设置一个可继承的属性，只需将它应用于父元素即可。*/
9
10
11    /*简单的理解就是： "子承父业"。*/
12
13    /*CSS最后的执行口诀： 龙生龙，凤生凤，老鼠生的孩子会打洞。*/
14
15   body{
16     color:red;
17   }
18 </style>
19 </head>
20 <body>
21   <div>看看我是什么颜色</div>
22 </body>
23 </html>
```

注意:

子元素可以继承父元素的样式 (`text-`, `font-`, `line-`这些元素开头的都可以继承, 以及`color`属性) 文字类的

`a`标签设置了`href`属性以后,不能继承颜色,

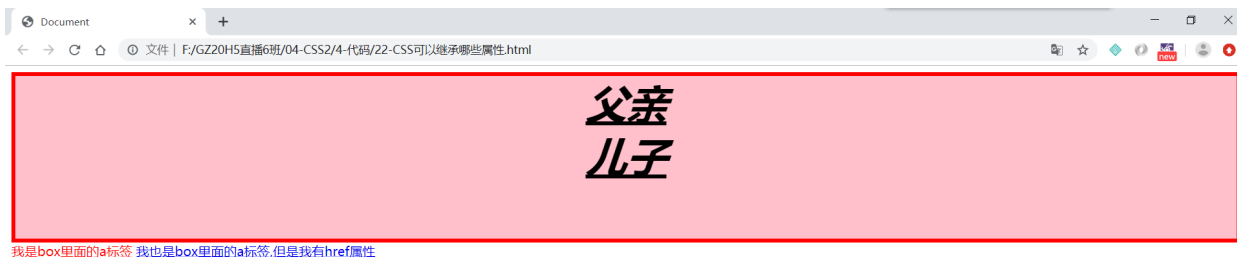
`h`系列标签不继承`font-size`和`font-weight`,因为`h`系列的标签使用的`font-size`设置字体大小单位是`em`,`em`是相对单位,1`em`等于一个父元素字体的大小

如果子元素自己有的样式,就不用从父元素那边继承相应的样式

举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*注意: */
8     /*子元素可以继承父元素的样式 (text-, font-, line-这些元素开头的都可以继承, 以及color属性) 文字类的*/
9     .father{
10       background-color:pink;
11       height: 200px;
12       text-align:center;
13       text-decoration: underline;
14       font-size:50px;
15       font-style:italic;
16       font-weight: bold;
```

```
17 border:5px solid red;
18 }
19 /*a标签设置了"href属性"以后,"不能"继承"颜色"*/
20 .box{
21 color:red;
22 }
23 /*h系列标签不继承font-size和font-weight,因为h系列的标签使用的font-size设置
字体大小单位是em,em是相对单位,1em等于一个父元素字体的大小*/
24 .box2{
25 /*因为标题标签自带加粗,所以font-weight属性不会继承过去,而标题标签不自带倾斜,
所以可以继承font-style属性*/
26 font-weight: normal;
27 font-style:italic;
28 font-size:50px;
29 }
30 /*一句话: 如果子元素自己有的样式,就不用从父元素那边继承相应的样式*/
31 </style>
32 </head>
33 <body>
34 <div class="father">
35 父亲
36 <div class="son">儿子</div>
37 </div>
38
39 <div class="box">
40 <a>我是box里面的a标签</a>
41 <a href="#">我也是box里面的a标签,但是我有href属性</a>
42 </div>
43
44 <div class="box2">
45 <h1>标题1标签</h1>
46 我是box2的文字内容
47 </div>
48 </body>
49 </html>
```



标题1标签

我是box2的文字内容

CSS优先级

定义CSS样式时，经常出现**两个或更多选择器选择在同一元素上**，这时就会出现**优先级的问题**。

基本选择器优先级比较：

继承<通配符选择器<标签选择器<类选择器<id选择器<行内样式<!important

注意:其中!important只有作用于某个属性,提高某个属性的优先级!

设置同个CSS属性时,优先级高会覆盖优先级低的!设置不同的CSS属性,就不会有优先级问题!

举例:body标签里面有一个div

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*刚才我们讲的层叠性有个前提,是他们的选择器优先级一样,我们才要考虑层叠性*/
8
9     /*CSS优先级*/
10    /*定义CSS样式时，经常出现两个或更多选择器选择在同一元素上，这时就会出现优先级的问题。*/
```

```
11
12  /*基础选择器优先级比较:
13  继承<通配符选择器<标签选择器<类选择器<id选择器<style行内式<!important属性
14  */
15
16  #two{
17  color:pink;
18  }
19  .one{
20  color:green;
21  }
22  div{
23  color:orange !important;
24  }
25  *{
26  color:blue !important;
27  }
28  body{
29  color:red;
30
31  /*!important语法格式*/
32  /*css属性:css属性值 !important;*/
33  /*color:red !important;*/
34  }
35
36
37  /*使用!important一定要注意以下几点*/
38  /*1.!important它是提升的属性的权重，而不是提升选择器的优先级! */
39  /*2.!important它只能提升作用于"自己身上的属性"，"并不能提升继承过来属性的权
重! "*/
40  </style>
41  </head>
42  <body>
43  <div class="one" id="two" style="color:purple">看看这行文字是什么颜色</di
v>
44  </body>
45  </html>
```

复合选择器我们不叫优先级了,我们称之为权重

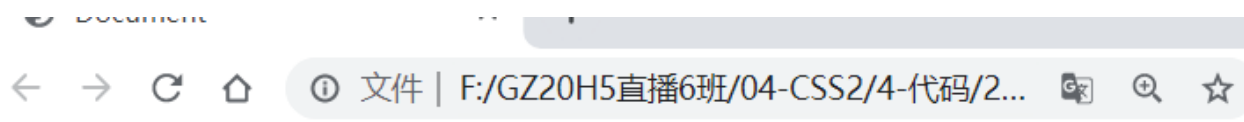
- 1 复合选择器的权重计算: (0 0 0 0)
- 2 从左到右:
- 3 第一个0表示!important的个数
- 4 第二个0表示id选择器的个数
- 5 第三个0表示类选择器或伪类的个数
- 6 第四个0表示标签选择器的个数

比较方法: 从左往右: 如果第一个0大, 表示这个选择器的权重高, 那么使用这个选择器的样式, 后面不用比较, 如果第一个0相等, 那么继续比较第二个0, 如果第二个0大, 那么就是这个选择器的权重高, 使用他的样式, 如果相等, 继续往下比较, 依次类推

举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*复合选择器的权重计算: (0 0 0 0) */
8     /* 从左到右:
9     第一个0表示!important的个数
10    第二个0表示id选择器的个数
11    第三个0表示类选择器或伪类的个数
12    第四个0表示标签选择器的个数 */
13
14    /*(0 1 0 1)*/
15    /* #father2 div{
16    color:blue;
17    } */
18
19    /*(0 0 1 1)*/
20    /* .father1 div{
21    color:red;
22    } */
23
```

```
24
25  /*比较方法：从左往右：如果第一个0大，表示这个选择器的权重高，那么使用这个选择
   器的样式，后面不用比较，如果第一个0相等，那么继续比较第二个0，如果第二个0大，那么就
   是这个选择器的权重高，使用他的样式，如果相等，继续往下比较，依次类推*/
26
27
28  /*(0 1 1 0)*/
29  .father1 #son2{
30    color:red;
31  }
32
33  /*(0 1 1 0)*/
34  #father2 .son1{
35    color:blue;
36  }
37
38  /*如果复合选择器权重相等的时候,应该考虑"层叠性"*/
39  /*这个不难，但是忽略很容易绕晕。其实，我们修改样式，"一定要看该标签有没有被选
   中"*/
40  /*1） 如果选中了，那么以上面的公式来计权重。谁大听谁的。 */
41  /*2） 如果没有选中，那么权重是0，因为继承的权重为0.*/
42  </style>
43  </head>
44  <body>
45    <div class="father1" id="father2">
46      <div class="son1" id="son2">猜猜我是什么颜色</div>
47    </div>
48  </body>
49  </html>
```



猜猜我是什么颜色

如果复合选择器权重相等的时候,应该考虑**层叠性**

这个不难,但是忽略很容易绕晕。其实,我们修改样式,一定要看该标签有没有被选中。

1) 如果选中了,那么以上面的公式来计权重。谁大听谁的。

2) 如果没有选中,那么权重是0,因为继承的权重为0.

CSS三大特性小结

子元素可以继承父元素的一些CSS样式

=> 继承性

后写的样式会覆盖先写的CSS 样式

=> 层叠性(同一个属性覆盖,不同

属性会叠加)

不同的CSS 选择器存在优先级现象

=> 优先级(基本选择器优先级以

及复合选择器的优先级)

!important属性

什么是important

important在英文中含义是“**重要的**”意思

!important在CSS中的作用

它主要是用来提升**属性的权重**。其属性的权重值无穷大!

!important语法格式

```
1 css属性:css属性值 !important;
```

一定要注意!important的语法规则:

正确的写法

```
1 css属性:css属性值 !important;
```

错误的写法

- 1 **CSS属性**:CSS值;**!important**; 不能将**!important**写在分号的外面 一定要写在分号的里面
- 2 **CSS属性**:CSS值 important; 不能将!给忘记了。

使用!important一定要注意以下几点

1.!important它是提升的属性的权重，而不是提升选择器的优先级！

2.**!important它只能提升作用于自己身上的属性，并不能提升继承过来属性的权重！**

举例：

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*刚才我们讲的层叠性有个前提,是他们的选择器优先级一样,我们才要考虑层叠性*/
8
9     /*CSS优先级*/
10    /*定义CSS样式时，经常出现两个或更多选择器选择在同一元素上，这时就会出现优先级的
    问题。*/
11
12    /*基础选择器优先级比较：
13    继承<通配符选择器<标签选择器<类选择器<id选择器<style行内式<!important属性
14    */
15
16    #two{
17      color:pink;
18    }
19    .one{
20      color:green;
21    }
22    div{
23      color:orange !important;
24    }
25    *{
26      color:blue !important;
27    }
28    body{
```



```

29  color:red;
30
31  /*!important语法格式*/
32  /*css属性:css属性值 !important;*/
33  /*color:red !important;*/
34  }
35
36
37  /*使用!important一定要注意以下几点*/
38  /*1.!important它是提升的属性的权重，而不是提升选择器的优先级! */
39  /*2.!important它只能提升作用于"自己身上的属性"，"并不能提升继承过来属性的权重! */
40  </style>
41  </head>
42  <body>
43  <div class="one" id="two" style="color:purple">看看这行文字是什么颜色</div>
44  </body>
45  </html>

```

练习:css 权重精华6题

CSS背景属性(background) 重点

CSS 可以为元素添加背景颜色和背景图片等相关背景设置。

背景颜色(background-color)

CSS3支持背景透明的写法语法格式是:

```

1 background-color: 英文颜色单词 或 十六进制 或 rgb(0~255,0~255,0~255) 或
  rgba(0, 0, 0, 0.3);

```

其中rgba的最后一个参数是alpha 透明度 取值范围 0~1之间,其他用法跟rgb一样

transparent 透明色

注意: 背景透明是指盒子背景半透明， 盒子里面的内容不受影响。

背景图片(background-image)

语法:

```

1 background-image : none 或 url

```

参数:

- 1 **none** : 无背景图 (默认的) **url** : 背景图像地址
- 2 **background-image** 属性允许指定一个图片展示在背景中 (只有CSS3才可以多背景)
- 3 可以和 **background-color** 连用。
- 4 如果图片不重复地话, 图片覆盖不到地地方都会被背景色填充。
- 5 如果有背景图片平铺, 则会覆盖背景颜色。

小技巧: 我们提倡 背景图片后面的地址, url不要加引号。

背景平铺(background-repeat)

语法:

- 1 **background-repeat** : repeat 或 no-repeat 或 repeat-x 或 repeat-y

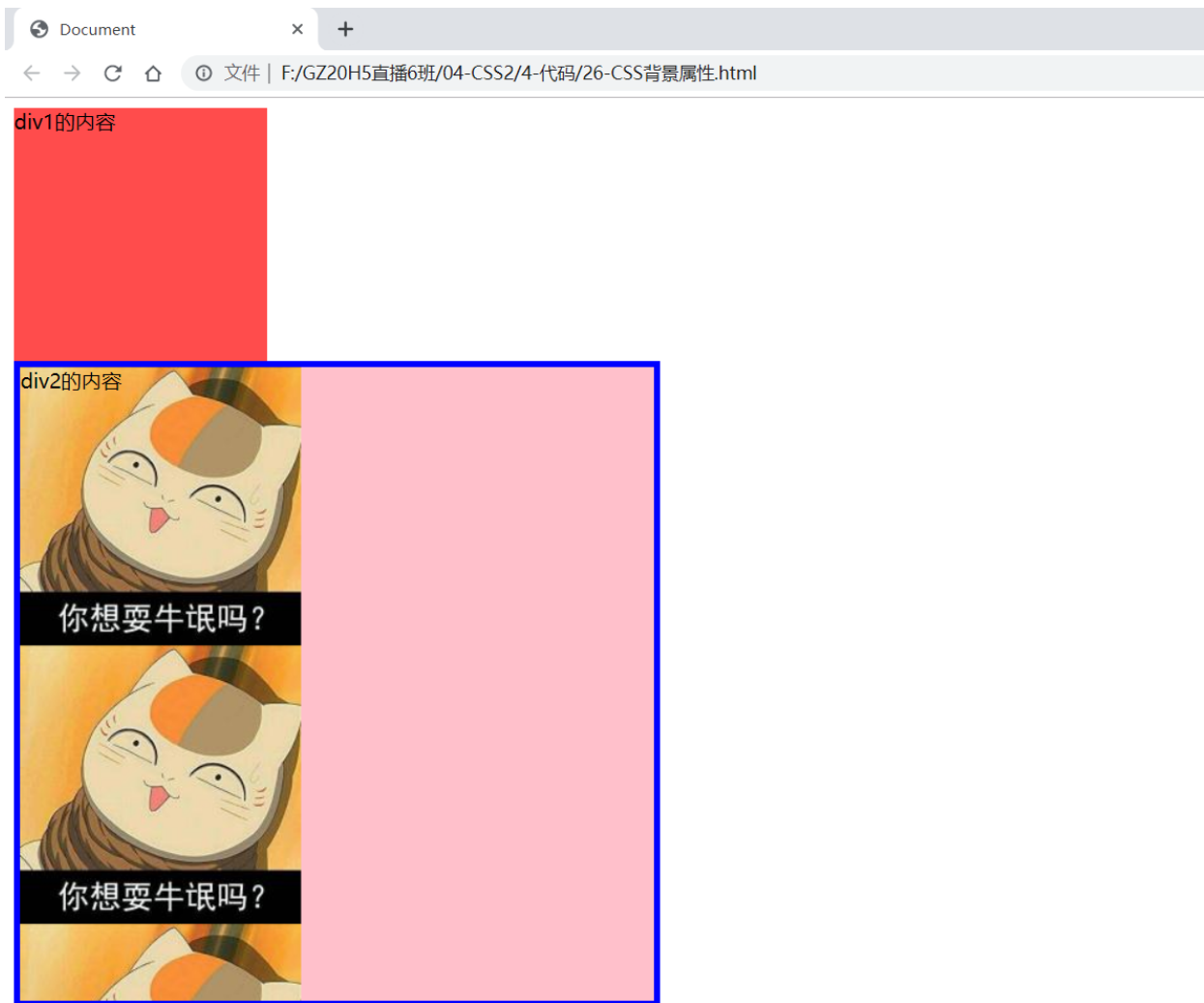
参数:

- 1 **repeat** : 背景图像在纵向和横向都平铺设置背景图片时,
- 2 默认把图片在水平和垂直方向平铺以铺满整个元素。
- 3
- 4 **no-repeat** : 背景图像不平铺
- 5 **repeat-x** : 背景图像在横向平铺
- 6 **repeat-y** : 背景图像在纵向平铺
- 7
- 8 设置背景图片时, 默认把图片在水平和垂直方向平铺以铺满整个元素。

举例:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     .div1{
8       /*background-color设置元素背景颜色,取值颜色值,英文颜色单词,rgb,十六进制,今天
       再教大家一种,rgba方式*/
9
10      width: 200px;
11      height: 200px;
12
13      /*注意:设置背景相关属性时候,需要留意有没有宽度高度或者有没有内容;*/
14      /*background-color:red;*/
15      /*background-color:rgb(0,255,0);*/
```

```
16  /*background-color: #f9264c;*/
17
18  /*rgba是rgb颜色值方式的升级版,a代表alpha,是透明度的意思,取值0~1之间所有数,允许小数,0代表完全透明,1代表完全不透明; 另外取值小数的时候,可以省略0,比如写0.7的时候,可以省略0,写成.7*/
19
20  /*background-color: rgba(255,0,0,0.7);*/
21  background-color: rgba(255,0,0,.7);
22  }
23
24  .div2{
25  width: 500px;
26  height: 500px;
27  border:5px solid blue;
28  /*background-image设置元素的背景图片,取值url(图片所在路径)*/
29  background-image: url(images/3.jpg);
30
31  /*background-repeat设置背景图片是否重复
32  取值
33  repeat-x 水平重复,垂直不重复
34  repeat-y 水平不重复,垂直重复
35  repeat 水平垂直都重复,默认值
36  no-repeat水平垂直都不重复
37  */
38  /*background-repeat: repeat-x;*/
39  background-repeat: repeat-y;
40  /*background-repeat: repeat;*/
41  /*background-repeat: no-repeat;*/
42
43  /*注意:如果同时设置了背景颜色和背景图片,那么背景图片会压住背景颜色;如果背景图片没有覆盖到的地方,就会使用背景颜色显示*/
44  background-color:pink;
45  }
46  </style>
47  </head>
48  <body>
49  <div class="div1">div1的内容</div>
50  <div class="div2">div2的内容</div>
51  </body>
52  </html>
```



背景定位(background-position) **重点,难点**

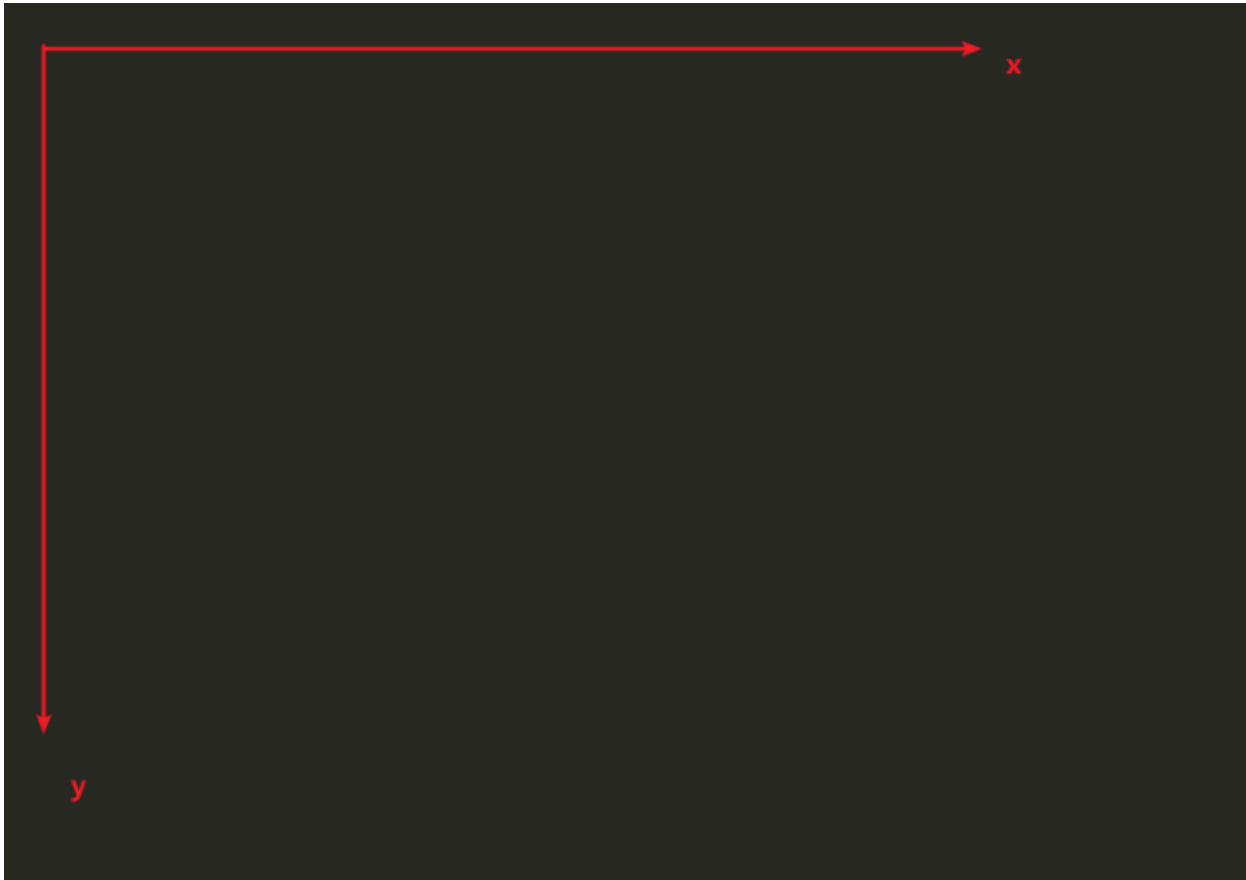
语法:

- 1 `background-position` : 水平位置 垂直位置;
- 2 `background-position` : length length;
- 3 `background-position` : position position ;

位置的表示方式有三种: 英文单词表示法 百分比表示 固定值表示法

- 1 英文单词表示法:
- 2 水平位置: left center right
- 3 垂直位置: top center bottom
- 4
- 5 百分比表示法: 50% 50%; 可以是正数也可以是负数
- 6
- 7 固定值表示法: 200px 300px; 可以是正数也可以是负数 这种用的最多
- 8

- 9 水平方向：正数向右走,负数向左走
- 10 垂直方向：正数向下走,负数向上走



参数：

- 1 **length**： 百分数 | 由浮点数字和单位标识符组成的长度值。
- 2 **position**： top | center | bottom | left | center | right 方位名词

说明：

- 1 设置或检索对象的背景图像位置。必须先指定background-image属性。
- 2 默认值为：(0% 0%)。
- 3 如果只指定了一个值，该值将用于横坐标。纵坐标将默认为50%。
- 4 第二个值将用于纵坐标。

注意：

- 必须先指定background-image属性
- position 后面是x坐标和y坐标。可以使用方位名词或者 精确单位。
- 如果指定两个值，两个值都是方位名字，则两个值前后顺序无关，比如left top和top left效果一致

- 如果只指定了一个方位名词，另一个值默认居中对齐。
- 如果position 后面是精确坐标，那么第一个，肯定是 x 第二的一定是 y
- 如果只指定一个数值,那该数值一定是x坐标，另一个默认垂直居中
- 如果指定的两个值是 精确单位和方位名字混合使用，则第一个值是x坐标，第二个值是y坐标
- 实际工作作用的最多的，就是背景图片居中对齐了。

举例:

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     .div1{
8       width: 600px;
9       height: 600px;
10      border:5px solid blue;
11      background-image: url(images/3.jpg);
12      background-repeat: no-repeat;
13      /*background-position设置"背景图片开始显示的位置",默认是从元素的"左上角"开始
        显示的*/
14
15      /*语法*/
16      /*background-position: 水平位置 垂直位置*/
17
18      /*取值方式有三种*/
19      /*第一种 方位位置英文单词
        水平 left center right
        垂直 top center bottom
        最多可以表示9个位置
20      */
21
22      /*background-position: left top;*/
23      /*background-position: right top;*/
24      /*background-position: center bottom;*/
25      /*background-position: center center;*/
26
27      /*第二种取值方式是百分比 百分比取值可以正值也可以负值*/
28
29

```

```
30  /*水平方向:正值背景图片往右边移动,负值背景图片往左边移动*/
31  /*垂直方向:正值背景图片往下边移动,负值背景图片往上边移动*/
32  /*background-position: 50% 100%;*/
33
34  /*第三种取值方式是数值带px单位,"这种方式用的最多",因为最准确,能够表示的位置非
    常多*/
35  /*数值取值可以正值也可以负值*/
36  /*水平方向:正值背景图片往右边移动,负值背景图片往左边移动*/
37  /*垂直方向:正值背景图片往下边移动,负值背景图片往上边移动*/
38  /*background-position: 0px 0px;*/
39  /*background-position: -52px -39px;*/
40
41  /*以上三种取值方式可以混搭使用*/
42  background-position: right 100px;
43  }
44  </style>
45  </head>
46  <body>
47    <div class="div1">div1的内容</div>
48  </body>
49  </html>
```



背景定位的应用-精灵图(雪碧图)



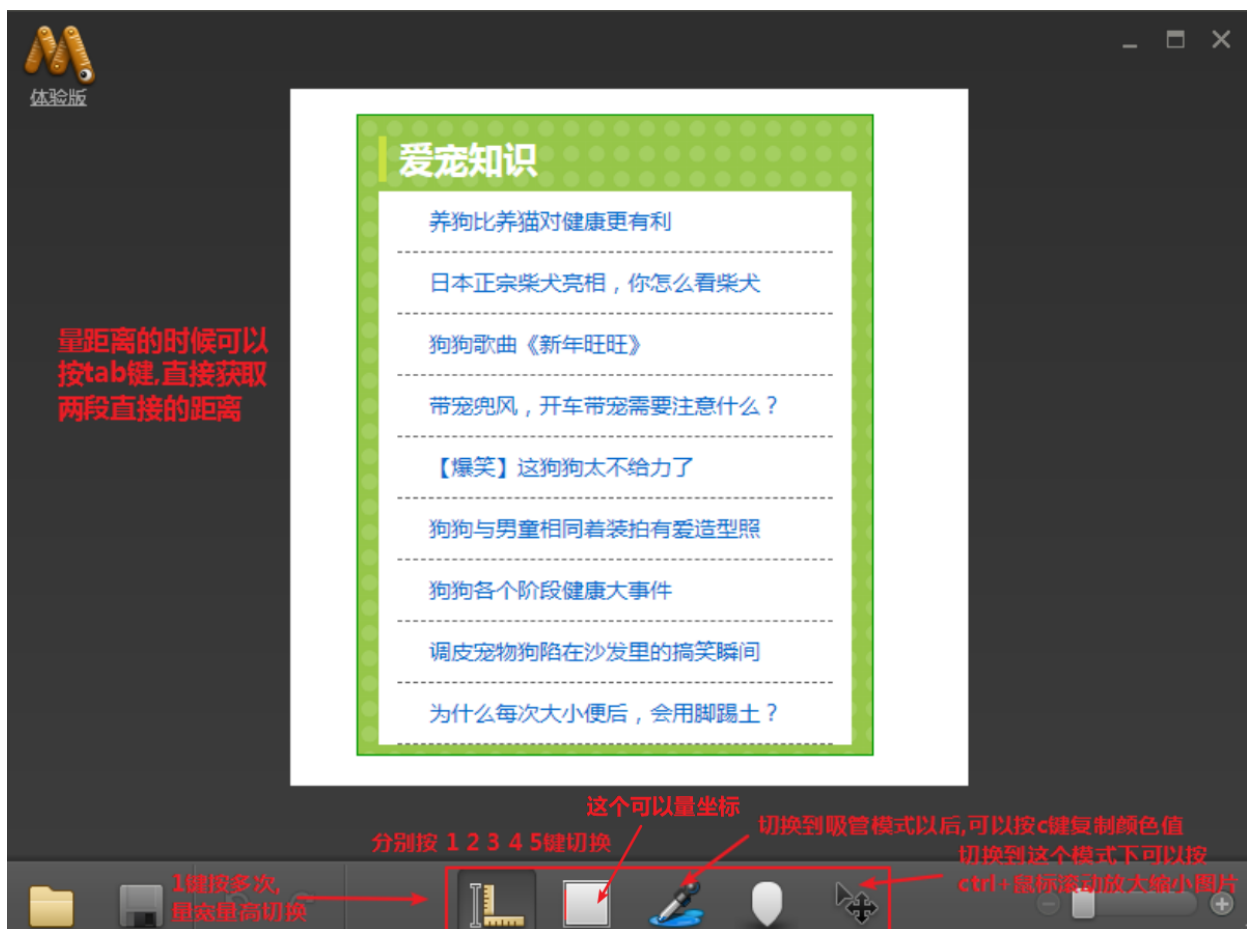
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <style>
7     /*我们经常使用背景定位从一个大图片中显示一个小图片出来*/
8     /*这种大图片里面放了很多小图片,这种图片我们称之为精灵图或者雪碧图*/
9
10    /*实现步骤:*/
```



```
11  /*第一步:设置元素背景图片*/
12  /*第二步:把图片中某个小图片的位置移到元素的左上角*/
13  /*第三步:设置一个合理的元素宽度和高度,让元素只显示小图片;*/
14
15  div{
16      background-image: url(images/abcd.jpg);
17      /*设置背景图片水平垂直都不重复*/
18      background-repeat: no-repeat;
19      /*转成行内块元素*/
20      display: inline-block;
21  }
22  .j{
23      width: 105px;
24      height: 107px;
25      background-position: -388px -143px;
26  }
27  .a{
28      width: 108px;
29      height: 121px;
30      background-position: 0px 0px;
31  }
32  .c{
33      width: 104px;
34      height: 114px;
35      background-position: -240px -9px;
36  }
37  .k{
38      width: 105px;
39      height: 111px;
40      /*background-position: -495px -143px;*/
41      background-position: right -143px;
42  }
43  </style>
44  </head>
45  <body>
46      <div class="j"></div>
47      <div class="a"></div>
48      <div class="c"></div>
49      <div class="k"></div>
50  </body>
```

> 此电脑 > 软件 (D:) > 软件安装包 > MarkMan安装

名称	修改日期	类型	大小
①先装这个AdobeAIRInstaller.exe	2018/9/17 14:21	应用程序	11,269 KB
②再装这个MarkMan.air	2018/9/17 14:20	Installer Package	1,607 KB
③markman使用说明.png	2019/3/11 16:01	看图王 PNG 图片...	296 KB



今日总结

今日作业

使用xmind总结当天知识点

完成购物车效果



完成五彩导航效果

默认显示效果





鼠标移上第一个a标签



鼠标移上第二个a标签



鼠标移上第三个a标签

拼出自己的拼音姓名



这4个字母改成自己的姓名首字母,比如张三,显示ZS即可

完成导航案例



默认显示效果



鼠标移上某个以后,就会改变成这样