

昨日回顾

CSS是什么?层叠样式表

CSS主要作用用于干嘛?美化已有HTML结构的网页

CSS的语法规则

```
CSS选择器{  
    CSS属性名1:CSS属性值1;  
    CSS属性名2:CSS属性值2;  
    CSS属性名3:CSS属性值3;  
    ...  
}
```

CSS选择器用于干嘛的? 选择目标标签的,选中标签以后才能设置CSS样式属性

CSS基础选择器

标签选择器	选择器名就是标签名	选择所有同名标签
类选择器	选择器名就是 .class属性值	选择一个或者多个具有这个类名的标签
多类名选择器	可以在一个标签上设置多个类名,类名之间使用空格隔开 只有class属性	
通配符选择器	选择器名就是*	选择所有标签

尺寸样式属性

宽度width 数值带px或者百分比(相对父元素)
高度height 数值带px或者百分比(相对父元素)

文本相关属性

字体颜色color
水平对齐方式text-align: left center right
行高 line-height 实现单行文字垂直居中,设置行高的值等于高度的值
首行缩进 text-indent:2em;
文本修饰线 text-decoration: underline或者none

字体相关属性

字体大小font-size 数值带px
字体是否倾斜font-style italic normal
字体是否加粗font-weight bold(700) normal(400)
何种字体font-family 字体名称
简写属性font 是否倾斜 是否加粗 字体大小/行高 何种字体;

学习目标

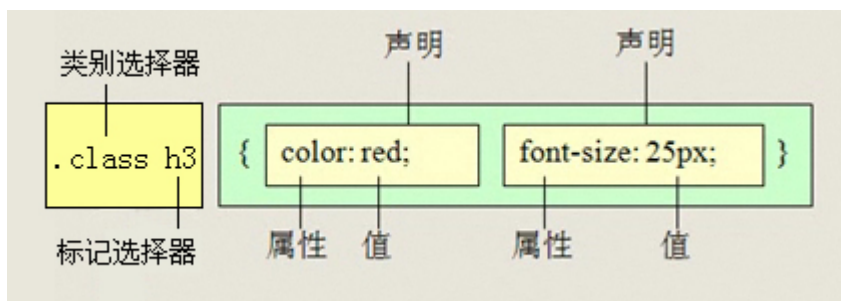
- 能够写出后代选择器的语法并解释其作用
- 能够理解并使用伪类选择器
- 能够使用列表属性
- 能够转换元素的显示方式
- 能够灵活运用背景属性

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:

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

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a标签在不同的父元素下面

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Document</title>  
    <style>  
        /* a{
```

```

        color:red;
    } */

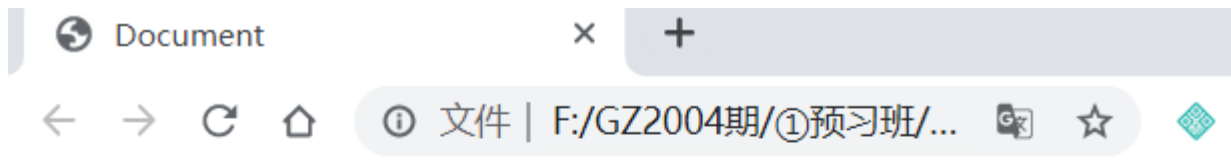
    /*后代选择器*/
    /*
    选择器1 选择器2 选择器3 ...{
        CSS属性名:CSS属性值;
    }
    */

    /*找到.shopping里面的所有a标签*/
    /*后代选择器找的是所有的子孙后代*/
    .shopping a{
        color:orange;
    }

    .video a{
        color:pink;
    }
</style>
</head>
<body>
    <div class="shopping">
        <a href="#">小米</a>
        <a href="#">京东</a>
        <a href="#">淘宝</a>
        <p>
            p段落
            <a href="#">百度</a>
        </p>
    </div>

    <!-- .类名再按tab键就可以生成对应类名的标签 -->
    <div class="video">
        <a href="#">爱奇艺</a>
        <a href="#">腾讯视频</a>
        <a href="#">优酷</a>
        <a href="#">芒果TV</a>
    </div>
</body>
</html>

```



小米 京东 淘宝

p段落 百度

爱奇艺 腾讯视频 优酷 芒果TV

小结:后代选择器选择子孙后代!!!

伪类选择器

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。

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
- :hover所有元素都可以使用,而且用的很频繁 用的最多

举例1:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*链接伪类选择器就是给超链接设置的选择器*/
    /*a:link未访问的链接状态*/
    a:link{
      color:red;
      text-decoration: none;
    }
    /*a:visited访问过来的链接状态*/
    a:visited{
      color:skyblue;
    }
    /*a:hover鼠标移上a链接以后的样式*/
    a:hover{
      color:green;
      text-decoration: underline;
    }
  </style>
</head>
</html>
```

```

/*a:active鼠标左键按下,不松开的样式*/
a:active{
    color:orange;
}

/*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)*/
</style>
</head>
<body>
    <a href="https://www.qq.com">腾讯</a>
</body>
</html>

```

举例2:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        /*
            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
            :hover所有元素都可以使用,而且用的很频繁    用的最多
        */
        div{
            width: 100px;
            height: 100px;
            background-color: red;
        }
        div:link{
            background-color: blue;
        }
        div:visited{
            background-color: green;
        }
        /*鼠标移上div以后的样式*/
        div:hover{
            background-color:skyblue;
        }
        div:active{
            background-color: orange;
        }
        p{
            width: 200px;

```

```
        height: 200px;
        background-color: blue;
    }
    p:hover{
        background-color:lightgreen;
    }
</style>
</head>
<body>
    <div>div的内容</div>
    <p>p段落标签</p>
</body>
</html>
```

超链接的美化 重点,常用

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：

```
a:link,a:visited{ 去掉超级链接的下划线;设置一个颜色;}
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：

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？

```
19 body {
20     height:100%;
21     font:400 1em/1.8'Microsoft Yahei', 'PingFang SC', 'Avenir', 'Segoe UI', 'Hiragino
    Sans GB', STHeiti, 'Microsoft Sans Serif', 'WenQuanYi Micro Hei', sans-serif
22 }
23 a {
24     color:#333;
25     text-decoration:none
26 }
27 a:hover {
28     color:#0c82ff
29 }
30 .cf:after, .cf:before {
31     content:" ";
32     display:table
33 }
34 .cf:after, .clearfix:after {
35     clear:both
36 }
```

```
a{color:#172c45;text-decoration:none}
/*上面的这种设置的方式 是集合了a:link与a:visited的功能*/
a:hover{color:#cd0200;text-decoration:underline}
```

举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        /*一般网页的超链接,我们能看到的只有两种状态*/
        /*未访问跟已经访问是一种状态*/
        /*鼠标移上跟鼠标按下又是一种状态*/

        /*第一种对超级链接进行美化的设置方法:*/
        /*a:link,a:visited{去掉超级链接的下划线;设置一个颜色;}*/
        /*a:hover {设置另外一种颜色;或者增加下划线;}*/

        /*分组选择器*/
        /* a:link,a:visited{
            text-decoration: none;
            color:#3c3c3c;
        }
        a:hover{
            color:#ff4400;
            text-decoration: underline;
        } */
```



```

/*第二种对超级链接进行美化的设置方法  同学们需要掌握第二种美化方法*/
a{
    text-decoration: none;
    color:#3c3c3c;
}
a:hover{
    color:#ff4400;
    text-decoration: underline;
}
</style>
</head>
<body>
    <a href="#">四件套</a>
    <a href="#">新款连衣裙</a>
</body>
</html>

```

结构伪类选择器 CSS3

E=>element 元素 可以理解为标签

- 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,只关心元素排第几个
- 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

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:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        /*后代选择器*/
        ul li{
            color:red;
        }

        /*结构伪类选择器*/
        /*E=>element元素,其实就是我们的标签*/
    
```

```
/*E:first-child找到第一个子元素*/
ul li:first-child{
    color:blue;
}
/*E:last-child找到最后一个子元素*/
ul li:last-child{
    color:orange;
}
/*E:nth-child(n) 找到第n个子元素,n是从1开始的数字 如果n=3,就代表找到第3个子元素*/
ul li:nth-child(3){
    color:purple;
}

/*E:nth-last-child(n) 找到倒数第n个子元素,n是从1开始的数字 如果n=3,就代表找到倒数第3个子元素
*/
ul li:nth-last-child(3){
    color:yellow;
}
</style>
</head>
<body>
    <ul>
        <li>香蕉</li>
        <li>苹果</li>
        <li>葡萄</li>
        <li>奇异果</li>
        <li>火龙果</li>
        <li>橘子</li>
        <li>草莓</li>
        <li>菠萝</li>
        <li>圣女果</li>
        <li>西瓜</li>
    </ul>
</body>
</html>
```

- 香蕉
- 苹果
- 葡萄
- 奇异果
- 火龙果
- 橘子
- 草莓
- 菠萝
- 圣女果
- 西瓜

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

隔行换色案例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /* ul li:first-child{
      background-color: skyblue;
    }
    ul li:nth-child(2){
```

```
        background-color: pink;
    }
    ul li:nth-child(3){
        background-color: skyblue;
    }
    ul li:nth-child(4){
        background-color: pink;
    } */

/*nth-child(odd)找到所有奇数子元素*/
ul li:nth-child(odd){
    background: skyblue;
}
/*nth-child(even)找到所有偶数子元素*/
ul li:nth-child(even){
    background: pink;
}
/*鼠标移上某个li以后*/
ul li:hover{
    background: gray;
}
</style>
</head>
<body>
    <ul>
        <li>香蕉</li>
        <li>苹果</li>
        <li>葡萄</li>
        <li>奇异果</li>
        <li>火龙果</li>
        <li>橘子</li>
        <li>草莓</li>
        <li>菠萝</li>
        <li>圣女果</li>
        <li>西瓜</li>
    </ul>
</body>
</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:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*列表样式属性给列表设置的*/
    /*我们暂时只有学了无序列表*/
    /*列表样式属性是给ul或者li标签设置都可以*/
    li{
      /*list-style-type设置列表项目符号类型*/
      /*
        取值
        disc实心圆点,默认值
        circle空心圆点
        square方块
        none 常用,必须记,去掉列表项目符号
      */
      /*list-style-type: disc;*/
      /*list-style-type: circle;*/
      /*list-style-type: square;*/
      /*list-style-type: none;*/

      /* list-style-position设置列表项目符号的位置
      取值 inside里面 outside外面,默认值 */
      border:1px solid red;
      /*list-style-position: outside;*/
      /*list-style-position: inside;*/

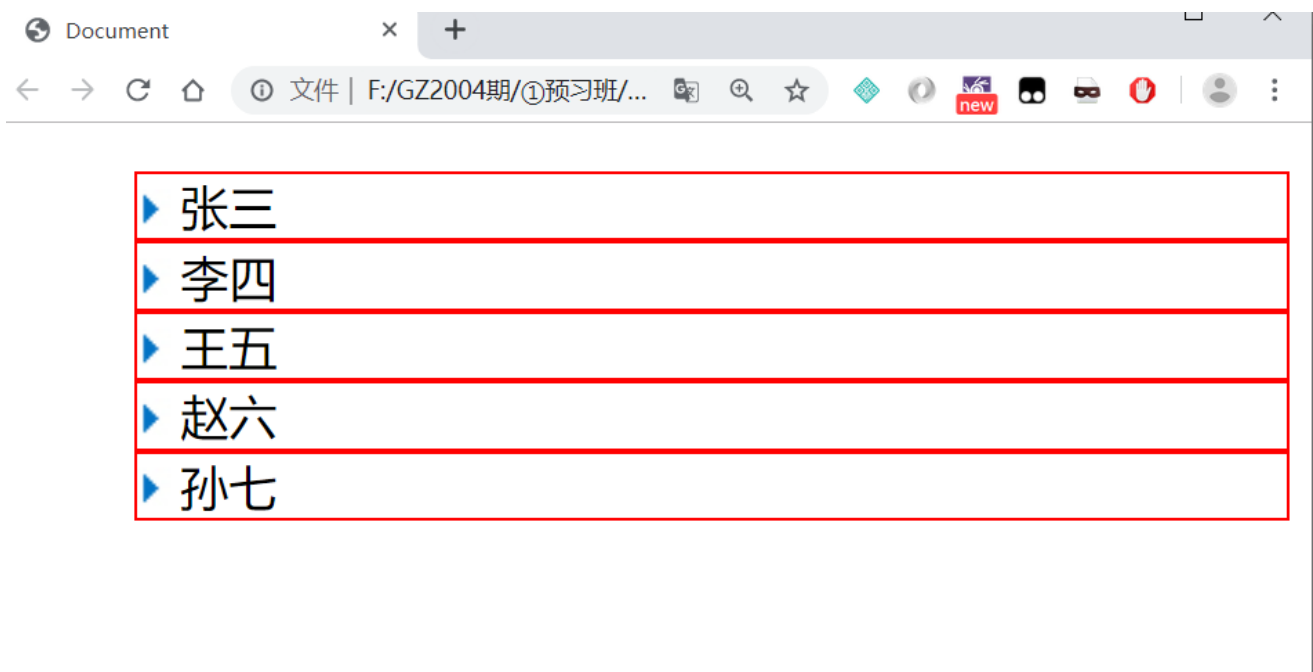
      /*list-style-image使用图片代替列表项目符号*/
      /*list-style-image:url(图片所在路径);*/
      /*list-style-image: url(images/sina.png);*/
```

```

        /*list-style简写属性,可以同时设置list-style-type,list-style-position,list-style-image,
        属性取值之间使用"空格"隔开,没有顺序要求,也没有说哪个属性值必须写,不写就会使用默认值*/
        list-style: circle inside url(images/li01.jpg);

        /*需要大家记忆就两句 去掉列表项目符号*/
        /* list-style: none;
        或者
        list-style-type:none; */
    }
</style>
</head>
<body>
    <ul>
        <li>张三</li>
        <li>李四</li>
        <li>王五</li>
        <li>赵六</li>
        <li>孙七</li>
    </ul>
</body>
</html>

```



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 列表样式属性可以给ul设置也可以给li标签设置

- **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

要求: 使用 `ul li a` 标签实现

- ▶ [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](#)

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

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*超链接的美化*/
    a{
      /*去掉下划线*/
      text-decoration: none;
      /*改变字体颜色*/
      color:#636363;
    }
    /*鼠标移上a链接以后的样式*/
    a:hover{
      color:red;
      text-decoration: underline;
    }

    li{
      list-style:url(images/li01.jpg);
    }
  </style>
</head>
<body>
  <ul>
    <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>
    <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>
    <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>
    <li><a href="#">推动绿富同兴内蒙古库布其沙漠治理经验</a></li>
    <li><a href="#">风雨无阻创造美好生活</a></li>
    <li><a href="#">给孩子清朗的网络空间</a></li>
    <li><a href="#">保障未成年人健康上网</a></li>
  </ul>
</body>
</html>
```

```
/*第一个li标签a元素*/
/* li:first-child a{
    color:red;
    text-decoration: underline;
} */

/*第四个li标签a元素*/
li:nth-child(4) a{
    color:red;
    text-decoration: underline;
}
</style>
</head>
<body>
<ul>
<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>
<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>
<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>
<li><a href="#">推动绿富同兴内蒙古库布其沙漠治理经验</a></li>
<li><a href="#">风雨无阻创造美好生活</a></li>
<li><a href="#">给孩子清朗的网络空间</a></li>
<li><a href="#">保障未成年人健康上网</a></li>
</ul>
</body>
</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各占一行, 且宽高属性可以起作用, 宽度默认是父元素的100%

块级元素显示方式: `display: block;`

特点:

1. 独占一行
2. 能够设置有效宽高
3. 宽度默认是父元素的100%

常见块级元素:

`div h1 h2 h3 h4 h5 h6 p ul li`

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。

行内元素的显示方式: `display: inline`

特点:

1. 一行可以显示多个
2. 默认宽度就是它本身内容的宽度, 不能设置有效宽高

常见的行内元素:

`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。 同时具有块元素的特点, 又具有行内元素的特点, 可以设置宽度高度, 又可以一行排列多个

行内块级元素: `display: inline-block;`

特点:

1. 一行显示多个
2. 能设置有效宽高
3. 默认宽度就是它本身内容的宽度

常见的行内块级元素:

`input img`

举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    .div1{
      width: 100px;
```

```

        height: 100px;
        background: skyblue;
    }
    .div2{
        background: pink;
    }
    ul{
        width: 100px;
        height: 200px;
        background: orange;
    }
    ul li{
        width: 50px;
        height: 50px;
        background: green;
    }

    span{
        width: 1000px;
        height: 1000px;
    }
    a{
        background: yellow;
        width: 200px;
        height: 200px;
    }

```

</style>

</head>

<body>

<!-- 在HTML标签中,我们会发现有的标签是从上到下排列,有的标签从左到右排列 -->

<!-- 那么我们会根据标签排列顺序的不同,把标签分为三种显示方式 -->

<!-- 块级元素,行内元素,行内块元素 -->

<!-- 块级元素特点:指的是独占一行,并且可以使用CSS设置宽度高度,默认宽度是父元素100% -->

<!-- 常见块级元素有div p h1~h6 ul li... -->

<div class="div1">div1</div>

<div class="div2">div2</div>

张三

李四

王五

苹果

西瓜

冬瓜

<!-- 行内元素特点:不独占一行,一行可以存放多个,直到放不下才会换行显示,CSS设置宽度高度无效 默认宽度由内容决定 -->

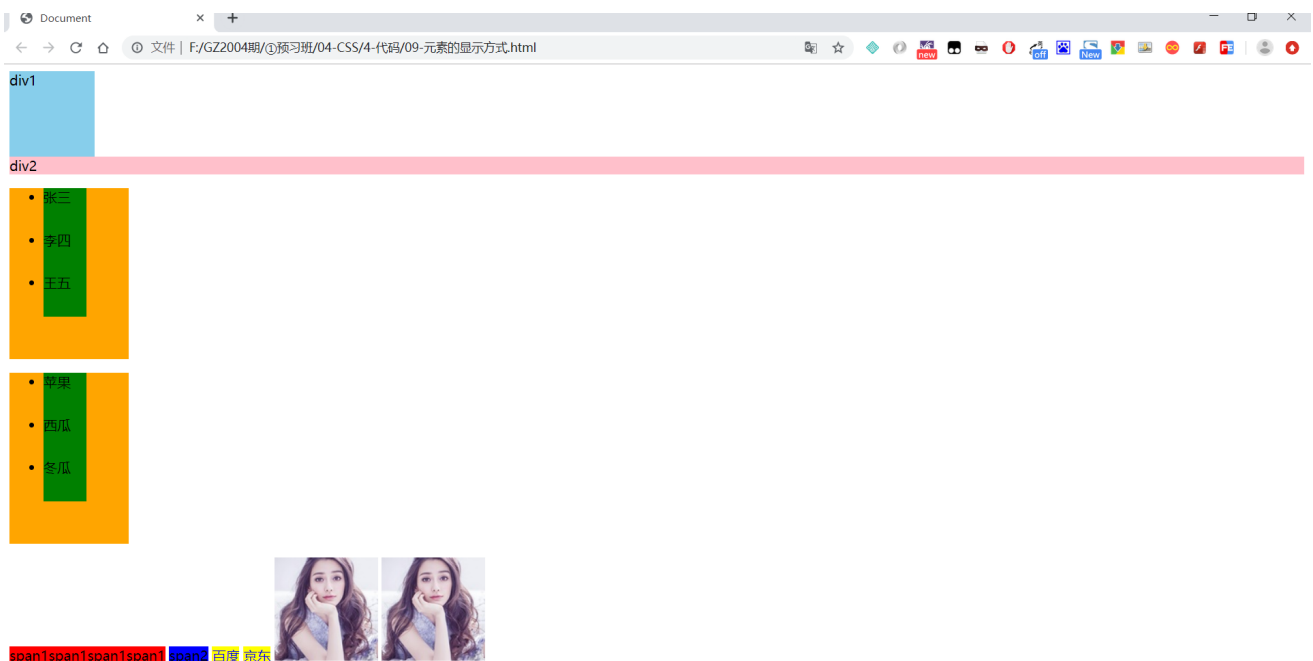
<!-- 常见行内元素 span a i em b strong u ins del s -->

```
<span style="background:red">span1span1span1span1</span>
<span style="background:blue">span2</span>
<a href="#">百度</a>
<a href="#">京东</a>
```

<!-- 行内块元素,同时具有行内元素的部分特点,又具有块元素的部分特点 一行可以显示多个 CSS可以设置宽度高度 默认宽度还是由内容决定 -->

```
<!-- 常见行内块元素 img input -->


</body>
</html>
```



元素显示方式的转换(display属性) 需掌握,重点

转成什么元素以后,就会具有什么元素的特点,比如转成块元素以后,就可以独占一行,设置宽度高度

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:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<title>Document</title>
<style>
    /*我们CSS中提供了一个display的属性,可以改变元素的显示方式*/
    /*display:block;          转成块级元素*/
    /*display:inline;        转成行内元素*/
    /*display:inline-block; 转成行内块元素*/
    /*一句话: 转成什么元素,就会具有什么元素的特点*/

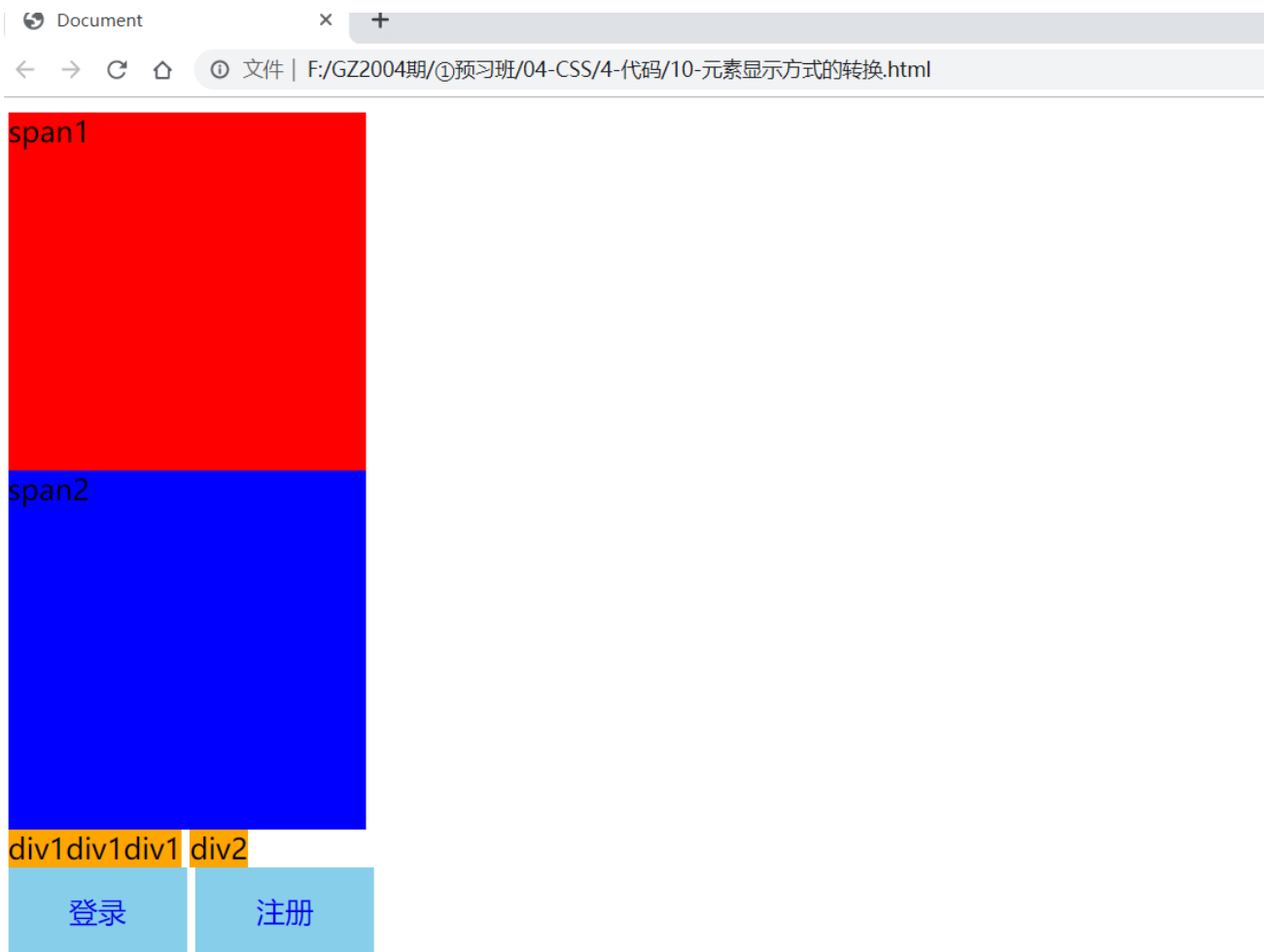
    span{
        width: 200px;
        height: 200px;
        /*display:block;      转成块级元素*/
        display: block;
    }
    .span1{
        background: red;
    }
    .span2{
        background: blue;
    }
    div{
        width: 100px;
        height: 100px;
        background: orange;
        /*display:inline;      转成行内元素*/
        display: inline;
    }
    a{
        background: skyblue;
        width: 100px;
        height: 50px;
        /*display:inline-block; 转成行内块元素*/
        display: inline-block;
        text-align: center;
        line-height: 50px;
        text-decoration: none;
    }
</style>
</head>
<body>
    <span class="span1">span1</span>
    <span class="span2">span2</span>

    <div>div1div1div1</div>
    <div>div2</div>

    <br/>

    <a href="#">登录</a>
    <a href="#">注册</a>
</body>
</html>

```



元素显示与隐藏

1. 隐藏元素: **display: none;**
2. 显示元素: **display: block;**

举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    .father{
      width: 500px;
      height: 500px;
      background: skyblue;
    }
    .father .son1{
      width: 100px;
```

```

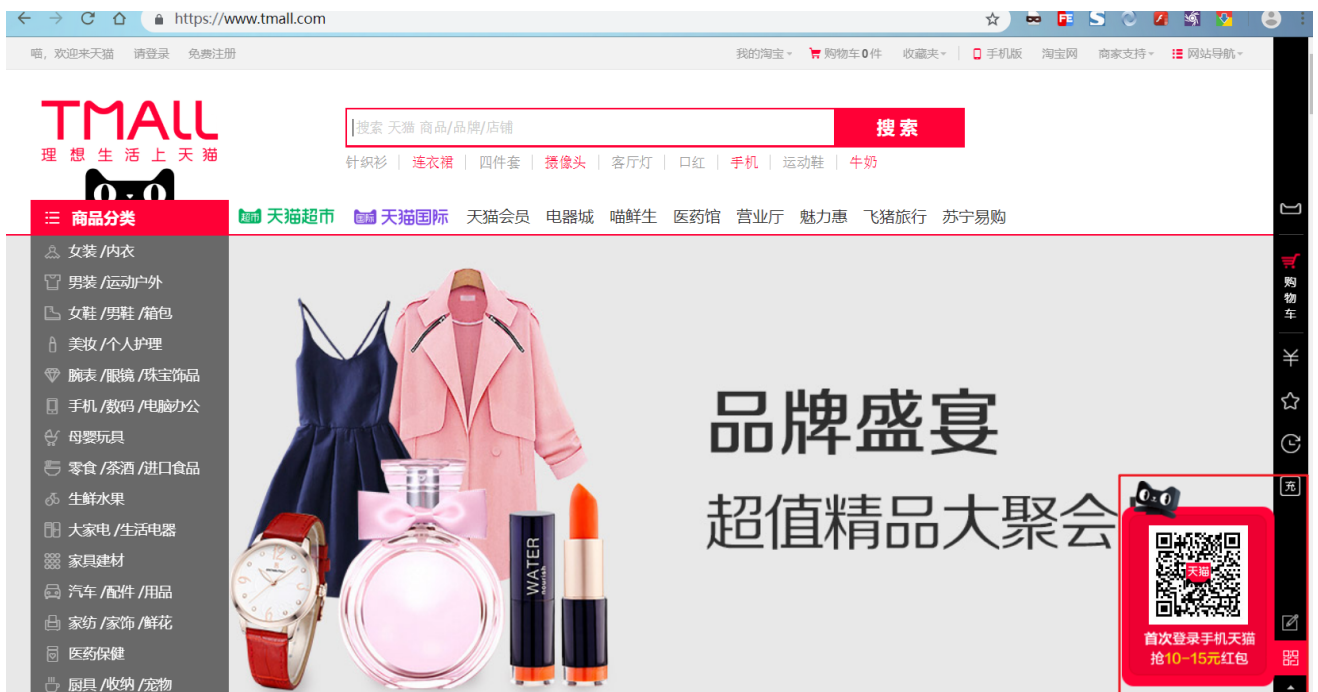
        height: 100px;
        background: red;
        /*隐藏元素使用的是 display:none; */
        /*整个元素被隐藏了,不占位置,后面会上来*/
        display: none;
    }

    /*鼠标移上.father以后, .son1对应的样式 */
    .father:hover .son1{
        /*显示元素 display:block;*/
        display: block;
    }

    .father .son2{
        width: 200px;
        height: 200px;
        background: orange;
    }
</style>
</head>
<body>
    <!-- >代表嵌套关系    +代表同级关系(兄弟关系) -->
    <!-- .father>.son1+.son2 -->
    <div class="father">
        <div class="son1">father的里面son1</div>
        <div class="son2">father的里面son2</div>
    </div>
</body>
</html>

```

线上案例:天猫侧边栏 二维码显示



天猫导航栏 中 鼠标移上网站导航 显示内容 鼠标移出隐藏内容

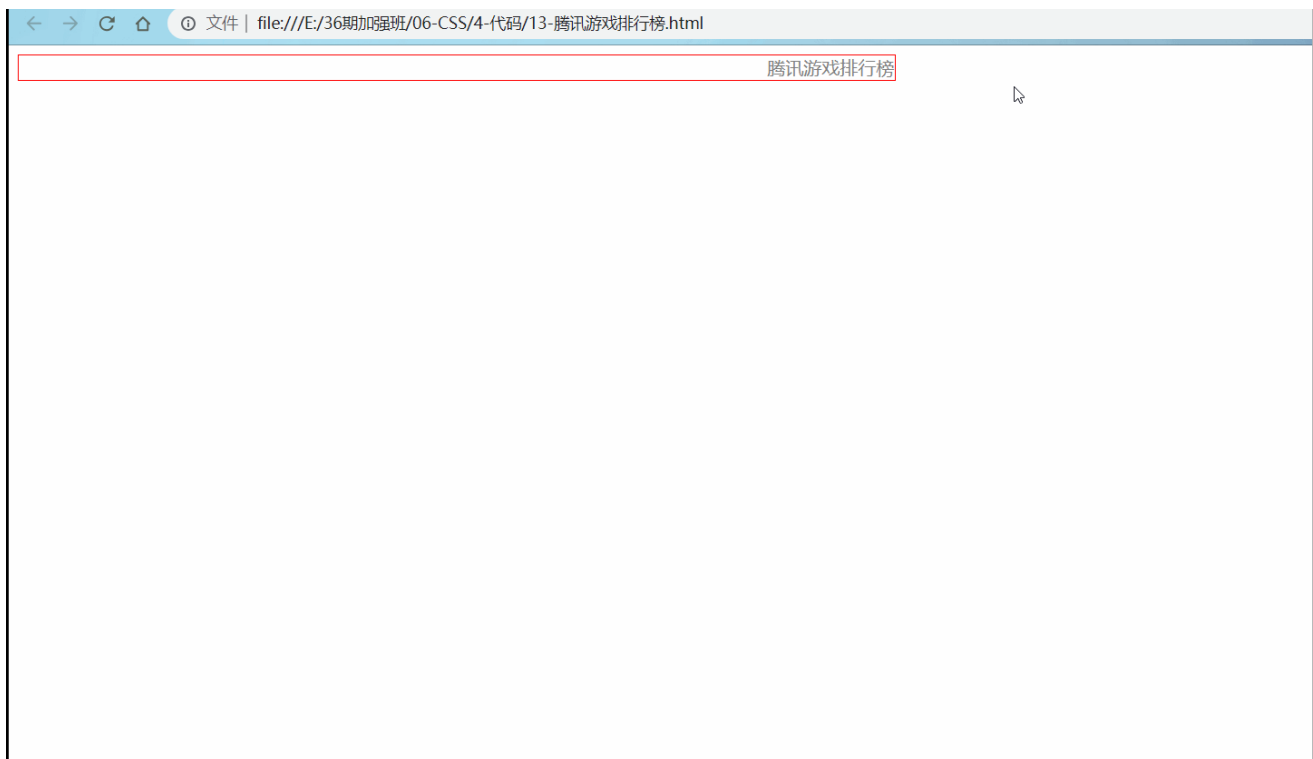


练习:腾讯排行榜-显示隐藏

边框样式: border:1px solid red;

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*写样式,我们先做整体,再做细节*/
    .box{
      border:1px solid red;
      width: 772px;
      text-align: right;
    }
    a{
      text-decoration: none;
      color:#808082;
    }
  </style>
</head>
<body>
  <div class="box">
    <a href="#">腾讯排行榜</a>
  </div>
</body>
</html>
```

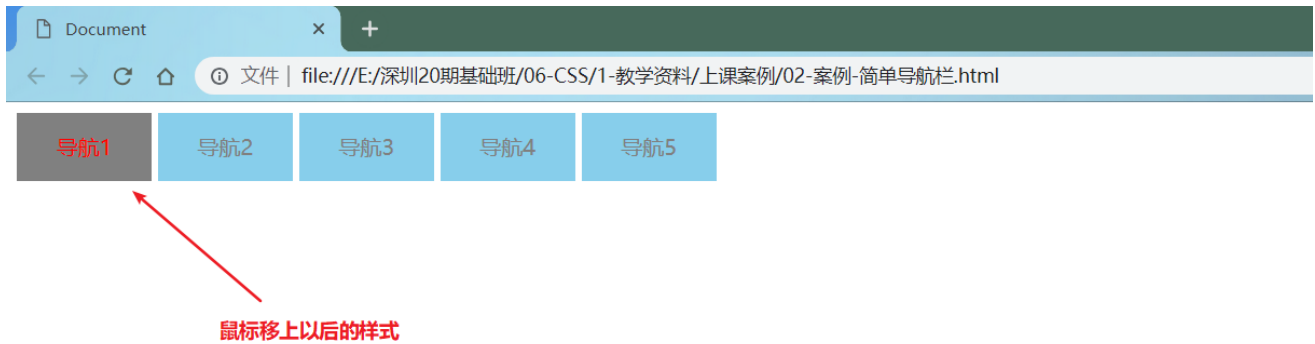
```
    }
    .box a img{
        /*隐藏元素*/
        display: none;
    }
    /*鼠标移上a标签以后,img对应的样式*/
    .box a:hover img{
        /*显示元素*/
        display: block;
    }
</style>
</head>
<body>
    <div class="box">
        <a href="#">
            腾讯游戏排行榜
            
        </a>
    </div>
</body>
</html>
```



练习:简单导航案例

要求: 使用a标签加上空链接以后, 完成以下效果 背景颜色属性是background-color

小盒子宽度100 高度50 背景颜色天空蓝,鼠标移上a链接以后,背景颜色变灰色 字体颜色变红



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    a{
      background: #87ceeb;
      width: 100px;
      height: 50px;
      /*使用display转换行内块元素 不独占一行,可以设置宽度高度*/
      display: inline-block;
      text-decoration: none;
      color:#808080;
      text-align:center;
      line-height: 50px;
    }
    a:hover{
      color:red;
      background: #808080;
    }
  </style>
</head>
<body>
  <a href="#">导航1</a>
  <a href="#">导航2</a>
  <a href="#">导航3</a>
  <a href="#">导航4</a>
  <a href="#">导航5</a>
</body>
</html>
```

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:

`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,比如0.8代表80%的透明度,如果设置的是0.8,可以写成.8 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:

`background-image` : `none` 或 `url(图片所在路径)`

参数:

`none`: 无背景图 (默认的) `url`: 背景图像地址

`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) 可以和 `background-color` 连用。

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：

```
background-repeat : repeat 或 no-repeat 或 repeat-x 或 repeat-y
```

取值说明：

- repeat：默认值,背景图像在纵向和横向都平铺设置背景图片时，默认把图片在水平和垂直方向平铺以铺满整个元素。
- **no-repeat**： 背景图像不平铺 这个取值用的比较多
- repeat-x： 背景图像在横向平铺
- repeat-y： 背景图像在纵向平铺

注意: 如果背景图片不重复平铺地话，背景图片覆盖不到的地方都会被背景色填充。 如果有背景图片平铺，则会覆盖背景颜色。

举例：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    body{
      background-color:skyblue;
    }
    .div1{
      width: 100px;
      height: 100px;
      /*background-color设置元素的背景颜色,取值 英文颜色单词 rgb 十六进制 rgba*/
      background-color:red;
      /*background-color: rgb(255,0,0);*/

      /*rgba其中a是alpha表示透明度的意思,取值0~1之间所有的数,包括0跟1,0代表完全透明,1代表完全不透明*/
      /*background-color:rgba(255,0,0,1);*/
      /*background-color:rgba(255,0,0,0.5);*/
      /*background-color:rgba(255,0,0,0.2);*/
      /*background-color:rgba(255,0,0,0.23456);*/
```

```

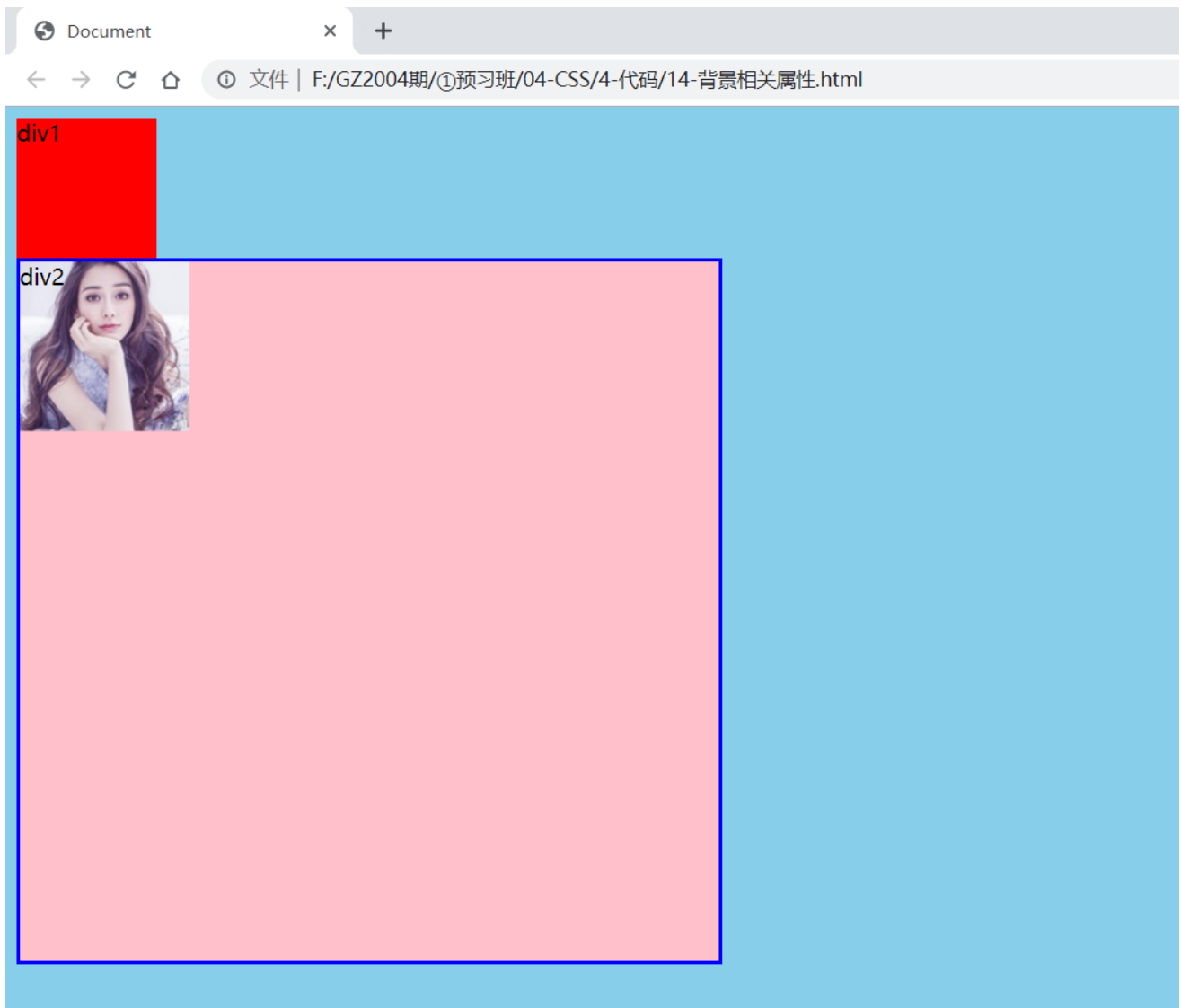
        /*rgba中的a写小数的时候,可以省略0,比如0.7就可以.7*/
        /*background-color: rgba(255,0,0,0.7);*/
        /*background-color: rgba(255,0,0,.7);*/

        /*background-color:#00ff00;*/
    }
    .div2{
        width: 500px;
        height: 500px;
        /*设置边框线*/
        border:2px solid blue;
        /*background-image设置元素背景图片*/
        /*background-image: url(图片所在路径);*/
        background-image: url(images/1.jpg);

        /*
            background-repeat设置背景图片的平铺方式
            取值:
            repeat-x    水平平铺,垂直不平铺
            repeat-y    水平不平铺,垂直平铺
            repeat      默认值,水平垂直都平铺
            no-repeat   水平垂直都不平铺
        */
        /*background-repeat: repeat-x;*/
        /*background-repeat: repeat-y;*/
        /*background-repeat: repeat;*/
        background-repeat: no-repeat;

        /*背景图片会覆盖背景颜色*/
        background-color: pink;
    }
</style>
</head>
<body>
    <!-- div[class="div$"]{div$}*5 -->
    <div class="div1">div1</div>
    <div class="div2">div2</div>
</body>
</html>

```



背景定位(background-position) 重点难点

语法:

`background-position` : 水平位置 垂直位置;

`background-position` : `length` `length`;

`background-position` : 百分比 百分比 ;

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

英文单词表示法:

水平位置: `left` `center` `right`

垂直位置: `top` `center` `bottom`

百分比表示法: 50% 50%; 可以是正数也可以是负数

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

水平方向: 正数向右走, 负数向左走

垂直方向: 正数向下走, 负数向上走

参数:

length: 百分数 | 由浮点数字和单位标识符组成的长度值。请参阅长度单位 position: top | center | bottom | left | center | right 方位名词

说明:

设置或检索对象的背景图像位置。必须先指定background-image属性。默认值为: (0% 0%)。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%。第二个值将用于纵坐标。

注意:

- 必须先指定background-image属性
- 背景图片默认是从元素的左上角开始显示的
- 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
- 实际工作用的最多的, 就是背景图片居中显示了。

举例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    .div1{
      width: 500px;
      height: 500px;
      border: 2px solid blue;
      background-color: pink;
      background-image: url(images/1.jpg);
      background-repeat: no-repeat;
      /*默认背景图片是从元素的左上角开始显示*/
      /*我们可以使用background-position改变背景图片显示的图片*/
      /*background-position: 水平位置 垂直位置;*/
      /*取值方式有三种*/
      /*
        第一种: 英文位置单词
        水平left center right
```

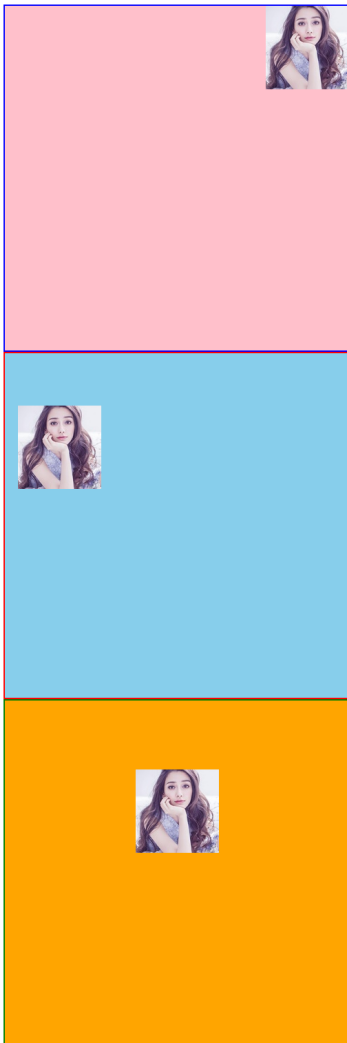
垂直top center bottom

这种方式最多表示9个位置

```
*/
/*background-position: left top;*/
background-position: right top;
/*background-position: center center;*/
/*background-position: left center;*/
}
.div2{
  width: 500px;
  height: 500px;
  border:2px solid red;
  background-color:skyblue;
  background-image: url(images/1.jpg);
  background-repeat: no-repeat;
  /*第二种取值方式 百分比 这种方式用的不多 正负值都可以 正值水平往右,垂直往下;负值相反*/
  /*background-position: 0% 0%;*/
  /*background-position: 100% 0%;*/
  /*background-position: 100% 50%;*/
  background-position: 5% 20%;
}
.div3{
  width: 500px;
  height: 500px;
  border:2px solid green;
  background-color:orange;
  background-image: url(images/1.jpg);
  background-repeat: no-repeat;
  /*第三种取值方式 数值带px单位 这种方式用的最多 正负值都可以 正值水平往右,垂直往下;负值相反
*/
  /*background-position: 0 0;*/
  /*background-position: 0px 0px;*/
  /*background-position: 10px -50px;*/

  /*三种取值方式可以混写 比如background-position:center 100px;*/
  background-position: center 100px;
}

</style>
</head>
<body>
  <div class="div1"></div>
  <div class="div2"></div>
  <div class="div3"></div>
</body>
</html>
```



背景简写(background) 用得多

background属性的值的**书写顺序官方并没有强制标准的**。为了可读性，建议大家如下写：

background:背景颜色 背景图片 背景平铺 背景滚动 背景位置

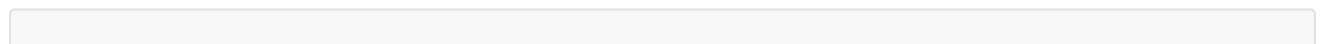
```
background: red url(image.jpg) repeat-y 50% 0 ;
```

注意:

如果先设置的背景单独属性,后面又设置了背景简写属性,背景简写属性会把背景单独属性都干掉(单独属性设置的会失效)

所以,建议先写简写属性,再写单独属性;不要先写单独属性,再写简写属性

举例:




```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    /*background是背景的简写属性,可以设置之前讲的背景属性,属性取值使用"空格"隔开 不需要顺序,可以省略某个或者某几个*/
    div{
      width: 500px;
      height: 300px;
      border:5px solid orange;

      /*background-color: lightpink;*/
      /*background-image:url(images/1.jpg);*/
      /*background-repeat: no-repeat;*/
      /*background-position: right bottom;*/

      /*background:lightpink url(images/1.jpg) no-repeat right bottom;*/
      /*background: url(images/1.jpg);*/
      /*background: pink;*/

      /*注意:*/
      /*如果先设置的背景单独属性,后面又设置了背景简写属性,背景简写属性会把背景单独属性都干掉(单独属性设置的会失效)*/
      /* background-image: url(images/1.jpg);
      background-repeat: no-repeat;
      background:pink; */

      /*background-color:pink;*/
      /*background: left top;*/

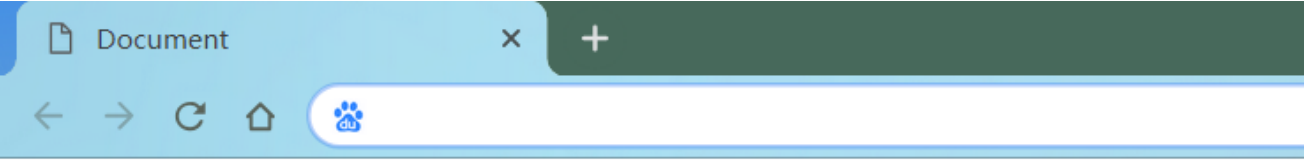
      /*所以,建议先写简写属性,再写单独属性;不要先写单独属性,再写简写属性*/
      background:red;
      background-image: url(images/1.jpg);
      background-position: right center;
      background-repeat: no-repeat;
    }
  </style>
</head>
<body>
  <div>div的内容</div>
</body>
</html>

```

背景属性总结

属性名	属性取值	功能
background-color	英文颜色单词,十六进制,rgb,rgba	设置背景颜色
background-image	url(图片所在路径) 或 none无背景图(默认的)	设置背景图片
background-repeat	repeat 或 no-repeat 或 repeat-x 或 repeat-y	设置背景平铺
background-position	长度值或百分比或方向名词	设置背景定位
background	可以同时设置背景颜色 背景图片 背景平铺 背景滚动 背景位置 没有书写顺序	背景属性简写,可以同时设置背景颜色 背景图片 背景平铺 背景滚动 背景位置 方便使用

案例-jack



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    div{
      display: inline-block;
      background-image:url(images/abcd.jpg);
    }
  </style>
</head>
<body>
  <div>
    JACK
  </div>
</body>
</html>
```

```
        background-repeat:no-repeat;
    }
    .j{
        width: 104px;
        height: 106px;
        background-position: -388px -146px;
    }
    .a{
        width: 107px;
        height: 111px;
        background-position: 0px -7px;
    }
    .c{
        width: 104px;
        height: 111px;
        background-position: -240px -8px;
    }
    .k{
        width: 106px;
        height: 110px;
        background-position: right -143px;
    }
    /*background-position背景定位的时候,取值方式可以混着写*/
</style>
</head>
<body>
    <div class='j'></div>
    <div class='a'></div>
    <div class='c'></div>
    <div class='k'></div>
</body>
</html>
```

今日总结

xmind总结知识点

元素的显示方式,分为几种? 三种

块级元素:独占一行,可以设置宽度高度,默认宽度是父元素的100%

行内元素:不独占一行,设置宽度高度无效,默认宽度是由内容决定

行内块元素:不独占一行,可以设置宽度高度,默认宽度是由内容决定

元素显示方式之前的转换 display

display:block;

display:inline;

display:inline-block;

display:none; 隐藏元素

