

HTML

字符集问题

```
1 <html>
2   <head>
3     <meta charset="GBK">
4     <title>你好,世界123</title>
5   </head>
6   <body>
7     你好,世界?
8 </body>
9 </html>
```

```
1 <html>
2   <head>
3     <meta charset="UTF-8" />
4     <title>百度一下,你就知道123</title>
5   </head>
6   <body>
7     能看到我吗?
8 </body>
9 </html>
```

DTD文档声明

主要作用是告诉浏览器的解析器使用哪种HTML规范或者XHTML规范来解析页面。

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8" />
5     <title>百度一下,你就知道123</title>
6   </head>
7   <body>
8     能看到我吗?
9   </body>
10 </html>
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8" />
5     <title>百度一下,你就知道123</title>
6     <style type="text/css">
7       p {
8         font-size: 50px;
9         color: red;
10        font-family: "微软雅黑";
```

```

11         }
12     </style>
13 </head>
14 <body>
15     <font size="7" color="red" face="微软雅黑">能看到我吗?</font>
16     <u>我是文字</u>
17     <p>我是PPPP</p>
18 </body>
19 </html>

```

H标签-hr标签-p标签

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>H系列标签和P标签和Hr标签</title>
6 </head>
7 <body>
8     <h1>我是标题1</h1>
9     <h2>我是标题2</h2>
10    <h3>我是标题3</h3>
11    <h4>我是标题4</h4>
12    <h5>我是标题5</h5>
13    <h6>我是标题6</h6>
14    <h7>我是标题7</h7>
15    我是普通文本
16
17    <hr />
18    <hr>
19
20    <p>我是一段文本</p>
21    <p>我是一段文本</p>
22    我是一段普通文本
23    我是一段普通文本
24
25 </body>
26 </html>

```

img标签

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>img标签</title>
6 </head>
7 <body>
8 <!--
9 1. img标签中的img其实是英文image的缩写
10 所以img标签的作用，就是告诉浏览器我们需要显示一张图片
11

```

```
12 2.img标签格式：<img src="">
13 其实img标签中的src是英文source的缩写
14 所以img标签中的src就是用来告诉img标签，需要显示的图片名称
15
16 3.注意点
17 - 和H系列标签/p标签还有Hr标签不一样，img标签不会独占一行
18 - 如果我们手动指定了img标签显示的图片的宽度和高度，有可能导致图片变形，那么如果又想指定
    宽度和高度，又不想让图片变形。我们可以只指定宽度和高度其中的一个值即可
19 - 只要指定了高度，系统会自动根据高度计算出宽度，只要指定了宽度，系统会自动根据宽度计算出高
    度，并且都是等比拉伸的，也就是说不会变形
20
21 4.img中的其它属性
22 width：宽度
23 height：高度
24 所以在img标签中width/height这两个属性的作用，就是用来告诉img标签将来需要显示的图片有多
    宽有多高
25 如果img标签没有指定需要显示的图片的宽高，那么系统会按照图片默认的宽高来显示
26 如果img标签指定的宽高，那么系统会按照指定的宽高来显示
27
28 title：用于告诉浏览器，当鼠标悬停在图片上时，需要弹出的描述框中显示什么内容
29
30 alt其实是英文alternate的缩写，它的作用就是用于告诉浏览器，当需要显示的图片找不到时显示什
    么内容
31
32 -->
33
34 
35
36 
37
38 
39
40 
41 
42
43 
44
45 
46 </body>
47 </html>
```

br标签

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>br标签</title>
6 </head>
7 <body>
8 <!--
9 br标签，如何在HTML中换行？ 可以使用br标签
10 1. br标签作用：换行
11 2. br标签格式：<br>
12 3. br标签的注意点：
```

```

13 3.1多个br标签可以连续使用，使用了多少个br标签就会换多少行
14 3.2由于HTML的作用就是用来给文本添加语义，而br标签的语义是不另起一个段落换行，而在企业开发
    中一般情况下需要换行都是因为需要另起一个段落，所以在企业开发中很少使用br标签
15 -->
16
17 <!--只设置了img标签的src属性<br><br><br>-->
18 <!--<br>-->
19 <!--设置了img标签的src和width以及height属性<br>-->
20 <!--<br>-->
21 <!--设置了img标签的src和width以及height，但是宽高不是等比拉伸的<br>-->
22 <!--<br>-->
23 <!--设置了img标签的src和height属性<br>-->
24 <!--<br>-->
25 <!--设置了img标签的src和width属性<br>-->
26 <!--<br>-->
27 <!--设置了img标签的src和width以及title属性<br>-->
28 <!--<br>-->
29 <!--设置了img标签的src和width以及alt属性<br>-->
30 <!--<br>-->
31
32
33 <p>只设置了img标签的src属性</p>
34 <p></p>
35 <p>设置了img标签的src和width以及height属性</p>
36 <p></p>
37 <p>设置了img标签的src和width以及height，但是宽高不是等比拉伸的</p>
38 <p></p>
39 <p>设置了img标签的src和height属性</p>
40 <p></p>
41 <p>设置了img标签的src和width属性</p>
42 <p></p>
43 <p>设置了img标签的src和width以及title属性</p>
44 <p></p>
45 <p>设置了img标签的src和width以及alt属性</p>
46 <p></p>
47
48 </body>
49 </html>

```

路径问题

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>路径问题</title>
6 </head>
7 <body>
8
9 <!--
10 路径问题
11 其实想给src属性赋值有两种方式

```

12 1. 通过相对路径赋值(掌握)

13 相对路径就是每次都从.html文件所在的文件夹开始查找，我们称之为相对路径

14

15 1.1同级

16 同级就是"图片"和".html文件"存储在同一个文件夹中

17 格式：src="QRCode.jpg"

18 含义：在.html文件所在的文件夹中查找名称叫做QRCode.jpg的图片

19

20 1.2下级

21 下级就是"存储图片的文件夹"和".html文件"在同一个文件夹中

22 格式：src="images/QRCode.jpg"

23 含义：在.html文件所在的文件夹中找到名称叫做images的文件夹

24 然后再在images文件夹中找到名称叫做QRCode.jpg的图片

25

26 1.3上级

27 上级就是"存储图片的位置"和存"存储代码的文件夹"在同一个文件夹中

28 格式：src="../QRCode.jpg"

29 含义：在.html文件所在的文件夹中找到这个文件夹的上一级文件夹，然后再在上一级文件夹中找到名称叫做QRCode.jpg

30 其中../代表找到当前文件夹的上一级文件夹

31

32 2. 通过绝对路径赋值(了解)

33 绝对路径就是每次都从指定的盘符开始查找

34

35 格式：src="C:\Users\Jonathan_Lee\Desktop\HTML基础\QRCode.jpg"

36 含义：在C盘下找到Users文件夹，然后在Users文件夹中找到Jonathan_Lee文件夹，然后在Jonathan_Lee文件夹中找到Desktop文件夹，然后在Desktop文件夹中找到HTML基础文件夹，然后在HTML基础文件夹中找到名称叫做QRCode.jpg的图片

37

38 注意：

39 1. 路径中不要出现中文，否则可能出现未知问题

40 2. 如果是通过"相对路径"来指定图片，不能跨盘符

41 2.1例如.html文件在C盘，那么不能去查找D盘图片

42 -->

43

44

45

46

47 </body>

48 </html>

a标签

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>10-a标签基本使用</title>
6 </head>
7 <body>
8 <!--
9 什么是a标签？
10 a标签的作用：就是用于控制页面与页面之间跳转的
11 a标签的格式：<a href="指定需要跳转的目标界面">需要展现给用户查看的内容</a>
12
```

```

13 a标签中有一个叫做target属性，这个属性的作用就是专门用于控制如何跳转
14 _self：用于在当前选项卡中跳转，也就是不新建界面跳转。默认就是_self
15 _blank：用于在新的选项卡中跳转，也就是新建界面跳转
16
17 a标签中还有一个属性，叫做title。a标签中的title和img标签中的title一样，都是用来控制鼠标
   悬停时显示的提示文本的
18
19 注意点：
20 1.a标签不仅可以让文字可以点击，还可以让图片也能够被点击
21 2.一个a标签必须有一个href属性，否则a标签不知道要跳转到什么地方
22 3.如果通过a标签的href属性指定一个URL地址，那么必须在地址前面加上http://或https://
23 4.a标签的href属性除了可以指定一个网络地址以外，还可以指定一个本地地址
24 -->
25
26 <a href="https://www.nuomi.com/?cid=002540">糯米</a>
27
28 <a href="https://binjiejw.tmall.com/category-1080022346.htm?
   search=y&parentCatId=1080019249&parentCatName=%A1%A4?catName=??%A1%A4???????
   spm=a220o.1000855.w5001-
   10945243352.20.Fq84rf&scene=taobao_shop&ali_trackid=30_e27f91873d121ea29b062
   a117ba2991b&spm=a21bo.50862.201862-4.1.9GysHu#bd">
29     
30 </a>
31
32 <a href="08-br标签.html">08-br标签.html</a>
33 <a href="test/demo/10-路径练习.html">10-路径练习.html</a>
34
35 <a href="https://www.nuomi.com/?cid=002540" target="_self">糯米</a>
36 <a href="https://www.nuomi.com/?cid=002540" target="_blank">糯米</a>
37
38
39 <a href="https://www.nuomi.com/?cid=002540" target="_blank" title="江哥提示：
   会跳转到百度糯米的首页">糯米</a>
40
41 </body>
42 </html>

```

base标签

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>11-base标签</title>
6     <base target="_blank">
7 </head>
8 <body>
9 <!--
10 base标签就是专门用来统一的指定当前网页中所有的超链接(a标签)需要如何打开
11
12 注意点：
13 1.base标签必须写在head标签的开始标签和结束标签之间
14 2.如果既在base中指定了target又在a标签中指定了target，那么浏览器会按照a标签中指定的来执行
15 -->
16 <a href="https://www.nuomi.com/?cid=002540" target="_self">糯米</a>

```

```
17 <a href="http://news.baidu.com/">新闻</a>
18 <a href="https://www.hao123.com/">hao123</a>
19 <a href="http://map.baidu.com/">地图</a>
20 <a href="http://v.baidu.com/">视频</a>
21 <a href="http://tieba.baidu.com/">贴吧</a>
22
23 </body>
24 </html>
```

假链接

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>12-假链接</title>
6 </head>
7 <body>
8 <!--
9 1.什么是假链接?
10 就是点击之后不会跳转的链接我们称之为假链接
11
12 2.假链接存在的意义:
13 在企业开发前期, 其它界面都没有写出来, 那么我们就不知道应该跳转到什么地方, 所以就只能使用假
   链接来代替. 当项目后期其它界面都已经完成时再将假链接体会为真链接
14
15 3.假链接的格式:
16 1.#
17 2.javascript:
18
19 4.两者之间的区别:
20 #的假链接会自动回到网页的顶部, 而javascript:的假链接不会自动回到网页顶部
21 -->
22 <h1>我是顶部</h1>
23 <a href="#">点我丫1</a>
24 <a href="javascript:">点我丫2</a>
25 </body>
26 </html>
```

锚点

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>13-锚点</title>
6 </head>
7 <body>
8
9 <!--
10 1.要想通过a标签跳转到指定的位置, 那么必须告诉a标签一个独一无二的身份证号码, 这样a标签才能
   在当前界面中找到需要跳转到的目标位置
```

```

11
12 2. 如果和HTML中的标签绑定一个独一无二的身份证号码呢？
13 在HTML中，每一个标签都有一个名称叫做id的属性，这个属性就是用来给标签指定一个独一无二的身
    身份证号码的
14
15 3. 所以要想实现通过a标签跳转到指定的位置分为两步
16 3.1 给目标位置的标签添加一个id属性，然后指定一个独一无二的值
17 3.2 告诉a标签你需要跳转到的目标标签对应的独一无二的身份证号码是多少
18 格式：
19 <a href="#center">跳转到中部</a>
20 <h2 id="center">我是中部</h2>
21
22 注意点：
23 1. 通过我们的a标签跳转到指定的位置，是没有过度动画的，是直接一下子就跳转到了指定位置
24 2. a标签除了可以跳转到当前界面的指定位置以外，还可以在跳转到其它界面的时候直接跳转到其它界
    面的指定位置
25 格式：
26 <a href="13-锚点测试界面.html#bottom" target="_blank">跳转到锚点测试界面</a>
27 <h2 id="bottom">我是锚点测试界面33333</h2>
28 -->
29 <a href="13-锚点测试界面.html#bottom" target="_blank">跳转到锚点测试界面</a>
30 <h2>我是顶部</h2>
31 <a href="#center">跳转到中部</a>
32 <h2 id="center">我是中部</h2>
33 <h2>我是底部</h2>
34 </body>
35 </html>

```

无序列表

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>14-无序列表</title>
6      <style type="text/css">
7          ul {
8              list-style: none;
9          }
10         li {
11             float: left;
12             background-color: yellow;
13             width: 150px;
14             height: 50px;
15             text-align: center;
16             line-height: 50px;
17         }
18     </style>
19 </head>
20 <body>
21 <!--
22 1. 什么是列表标签？
23 列表标签的作用：给一堆数据添加列表语义，也就是告诉搜索引擎告诉浏览器这一堆数据是一个整体
24
25 2. HTML中列表标签的分类

```



```
26 2.1无序列表(最多)(unordered list)
27 2.2有序列表(最少)(ordered list)
28 2.3定义列表(其次)(definition list)
29
30 3.无序列表作用：
31 给一堆数据添加列表语义，并且这一堆数据中所有的数据都没有先后之分
32
33 什么叫有先后之分？
34 例如：排行榜
35 什么叫没有先后之分？
36 例如：中国有哪些城市
37
38 4.无序列表格式：
39 <ul>
40     <li>需要显示的条目内容</li>
41 </ul>
42 li其实是英文list item的缩写
43 list是列表的意思
44 item是条目的意思
45 所以结合起来就是 列表条目的意思
46
47 5.注意点：
48 1.一定要记住ul标签是用来给一堆数据添加列表语义的，而不是用来给他们添加小圆点的
49 2. ul标签和li标签是一个整体，是一个组合。所以一般情况下ul标签和li标签都是一起出现，不会
    单个出现。也就是说不会单独使用一个ul标签或者单独使用一个li 标签，都是结合在一起使用
50 3.由于ul标签和li标签是一个组合，所以ul标签中不推荐包含其它标签，也就是说以后你在ul标签中
    只会看到li标签
51
52 无序列表应用场景：
53 1.新闻列表
54 2.商品列表
55 3.导航条
56
57 -->
58
59 <!--
60 无序列表中的li标签中除了可以添加其它标签来丰富我们的界面以外，还可以添加ul标签来丰富我的界
    面
61 也就是说ul中有li，li中又可以有ul
62
63 在webstorm中如何快速编写一个ul的格式
64 ul>li
65 含义：生成一对ul标签，然后在这对ul标签中再生成一对li标签
66
67 ul>li*3
68 含义：生成一对ul标签，然后在这对ul标签中再生成3对li标签
69 -->
70
71 <h2>中国的城市有哪些</h2>
72 <ul>
73     <li>广州</li>
74     <li>北京</li>
75     <li>上海</li>
76     <li>武汉</li>
77 </ul>
78 </body>
79 </html>
```

有序列表

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>18-有序列表</title>
6 </head>
7 <body>
8 <!--
9 什么是有序列表？
10 有序列表的作用：给一堆数据添加列表语义，并且这一堆数据中所有的数据都有先后之分
11
12 有序列表格式：
13 <ol>
14     <li></li>
15 </ol>
16
17 其它用法和ul都差不多，也就是应用场景/注意点都和ul差不多
18
19 -->
20
21 <ol>
22     <li>演员</li>
23     <li>小丑</li>
24     <li>女人不应该让男人太累</li>
25     <li>男人不应该让女人流泪</li>
26 </ol>
27 </body>
28 </html>
```

定义列表

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>19-定义列表</title>
6 </head>
7 <body>
8 <!--
9 1. 定义列表的作用：
10 1.1 给一堆数据添加列表语义
11 1.2 先通过dt标签定义列表中的所有标题，然后再通过dd标签给每个标题添加描述信息
12
13 2. 定义列表的格式：
14 <dl>
15     <dt></dt>
16     <dd></dd>
17     <dt></dt>
18     <dd></dd>
19 </dl>
```

```
20 其实dt和dd都是英文的缩写
21 dt是英文definition title的缩写，所以dt的含义就是用来定义列表中的标题
22 dd是英文definition description的缩写，所以dd的含义就是用来定义标题对应的描述的
23
24 3. 定义列表的应用场景
25 1. 做网站尾部的相关信息
26 2. 做图文混排
27
28 4. 定义列表的注意点
29 4.1和ul/ol一样，dl和dt/dd是一个整体，所以他们一般情况下不会单独出现，都是一起出现
30 4.2和ul/ol一样，由于dl和dt/dd是一个组合标签，所以dl中建议只放dt和dd标签
31 4.3一个dt可以没有对应的dd,也可以有多个对应的dd，但是无论有或者没有dd都不推荐使用。
32 推荐使用一个dt对应一个dd
33 4.4和li标签一样，当需要丰富界面时，我们可以在dt和dd标签中继续添加其它标签
34 -->
35
36 <dl>
37     <dt>北京</dt>
38     <dd>中国的首都</dd>
39     <dd>雾都，雾霾比较严重</dd>
40     <dt>上海</dt>
41     <dd>富人集中地</dd>
42 </dl>
43 </body>
44 </html>
```

定义列表练习

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>20-定义列表练习(了解)</title>
6      <style type="text/css">
7          *{ margin:0; padding:0; border: 0;}
8          dl{ width:400px; height:200px; background:#ccc; margin:100px auto;}
9          dl dt{ width:150px; height:200px; float:left;}
10         dl dd{ width:250px; height:200px; float:right;}
11         dl dd h2{ line-height:50px; color:orange; margin:0 10px; font-size:16px;}
12         dl dd p{ line-height:19px; color:#000; font-size:12px; margin:0 10px;}
13     </style>
14 </head>
15 <body>
16
17 <!--
18 dl>dt+dd
19
20 <dl>
21     <dt></dt>
22     <dd></dd>
23 </dl>
24 -->
25 <dl>
```

```

26     <dt>
27         
28     </dt>
29     <dd>
30         <h2>李南江（NJ）</h2>
31         <p>多年iOS、Android、HTML开发及教学经验，对NativeApp、HybridApp、WebApp开
        发有独到见解和深入的研究，除此之外李老师还精通JavaScript、AngularJS、NodeJS、Ajax、
        jQuery、Cordova、React Native等多种Web前端技术及Java、PHP等服务端技术</p>
32     </dd>
33 </dl>
34 </body>
35 </html>

```

表格标签

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>21-表格标签基本使用</title>
6  </head>
7  <body>
8  <!--
9  其实在过去表格标签用的非常非常的多，绝大多数的网站都是使用表格标签来制作的，也就是说表格标
    签是一个时代的代表
10
11  1. 什么是表格标签？
12  表格标签作用：用来给一堆数据添加表格语义
13  其实表格是一种数据的展现形式，当数据量非常大的时候，表格这种展现形式被认为是最为清晰的一种
    展现形式
14
15  2. 表格标签格式：
16  <table>
17      <tr>
18          <td>需要显示的内容</td>
19      </tr>
20  </table>
21  其实表格标签中的table代表整个表格，也就是一堆table标签就是一个表格
22  其实表格标签中的tr标签代表整个表格中的一行数据，也就是说一对tr标签就是表格中的一行
23  其实表格标签中的td标签代表表格中一行中的一个单元格，也就是说一对td标签就是一行中的一个单元
    格
24
25  3. 注意点
26  3.1 表格标签有一个边框属性，这个属性决定了边框的宽度。默认情况下这个属性的值是0，所以看不
    到边框
27  3.2 表格标签和列表标签一样，它是一个组合标签，所以table/tr/td要么一起出现，要么一起不出
    现，不会单个出现
28  -->
29
30  <!--要求写一个表格，这个表格中有2行3列-->
31  <table border="1">
32      <tr>
33          <td>1.1</td>
34          <td>1.2</td>
35          <td>1.3</td>

```

```

36     </tr>
37     <tr>
38         <td>2.1</td>
39         <td>2.2</td>
40         <td>2.3</td>
41     </tr>
42 </table>
43 </body>
44 </html>

```

表格标签的属性

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>22-表格标签的属性</title>
6  </head>
7  <body>
8      <!--
9      1. 宽度和高度的属性
10     可以给table标签和td标签使用
11     1.1 表格的宽度和高度默认是按照内容的尺寸来调整的，也可以通过给table标签设置width/height
        属性的方式来手动指定表格的宽度和高度
12     1.2 如果给td标签设置width/height属性，会修改当前单元格的宽度和高度，不会影响整个表格的宽
        度和高度
13
14     2. 水平对齐和垂直对齐的属性
15     其中水平对齐可以给table标签和tr标签和td标签使用
16     垂直对齐只能给tr标签和td标签使用
17     2.1 给table标签设置align属性，可以控制表格在水平方向的对齐方式
18     2.2 给tr标签设置align属性，可以控制当前行中所有单元格内容的水平方向的对齐方式
19     2.3 给td标签设置align属性，可以控制当前单元格中内容在说方向的对齐方式
20     注意点：如果td中设置了align属性，tr中也设置了align属性，那么单元格中的内容会按照td中设
        置的来对齐
21
22     2.4 给tr标签设置valign属性，可以控制当前行中所有单元格中的内容，在垂直方向的对齐方式
23     2.5 给td标签设置valign属性，可以控制当前单元格中的内容在垂直方向的对齐方式
24     注意点：如果td中设置了valign属性，tr中也设置了valign属性，那么单元格中的内容会按照td
        中设置的来对齐
25
26     3. 外边距和内边距的属性
27     只能给table标签使用
28     3.1. 外边距就是单元格和单元格之间的距离，我们称之为外边距
29     默认情况下单元格和单元格之间的外边距的距离是2px
30
31     3.2 内边距就是单元格的边框和文字之间的间隙，我们称之为内边距
32     默认情况下内边距是1
33
34
35     注意：以上讲解的内容仅仅作为了解，以后在企业开发中所有控制样式的都是通过CSS
36     -->
37
38 <!--编写一个2行2列的表格-->

```

```

39 <table border="1" width="500px" height="300px" align="right"
    cellspacing="12px" cellpadding="1">
40     <tr valign="bottom">
41         <td width="150px" height="50px">1.1</td>
42         <td valign="top">1.2</td>
43     </tr>
44     <tr align="center">
45         <td align="right">2.1</td>
46         <td>2.2</td>
47     </tr>
48 </table>
49 </body>
50 </html>

```

细线表格

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>23-细线表格</title>
6 </head>
7 <body>
8
9 <!--
10 在表格标签中想通过指定外边距为0来实现细线表格是不靠谱的，其实它是将2条线合并为了一条线，所以看上去很不舒服
11
12 细线表格的制作方式：
13 1. 给table标签设置bgcolor
14 2. 给tr标签设置bgcolor
15 3. 给table标签设置cellspacing = "1px"
16
17 注意点：
18 table标签和tr标签以及td标签都支持bgcolor属性
19 但是以上内容仅仅作为了解，因为样式以后都是通过css来控制
20 -->
21 <table bgcolor="black" cellspacing="1px">
22     <tr bgcolor="white">
23         <td bgcolor="red">1.1</td>
24         <td>1.2</td>
25     </tr>
26     <tr bgcolor="white">
27         <td>2.1</td>
28         <td>2.2</td>
29     </tr>
30 </table>
31 </body>
32 </html>

```

表格标签中的其他标签

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>24-表格标签的其它标签</title>
6 </head>
7 <body>
8 <!--
9 1. 表格标题
10 在表格标签中提供了一个标签专门用来设置表格的标题，这个标签叫做caption。只要将标题写在
    caption标签中，那么标题就会自动相对于表格的宽度居中
11
12 2.caption标签的注意点：
13 2.1caption一定要写在table标签中，否则无效
14 2.2caption一定要紧跟在table标签后面
15
16 3. 标题单元格标签
17 3.1. 在表格标签中提供了一个标签专门用来存储每一列的标题，这个标签叫做th标签，只要将当前列
    的标题存储在这个标签中就会自动居中+加粗文字
18
19 3.2. 到此为止我们就发现，其实表格中有两种单元格，一种是td，一种是th。td是专门用来存储数
    据的，th是专门用来存储当前列的标题的
20 -->
21
22 <table bgcolor="black" cellspacing="1px" width="800px" align="center">
23     <caption>
24         <h2>今日小说排行榜</h2>
25     </caption>
26     <tr bgcolor="#a9a9a9">
27         <th>排名</th>
28         <th>关键词</th>
29         <th>趋势</th>
30         <th>今日搜索</th>
31         <th>最近七日</th>
32         <th>相关链接</th>
33     </tr>
34     <tr bgcolor="white" align="center">
35         <td>1</td>
36         <td align="left">暴走大事件</td>
37         <td>
38             
39         </td>
40         <td>623557</td>
41         <td>4088311</td>
42         <td>
43             <a href="#">贴吧</a>
44             <a href="#">图片</a>
45             <a href="#">百科</a>
46         </td>
47     </tr>
48     <tr bgcolor="white" align="center">
49         <td>1</td>
50         <td align="left">暴走大事件</td>
51         <td>
52             
53         </td>
54         <td>623557</td>
55         <td>4088311</td>

```

```
56         <td>
57             <a href="#">贴吧</a>
58             <a href="#">图片</a>
59             <a href="#">百科</a>
60         </td>
61     </tr>
62     <tr bgcolor="white" align="center">
63         <td>1</td>
64         <td align="left">暴走大事件</td>
65         <td>
66             
67         </td>
68         <td>623557</td>
69         <td>4088311</td>
70         <td>
71             <a href="#">贴吧</a>
72             <a href="#">图片</a>
73             <a href="#">百科</a>
74         </td>
75     </tr>
76     <tr bgcolor="white" align="center">
77         <td>1</td>
78         <td align="left">暴走大事件</td>
79         <td>
80             
81         </td>
82         <td>623557</td>
83         <td>4088311</td>
84         <td>
85             <a href="#">贴吧</a>
86             <a href="#">图片</a>
87             <a href="#">百科</a>
88         </td>
89     </tr>
90     <tr bgcolor="white" align="center">
91         <td>1</td>
92         <td align="left">暴走大事件</td>
93         <td>
94             
95         </td>
96         <td>623557</td>
97         <td>4088311</td>
98         <td>
99             <a href="#">贴吧</a>
100             <a href="#">图片</a>
101             <a href="#">百科</a>
102         </td>
103     </tr>
104     <tr bgcolor="white" align="center">
105         <td>1</td>
106         <td align="left">暴走大事件</td>
107         <td>
108             
109         </td>
110         <td>623557</td>
111         <td>4088311</td>
112         <td>
113             <a href="#">贴吧</a>
```



```

114         <a href="#">图片</a>
115         <a href="#">百科</a>
116     </td>
117 </tr>
118 <tr bgcolor="white" align="center">
119     <td>1</td>
120     <td align="left">暴走大事件</td>
121     <td>
122         
123     </td>
124     <td>623557</td>
125     <td>4088311</td>
126     <td>
127         <a href="#">贴吧</a>
128         <a href="#">图片</a>
129         <a href="#">百科</a>
130     </td>
131 </tr>
132 </table>
133 </body>
134 </html>

```

表格的结构

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>25-表格的结构</title>
6  </head>
7  <body>
8      <!--
9      1. 由于表格中存储的数据比较复杂，为了方便管理和阅读以及提升语义，我们可以对表格中存储的数据
10     进行分类
11     表格中存储的数据可以分为4类
12     1.1 表格的标题
13     1.2. 表格的表头信息
14     1.3. 表格的主体信息
15     1.4. 表格的页尾信息
16
17     2. 表格的完整结构
18     <table>
19         <caption>表格的标题</caption>
20         <thead>
21             <tr>
22                 <th>每一列的标题</th>
23             </tr>
24         </thead>
25         <tbody>
26             <tr>
27                 <td>数据</td>
28             </tr>
29         </tbody>
30         <tfoot>
31             <tr>

```

```

31         <td>数据</td>
32     </tr>
33 </tfoot>
34 </table>
35
36 caption作用：指定表格的标题
37 thead作用：指定表格的表头信息
38 tbody作用：指定表格的主体信息
39 tfoot作用：指定表格的附加信息
40
41 3.注意点：
42 3.1.如果我们没有编写tbody，系统会系统给我们添加tbody
43 3.2.如果指定了thead和tfoot，那么在修改整个表格的高度时，thead和tfoot有自己默认的高度，
    不会随着表格的高度变化而变化
44 -->
45
46 <table border="1px" width="500px" height="500px">
47     <caption>学生信息</caption>
48     <thead>
49         <tr>
50             <th>姓名</th>
51             <th>年龄</th>
52         </tr>
53     </thead>
54     <tbody>
55         <tr>
56             <td>张三</td>
57             <td>20</td>
58         </tr>
59         <tr>
60             <td>李四</td>
61             <td>40</td>
62         </tr>
63     </tbody>
64     <tfoot>
65         <tr>
66             <td>2</td>
67             <td>30</td>
68         </tr>
69     </tfoot>
70 </table>
71 </body>
72 </html>

```

单元格合并

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>26-单元格合并</title>
6 </head>
7 <body>
8 <!--
9 1.水平方向上的单元格合并

```

```

10 可以给td标签添加一个colspan属性，来指定把某一个单元格当做多个单元格来看待(水平方向)
11 例如：
12 <td colspan="2"></td>
13 含义：把当前单元格当做两个单元格来看待
14
15 注意点：
16 1.由于把某一个单元格当做了多个单元格来看，所以就会多出一些单元格，所以需要删掉一些单元格
   才能正常显示
17 2.一定要记住单元格合并永远都是向后或者向下合并，而不能向前或者向上合并
18
19 2.垂直方向上的单元格合并
20 可以给td标签设置一个rowspan属性，来指定把某一个单元格当做多个单元格来看待(垂直方向)
21 例如：
22 <td rowspan="2"></td>
23 含义：把当前单元格当做两个单元格来看待
24 -->
25 <table bgcolor="black" width="500px" height="300px" align="center">
26     <tr bgcolor="white">
27         <td rowspan="2"></td>
28         <td></td>
29         <td></td>
30     </tr>
31     <tr bgcolor="white">
32         <!--<td></td>-->
33         <td></td>
34         <td></td>
35     </tr>
36     <tr bgcolor="white">
37         <td colspan="2"></td>
38         <td></td>
39     </tr>
40 </table>
41
42 <!--<table bgcolor="black" width="500px" height="300px" align="center">-->
43     <!--<tr bgcolor="white">-->
44         <!--<td colspan="2"></td>-->
45         <!--&lt;td></td>&dash;&gt;-->
46         <!--<td></td>-->
47     <!--</tr>-->
48     <!--<tr bgcolor="white">-->
49         <!--<td></td>-->
50         <!--<td></td>-->
51         <!--<td></td>-->
52     <!--</tr>-->
53     <!--<tr bgcolor="white">-->
54         <!--<td></td>-->
55         <!--<td colspan="2"></td>-->
56         <!--&lt;td></td>&dash;&gt;-->
57     <!--</tr>-->
58 <!--</table>-->
59 </body>
60 </html>

```

单元格合并练习

```
1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8">
6     <title>27-单元格合并练习</title>
7 </head>
8 <body>
9 <!--<table bgcolor="black" cellspacing="1px" width="500px" height="300px"
10 align="center">-->
11     <!--<tr bgcolor="white">-->
12         <!--<td></td>-->
13         <!--<td></td>-->
14         <!--<td></td>-->
15     <!--</tr>-->
16     <!--<tr bgcolor="white">-->
17         <!--<td colspan="3"></td>-->
18         <!--&lt; !&ndash;<td></td>&ndash;&gt;-->
19         <!--&lt; !&ndash;<td></td>&ndash;&gt;-->
20     <!--</tr>-->
21     <!--<tr bgcolor="white">-->
22         <!--<td></td>-->
23         <!--<td></td>-->
24         <!--<td></td>-->
25     <!--</tr>-->
26     <!--<tr bgcolor="white">-->
27         <!--<td></td>-->
28         <!--<td></td>-->
29         <!--<td></td>-->
30 <!--</table>-->
31
32 <!--<table bgcolor="black" cellspacing="1px" width="500px" height="300px"
33 align="center">-->
34     <!--<tr bgcolor="white">-->
35         <!--<td></td>-->
36         <!--<td rowspan="4"></td>-->
37         <!--<td></td>-->
38     <!--</tr>-->
39     <!--<tr bgcolor="white">-->
40         <!--<td></td>-->
41         <!--&lt; !&ndash;<td></td>&ndash;&gt;-->
42         <!--<td></td>-->
43     <!--</tr>-->
44     <!--<tr bgcolor="white">-->
45         <!--<td></td>-->
46         <!--&lt; !&ndash;<td></td>&ndash;&gt;-->
47         <!--<td></td>-->
48     <!--</tr>-->
49     <!--<tr bgcolor="white">-->
50         <!--<td></td>-->
51         <!--&lt; !&ndash;<td></td>&ndash;&gt;-->
52         <!--<td></td>-->
53 <!--</tr>-->
54 <!--</table>-->
55
```

```

56 <!--<table bgcolor="black" cellspacing="1px" width="500px" height="300px"
align="center">-->
57     <!--<tr bgcolor="white">-->
58         <!--<td>/td>-->
59         <!--<td>/td>-->
60         <!--<td>/td>-->
61     <!--</tr>-->
62     <!--<tr bgcolor="white">-->
63         <!--<td colspan="2">/td>-->
64         <!--&lt;td>/td>&ndash;&gt;-->
65         <!--<td>/td>-->
66     <!--</tr>-->
67     <!--<tr bgcolor="white">-->
68         <!--<td>/td>-->
69         <!--<td rowspan="2">/td>-->
70         <!--<td>/td>-->
71     <!--</tr>-->
72     <!--<tr bgcolor="white">-->
73         <!--<td>/td>-->
74         <!--&lt;td>/td>&ndash;&gt;-->
75         <!--<td>/td>-->
76     <!--</tr>-->
77 <!--</table>-->
78
79 <!--
80 快速移动选中的代码，上下移动
81 往上移动 Ctrl + shift + ↑(方向键上)
82 往下移动 Ctrl + shift + ↓(方向键下)
83
84 快速合并和展开代码(合并和展开的是某一个标签)
85 合并：Ctrl + -
86 展开：Ctrl + +
87
88 快速合并和展开代码(合并和展开选中的所有标签)
89 合并：Ctrl + shift + -
90 展开：Ctrl + shift + +
91
92 快速新启一行
93 shift + enter
94 -->
95 <table bgcolor="black" cellspacing="1px" width="500px" height="300px"
align="center">
96     <caption>单元格合并</caption>
97     <tr bgcolor="white">
98         <td>/td>
99         <td>/td>
100        <td>/td>
101    </tr>
102    <tr bgcolor="white">
103        <td>/td>
104        <td colspan="2" rowspan="2">/td>
105        <!--<td>/td>-->
106    </tr>
107    <tr bgcolor="white">
108        <td>/td>
109        <!--<td colspan="2">/td>-->
110        <!--<td>/td>-->
111    </tr>

```

```
112     <tr bgcolor="white">
113         <td></td>
114         <td></td>
115         <td></td>
116     </tr>
117 </table>
118 </body>
119 </html>
```

表单标签

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>28-表单标签</title>
6  </head>
7  <body>
8      <!--
9      1. 什么是表单？
10     表单就是专门用来收集用户信息的
11
12     2. 什么是表单元素？
13     2.1 什么是元素？
14     在HTML中 标签/标记/元素都是指HTML中的标签
15     例如： <a> a标签/a标记/a元素
16
17     表单元素其实还是HTML中的一些标签，只不过这些标签比较特殊，在浏览器中所有的表单标签都有特殊
18     的外观和默认的功能
19
20     3. 表单的格式：
21     <form>
22         <表单元素>
23     </form>
24
25     4. 常见的表单元素
26     input标签，input标签有一个type属性，这个属性有很多类型的取值，取值的不同就决定了input
27     标签的功能和外观不同
28
29     5. 注意：
30     表单元素一定要写在表单中
31     -->
32     <form>
33         <!-- 明文输入框 -->
34         账号:<input type="text"><br>
35         <!-- 暗文输入框 -->
36         密码:<input type="password"><br>
37         <!-- 给输入框设置默认值 -->
38         账号:<input type="text" value="lnj"><br>
39         密码:<input type="password" value="123"><br>
40
41         <!--
42         单选框
43         注意点:
```

```

42      1.默认情况下单选框不会互斥，要想单选框互斥那么必须给每一个单选框标签都设置一个name属性，然后name属性还必须设置相同的值
43      2.要想让单选框默认选中某一个框子，那么可以给input标签添加一个checked属性
44      3.在HTML中如果属性的取值和属性的名称一样，可以只写一个。但是在XHTML中必须写上取值，所以在企业开发中我们推荐大家不要省略取值
45      -->
46      性别：
47      <input type="radio" name="xx" checked>男
48      <input type="radio" name="xx">女
49      <input type="radio" name="xx" >保密<br>
50
51      <!--多选框-->
52      爱好：
53      <input type="checkbox">篮球
54      <input type="checkbox">足球
55      <input type="checkbox" checked="checked">棒球
56      <input type="checkbox" checked="checked">足浴
57 </form>
58 </body>
59 </html>

```

表单标签2

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>29-表单标签2</title>
6  </head>
7  <body>
8  <form action="http://www.baidu.com">
9      <!--明文输入框-->
10     账号:<input type="text" name="aa"><br>
11     <!--暗文输入框-->
12     密码:<input type="password" name="bb"><br>
13     <!--
14     定义一个普通按钮
15     作用:配合JS完成一些操作
16     -->
17     <input type="button" value="我是按钮" onclick="alert('lnj')">
18
19     <!--
20     定义一个图片按钮
21     作用:配合JS完成一些操作
22     -->
23     <input type="image" src="images/register.jpg" onclick="alert('lnj')">
24
25     <!--
26     定义重置按钮
27     作用:清空表单中的数据
28     注意点:
29     重置按钮有默认的按钮标题，默认叫做 重置
30     也可以通过value属性来修改默认标题
31     -->
32     <input type="reset" value="清空">

```

```

33
34     <!--
35     定义提交按钮
36     作用:将表单中的数据提交到远程服务器
37     注意点:
38     要想把表单中的数据提交到远程服务器,必须满足两个条件
39     1.告诉表单需要提交到哪个服务器
40     可以通过form标签的action属性来告诉表单,需要提交到那个服务器
41     2.告诉表单,表单中的哪些数据需要提交
42     -->
43     <input type="submit">
44
45     <!--
46     隐藏域
47     作用: 用于悄悄咪咪的收集用户的一些数据, 隐藏域是不会出现在界面中的
48     -->
49     <input type="hidden" name="cc" value="it666">
50 </form>
51 </body>
52 </html>

```

label标签

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>30-Label标签</title>
6  </head>
7  <body>
8  <!--
9  1.默认情况下文字和输入框是没有关联关系的, 也就是说点击文字输入框不会聚焦, 如果想点击文字时
   让对应的输入框聚焦, 那么就需要让文字和输入框进行绑定
10
11  2.要想让输入框和文字绑定在一起, 那么我们可以使用Label标签
12
13  3.绑定的格式:
14  3.1将文字利用label标签包裹起来
15  3.2给输入框添加一个id属性
16  3.3在label标签中通过for属性和输入框中的id进行绑定即可
17     <label for="account">账号:</label><input type="text" id="account">
18     -->
19     <form action="">
20         <label for="account">账号:</label><input type="text" id="account"><br>
21         <label for="pwd">密码:</label><input type="password" id="pwd"><br>
22
23         <hr>
24         <label>
25             账号:<input type="text">
26         </label><br>
27         <hr>
28
29         <label for="def">账号:</label><input type="text" id="abc"><br>
30         <label for="abc">密码:</label><input type="password" id="def"><br>
31     </form>

```



```
32 </body>
33 </html>
```

datalist标签

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>31-Datalist标签</title>
6 </head>
7 <body>
8 <!--
9 1.datalist标签
10 作用：给输入框绑定待选项
11
12 2.datalist格式：
13 <datalist>
14   <option>待选项内容</option>
15 </datalist>
16
17 3.如何给输入框绑定待选列表
18 1.搞一个输入框
19 2.搞一个datalist列表
20 3.给datalist列表标签添加一个id
21 4.给输入框添加一个list属性,将datalist的id对应的值赋值给list属性即可
22 -->
23
24 请输入你的车型：<input type="text" list="cars">
25
26 <datalist id="cars">
27   <option>奔驰</option>
28   <option>宝马</option>
29   <option>奥迪</option>
30   <option>路虎</option>
31   <option>宾利</option>
32 </datalist>
33
34 </body>
35 </html>
```

表单标签H5

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>32-表单标签-H5</title>
6 </head>
7 <body>
8 <form>
9   <!--
```

```
10  可以自动校验输入的内容是否符合邮箱的格式
11  -->
12  邮箱:<input type="email"><br>
13  <!--
14  可以自动校验输入的内容是否是URL地址
15  -->
16  域名:<input type="url"><br>
17  <!--
18  输入框中只能输入数字
19  -->
20  电话:<input type="number"><br>
21  <!--
22  可以通过日历来选择时间
23  -->
24  时间:<input type="date"><br>
25
26  <!--
27  可以通过取色板来选择颜色
28  -->
29  颜色: <input type="color"><br>
30
31  <input type="submit">
32 </form>
33 </body>
34 </html>
```

select和option标签

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>33-表单标签3</title>
6      <style type="text/css">
7          textarea{
8              resize: none;
9          }
10     </style>
11 </head>
12 <body>
13 <!--
14 1.select标签
15 作用：用于定义下拉列表
16
17 格式：
18 <select>
19     <optgroup label="分组名称">
20         <option>列表数据</option>
21     </optgroup>
22 </select>
23
24 注意点：
25 1. 下拉列表不能输入内容，但是可以直接在列表中选择内容
26 2. 可以通过给option标签添加一个selected属性来指定列表的默认值
27 3. 可以通过给option标签包裹一层optgroup标签来给下拉列表中的数据分类
```

```

28
29 2. textarea标签
30 作用：定义一个多行输入框
31
32 格式：
33 <textarea>
34 </textarea>
35
36 注意点：
37 1. 默认情况下输入框可以无限换行
38 2. 默认情况下输入框有自己的宽度和高度
39 3. 可以通过cols和rows来指定输入框的宽度和高度，但是虽然指定了列数和行数，但是还是可以无限
    往下输入
40 4. 默认情况下输入框是可以手动拉伸的
41 -->
42
43 <!--<select>-->
44     <!--<option>北京</option>-->
45     <!--<option>上海</option>-->
46     <!--<option>广州</option>-->
47     <!--<option>广西</option>-->
48     <!--<option selected="selected">武汉</option>-->
49 <!--</select>-->
50
51 <select>
52     <optgroup label="北京">
53         <option>朝阳区</option>
54         <option>昌平区</option>
55         <option>通州区</option>
56     </optgroup>
57     <optgroup label="广州">
58         <option>天河区</option>
59         <option>越秀区</option>
60         <option>黄浦区</option>
61     </optgroup>
62 </select>
63
64 <hr>
65
66 <textarea cols="20" rows="5">
67 </textarea>
68 </body>
69 </html>

```

表单练习

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>10-表单练习</title>
6 </head>
7 <body>
8 <form action="http://www.520it.com">
9     <fieldset>

```


[illegible]

```
70 &desc=just+do+it
71 &birthday=2018-03-26
72 &email=123%40qq.com
73 &phone=01234567890
74     -->
75     </fieldset>
76 </form>
77 </body>
78 </html>
```

video标签

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>35-video标签</title>
6 </head>
7 <body>
8 <!--
9 1.什么是video标签?
10 作用：播放视频
11
12 格式：
13 <video src="">
14 </video>
15
16 video标签的属性
17 src：用于告诉video标签需要播放的视频地址
18 autoplay：用于告诉video标签是否需要自动播放视频
19 controls：用于告诉video标签是否需要显示控制条
20 poster：用于告诉video标签视频没有播放之前显示的占位图片
21 loop：一般用于做广告视频，用于告诉video标签视频播放完毕之后是否需要循环播放
22 preload：预加载视频，但是需要注意preload和autoplay相冲，如果设置了autoplay属性，那么
    preload属性就会失效
23 muted：静音
24 width/height：和img标签中的一模一样
25 -->
26
27 <!--<video src="images/sb1.webm" autoplay="autoplay" controls="controls">
    </video>-->
28 <!--<video src="images/sb1.webm" controls="controls"
    poster="images/NJ.jpg"></video>-->
29 <video src="images/sb1.webm" autoplay="autoplay" loop="loop" muted="muted"
    width="800px"></video>
30 </body>
31 </html>
```

video标签的第二种用法

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

```

3 <head>
4   <meta charset="UTF-8">
5   <title>36-video标签的第二种格式</title>
6 </head>
7 <body>
8 <!--
9 1.格式:
10 <video>
11   <source src="" type=""></source>
12   <source src="" type=""></source>
13 </video>
14
15 2.第二种格式存在的意义:
16 由于视频数据非常非常的重要, 所以五大浏览器厂商都不愿意支持别人的视频格式, 所以导致了没有一
17 种视频格式是所有浏览器都支持的
18 这个时候W3C为了解决这个问题, 所以推出了第二个video标签的格式
19
20 video标签的第二种格式存在的意义就是为了解决浏览器适配问题
21 video 元素支持三种视频格式, 我们可以把这三种格式都通过source标签指定给video标签, 那么以
22 后当浏览器播放视频时它就会从这三种中选择一种自己支持的格式来播放
23
24 3.注意点:
25 3.1当前通过video标签的第二种格式虽然能够指定所有浏览器都支持的视频格式, 但是想让所有浏览
26 器都通过video标签播放视频还有一个前提条件, 就是浏览器必须支持HTML5标签, 否则同样无法播放
27 3.2在过去的一些浏览器是不支持HTML5标签的, 所以为了让过去的一些浏览器也能够通过video标签
28 来播放视频, 那么我们以后可以通过一个JS的框架叫做html5media来实现
29
30 -->
31 <video autoplay="autoplay" controls="controls">
32   <source src="images/sb1.webm" type="video/webm"></source>
33   <source src="images/sb1.mp4" type="video/mp4"></source>
34   <source src="images/sb1.ogg" type="video/ogg"></source>
35 </video>
36 </body>
37 </html>

```

audio标签

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>37-audio标签</title>
6 </head>
7 <body>
8 <!--
9 1.什么是audio标签?
10 作用: 播放音频
11
12 格式:
13 <audio src="">
14 </audio>
15
16 <audio>
17   <source src="" type="">
18 </audio>

```

```

19
20  注意：
21  audio标签的使用和video标签的使用基本一样，video中能够使用的属性在audio标签中大部分都能
    够使用，并且功能都一样
22  只不过有3个属性不能用，height/width/poster
23  -->
24
25  <!--<audio src="images/yinyue.mp3" autoplay="autoplay" controls="controls">
    </audio>-->
26
27  <audio autoplay="autoplay" controls="controls">
28      <source src="images/yinyue.mp3" type="audio/mp3">
29  </audio>
30  </body>
31  </html>

```

详情和概要标签

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>38-详情和概要标签</title>
6  </head>
7  <body>
8  <!--
9  1. 什么是详情和概要标签？
10  作用：利用summary标签来描述概要信息，利用details标签来描述详情信息
11  默认情况下是折叠展示，想看见详情必须点击
12
13  格式：
14  <details>
15      <summary>概要信息</summary>
16      详情信息
17  </details>
18  -->
19
20  <details>
21      <summary>郑伊健</summary>
22  简介：郑伊健，1967年10月4日出生于中国香港，籍贯广东恩平，香港影视演员、流行男歌手。1988年
    参加新秀歌唱大赛加入无线电视，因拍摄“阳光柠檬茶”广告而入行，拜罗文为师。1991年...
23  </details>
24  </body>
25  </html>

```

marquee标签

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>39-marquee标签</title>

```



```

6      <style>
7          marquee {
8              width:200px;
9              height:200px;
10             background-color: skyblue;
11         }
12     </style>
13 </head>
14 <body>
15 <!--
16 1.什么是marquee标签?
17 作用: 跑马灯
18
19 格式:
20 <marquee>内容</marquee>
21
22 属性:
23 direction: 设置滚动方向 left/right/up/down
24 scrollamount: 设置滚动速度, 值越大就越快
25 loop: 设置滚动次数, 默认是-1, 也就是无限滚动
26 behavior: 设置滚动类型 slide滚动到边界就停止, alternate滚动到边界就弹回
27
28 注意点:
29 marquee标签不是W3C推荐的标签, 在W3C官方文档中也无法查询这个标签, 但是各大浏览器对这个标
    签的支持非常好
30 -->
31
32 <!--滚动方向-->
33 <marquee>随便写点内容</marquee>
34 <marquee direction="right">随便写点内容</marquee>
35 <marquee direction="up">随便写点内容</marquee>
36 <marquee direction="down">随便写点内容</marquee>
37 <hr>
38 <!--设置滚动速度-->
39 <marquee scrollamount="1">随便写点内容</marquee>
40 <marquee scrollamount="100">随便写点内容</marquee>
41 <hr>
42 <!--设置滚动次数-->
43 <marquee loop="1">随便写点内容</marquee>
44 <hr>
45 <!--设置滚动类型-->
46 <marquee behavior="slide">随便写点内容</marquee>
47 <marquee behavior="alternate">随便写点内容</marquee>
48
49 <marquee>
50     
51 </marquee>
52 </body>
53 </html>

```

html中被废弃的标签

```

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

```

```

4      <meta charset="UTF-8">
5      <title>40-HTML中被废弃的标签</title>
6      <style type="text/css">
7          .one {
8              font-weight: bold;
9          }
10         .two {
11             text-decoration: underline;
12         }
13         .three {
14             font-style: italic;
15         }
16         .four {
17             text-decoration: line-through;
18         }
19     </style>
20 </head>
21 <body>
22 <!--
23 1. 为什么HTML中有一部分标签会被废弃?
24 因为当前的HTML中的标签只有一个作用，就是用来添加语义
25 而早期的HTML标签中有一部分标签是没有语义的，
26 有一部分标签是用来修改样式的
27 所以这部分标签就被淘汰了
28
29 <br> <hr> <font>
30 <b> <u> <i> <s>以上标签都是没有语义的，都是用来修改样式的
31 b(bold) 加粗文本，没有任何语义的
32 u(underline) 给文本天津下划线，没有任何语义的
33 i(italic) 将文本倾斜，没有任何语义的
34 s(strikethourgh) 给文本添加删除线，没有任何语义的
35
36 注意点：
37 以后在企业开发中，不到万不得已一定不要使用这些被废弃掉的标签
38 如果一定要使用，一般情况下都是用来作为CSS的钩子来使用
39
40 strong == b
41 ins == u
42 em == i
43 del == s
44
45 strong语义：定义重要性强调的文字
46 ins语义(inseted)：定义插入的文字
47 em语义(emphasized)：定义强调的文字
48 del语义(deleted)：定义被删除的文字
49 -->
50
51 <b>我是文字</b>
52 <u>我是文字</u>
53 <i>我是文字</i>
54 <s>我是文字</s>
55 <hr>
56 <strong>我是文字</strong>
57 <ins>我是文字</ins>
58 <em>我是文字</em>
59 <del>我是文字</del>
60 <hr>
61 <p class="one">我是文字</p>

```

```
62 <p class="two">我是文字</p>
63 <p class="three">我是文字</p>
64 <p class="four">我是文字</p>
65 </body>
66 </html>
```

字符实体

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>41-字符实体</title>
6 </head>
7 <body>
8 <!--
9 1. 在HTML中对空格/回车/tab不敏感，会把多个空格/回车/tab当做一个空格来处理
10
11
12 2. 什么是字符实体？
13 在HTML中有的字符是被HTML保留的，有的HTML字符在HTML中是有特殊含义的，是不能在浏览器中直接显示出来的，那么这些东西要想显示出来就必须通过字符实体
14
15 &nbsp; 空格，一个&nbsp;就是一个空格，有多少个&nbsp;就有多少个空格
16 &lt; 小于符号 <
17 (less than)
18 &gt; 大于符号 >
19 (greater than)
20 &copy; 版权符号
21 -->
22
23 <p>我&nbsp;&nbsp;&nbsp;爱你</p>
24
25 <p>到此为止我们的HTML的基础标签就学习完毕了，例如我们学习了&lt;h1&gt;标签，
26 &lt;table&gt;标签，&lt;img&gt;标签....</p>
27 &copy;
28 </body>
29 </html>
```

CSS

css格式

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>02-CSS的格式</title>
6     <style type="text/css">
7         p{
```

```

8         text-align: center;
9         color: red;
10    }
11    </style>
12 </head>
13 <body>
14 <!--
15 1.格式:
16 <style type="text/css">
17     标签名称{
18         属性名称: 属性对应的值;
19         ...
20     }
21 </style>
22
23 2.注意点:
24 1.style标签必须写在head标签的开始标签和结束标签之间(也就是必须和title标签是兄弟关系)
25 2.style标签中的type属性其实可以不用写, 默认就是type="text/css"
26 3.设置样式时必须按照固定的格式来设置. key: value;
27 其中:不能省略, 分号大多数情况下也不能省略
28
29 4.CSS怎么学?
30 CSS的学习一共分为两大部分, 一个是CSS的属性, 另一个是CSS选择器. 也就是说着两部分学完CSS就没有别的东西了
31 -->
32 <h1>我是标题</h1>
33 <p>我是段落</p>
34 <p>我是段落</p>
35 <p>我是段落</p>
36 </body>
37 </html>

```

字体属性

```

1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8">
6     <title>03-文字属性</title>
7     <style>
8         p{
9             font-style: italic;
10            font-weight: bold;
11            font-size: 10px;
12            font-family:"楷体";
13        }
14    </style>
15 </head>
16 <body>
17 <!--
18 1.规定文字样式的属性
19 格式: font-style: italic;
20 取值:
21 normal : 正常的, 默认就是正常的

```

```

22 italic :  倾斜的
23 快捷键:
24 fs font-style: italic;
25 fsn font-style: normal;
26
27 2.规定文字粗细的属性
28 格式: font-weight: bold;
29 单词取值:
30 bold 加粗
31 bolder 比加粗还要粗
32 lighter 细线, 默认就是细线
33 数字取值:
34 100-900之间整百的数字
35
36 快捷键
37 fw font-weight:;
38 fwb font-weight: bold;
39 fwbr font-weight: bolder;
40
41 3.规定文字大小的属性
42 格式: font-size: 30px;
43 单位: px (像素 pixel)
44 注意: 通过font-size设置大小一定要带单位, 也就是一定要写px
45 快捷键
46 fz font-size:;
47 fz30 font-size: 30px;
48
49 4.规定文字字体的属性
50 格式: font-family:"楷体";
51 注意点:
52 1.如果取值是中文, 需要用双引号或者单引号括起来
53 2.设置的字体必须是用户电脑里面已经安装的字体
54 快捷键
55 ff font-family:;
56
57 -->
58 <i>我是文字</i>
59 <b>我是文字</b>
60 <p>abc我是段落</p>
61 </body>
62 </html>

```

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>04-字体属性补充</title>
6      <style>
7          p{
8              /*
9              被注释掉的内容
10             */
11             /*font-family:"乱七八糟的字体", "微软雅黑";*/
12             /*font-family: Arial, "微软雅黑";*/
13             font-family:"Microsoft YaHei";
14         }
15     </style>

```

```

16 </head>
17 <body>
18 <!--
19 1.如果设置的字体不存在，那么系统会使用默认的字体来显示
20 默认使用宋体
21
22 2.如果设置的字体不存在，而我们又不想用默认的字体来显示怎么办？
23 可以给字体设置备选方案
24 格式:font-family:"字体1", "备选方案1", ...;
25
26 3.如果想给中文和英文分别单独设置字体，怎么办？
27 但凡是中文字体，里面都包含了英文
28 但凡是英文字体，里面都没有包含中文
29 也就是说中文字体可以处理英文，而英文字体不能处理中文
30 **注意点：如果想给界面中的英文单独设置字体，那么英文的字体必须写在中文的前面
31
32 4.补充在企业开发中最常见的字体有以下几个
33 中文：宋体/黑体/微软雅黑
34 英文："Times New Roman"/Arial
35
36 还需要知道一点，就是并不是名称是英文就一定是英文字体
37 因为中文字体其实都有自己的英文名称，所以是不是中文字体主要看能不能处理中文
38 宋体 SimSun
39 黑体 SimHei
40 微软雅黑 Microsoft YaHei
41 -->
42 <p>abc我是段落</p>
43 </body>
44 </html>

```

文字属性的缩写

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>05-文字属性的缩写</title>
6   <style>
7     p{
8       /*
9         font-style: italic;
10        font-weight: bold;
11        font-size: 10px;
12        font-family:"楷体";
13       */
14       font:bold italic 10px "楷体";
15     }
16   </style>
17 </head>
18 <body>
19 <!--
20 缩写格式：
21 font: style weight size family;
22 例如：
23 font:italic bold 10px "楷体";

```

```
24
25 注意点：
26 1. 在这种缩写格式中有的属性值可以省略
27 style可以省略
28 weight可以省略
29 2. 在这种缩写格式中style和weight的位置可以交换
30 3. 在这种缩写格式中有的属性值是不可以省略的
31 size不能省略
32 family不能省略
33 4. size和family的位置是不能随便乱放的
34 size一定要写在family的前面，而且size和family必须写在所有属性的最后
35 -->
36 <p>abc我是段落</p>
37 </body>
38 </html>
```

文本属性

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>06-文本属性</title>
6      <style>
7          p{
8              text-decoration: none;
9              text-align: left;
10             text-indent: 2em;
11         }
12         a{
13             text-decoration: none;
14         }
15     </style>
16 </head>
17 <body>
18 <!--
19 1. 文本装饰的属性
20 格式:text-decoration: underline;
21 取值：
22 underline 下划线
23 line-through 删除线
24 overline 上划线
25 none 什么都没有，最常见的用途就是用于去掉超链接的下划线
26 快捷键：
27 td text-decoration: none;
28 tdu text-decoration: underline;
29 tdl text-decoration: line-through;
30 tdo text-decoration: overline;
31
32
33 2. 文本水平对齐的属性
34 格式: text-align: right;
35 取值：
36 left 左
37 right 右
```

[illegible]

颜色属性

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>07-颜色属性</title>
6      <style>
7          p{
8              /*color: red;*/
9              /*color: rgb(255,0,0);*/
10             /*color: rgba(255,0,0,1);*/
11             color: #FF0000;
12             color: #F00;
13             /*color: rgba(255,0,0,0.2);*/
14             color: #ffee00;
15             color: #fe0;
16             color: #769abb;
17         }
18     </style>
19 </head>
20 <body>
21 <!--
22 1. 在CSS中如何通过color属性来修改文字颜色
23 格式: color: 值;
24 取值:
25 1.1 英文单词

```


26 一般情况下常见的颜色都有对应的英文单词，但是英文单词能够表达的颜色是有限制的，也就是说不是所有的颜色都能够通过英文单词来表达

27

28 1.2rgb

29 **rgb**其实就是三原色，其中**r**(red 红色) **g**(green 绿色) **b**(blue 蓝色)

30 格式：**rgb(0,0,0)**

31 那么这个格式中的

32 第一个数字就是用来设置三原色的光源元件红色显示的亮度

33 第二个数字就是用来设置三原色的光源元件绿色显示的亮度

34 第三个数字就是用来设置三原色的光源元件蓝色显示的亮度

35 这其中的每一个数字它的取值是**0-255**之前，**0**代表不发光，**255**代表发光，值越大就越亮

36

37 红色：**rgb(255,0,0);**

38 绿色：**rgb(0,255,0);**

39 蓝色：**rgb(0,0,255);**

40 黑色：**rgb(0,0,0);**

41 白色：**rgb(255,255,255);**

42

43 在前端开发中其实并不常用黑色

44 只要让红色/绿色/蓝色的值都一样就是灰色

45 而且如果这三个值越小那么就越偏黑色，越大就越偏白色

46 例如：**color: rgb(200,200,200);**

47

48 1.3rgba

49 **rgba**中的**rgb**和前面讲解的一样，只不过多了一个**a**

50 那么这个**a**呢代表透明度，取值是**0-1**，取值越小就越透明

51 例如：**color: rgba(255,0,0,0.2);**

52

53 1.4十六进制

54 在前端开发中通过十六进制来表示颜色，其实本质就是**RGB**

55 十六进制中是通过每两位表示一个颜色

56 例如：**#FFEE00** **FF**表示**R** **EE**表示**G** **00**表示**B**

57

58 什么是十六进制？

59 十六进制和十进制一样都是一种计数的方式

60 在十进制中取值范围**0-9**，逢十进一

61 在十六进制中取值范围**0-F**，逢十六进一

62

63 9

64 a 10

65 b 11

66 c 12

67 d 13

68 e 14

69 f 15

70 10 16

71 11 17

72 12 18

73

74 十六进制和十进制转换的公式

75 用十六进制的第一位***16** + 十六进制的第二位 = 十进制

76 **15 == 1*16 + 5 = 21**

77 **12 == 1*16 + 2 = 18**

78 **FF == F*16 + F == 15*16 + 15 == 240 + 15 = 255**

79 **00 == 0*16 + 0 = 0**

80

81 1.5十六进制缩写

82 在**CSS**中只要十六进制的颜色每两位的值都是一样的，那么就可以简写为一位

```
83 例如：
84 #FFEE00 == #FE0
85
86 注意点：
87 1. 如果当前颜色对应的两位数字不一样， 那么就不能简写
88 #123456；
89
90 2. 如果两位相同的数字不是属于同一个颜色的， 也不能简写
91 #122334
92
93 -->
94 <p>我是段落</p>
95 </body>
96 </html>
```

标签选择器

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>08-标签选择器</title>
6      <style>
7          p{
8              color: red;
9          }
10         h1{
11             color: blue;
12         }
13     </style>
14 </head>
15 <body>
16 <!--
17 1. 什么是标签选择器？
18 作用： 根据指定的标签名称， 在当前界面中找到所有该名称的标签， 然后设置属性
19
20 格式：
21 标签名称{
22     属性:值;
23 }
24
25 注意点：
26 1. 标签选择器选中的是当前界面中所有的标签， 而不能单独选中某一个标签
27 2. 标签选择器无论标签藏得多深都能选中
28 3. 只要是HTML中的标签就可以作为标签选择器(h/a/img/ul/ol/dl/input....)
29 -->
30 <p>我是段落</p>
31 <p>我是段落</p>
32 <p>我是段落</p>
33 <p>我是段落</p>
34 <ul>
35     <li>
36         <ul>
37             <li>
38                 <ul>
```

```

39         </li>
40         <p>我是段落</p>
41     </li>
42 </ul>
43 </li>
44 </ul>
45 </li>
46 </ul>
47 <h1>我是标题</h1>
48 </body>
49 </html>

```

id选择器

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>09-id选择器</title>
6      <style>
7          #identity1{
8              color: red;
9          }
10         #identity2{
11             color: green;
12         }
13         #identity3{
14             color: blue;
15         }
16         #identity4{
17             color: yellow;
18         }
19     </style>
20 </head>
21 <body>
22 <!--
23 1.什么是id选择器?
24 作用: 根据指定的id名称找到对应的标签, 然后设置属性
25
26 格式:
27 #id名称{
28     属性:值;
29 }
30
31 注意点:
32 1.每个HTML标签都有一个属性叫做id, 也就是说每个标签都可以设置id
33 2.在同一个界面中id的名称是不可以重复的
34 3.在编写id选择器时一定要在id名称前面加上#
35 4.id的名称是有一定的规范的
36 4.1id的名称只能由字母/数字/下划线
37 a-z 0-9 _
38 4.2id名称不能以数字开头
39 4.3id名称不能是HTML标签的名称
40 不能是a h1 img input ...

```

```
41 5.在企业开发中一般情况下如果仅仅是为了设置样式，我们不会使用id，因为在前端开发中id是留给js使用的
42 -->
43 <p id="identity1">迟到毁一生</p>
44 <p id="identity2">早退穷三代</p>
45 <p id="identity3">按时上下班</p>
46 <p id="identity4">必成高富帅</p>
47 </body>
48 </html>
```

class选择器

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>10-类选择器</title>
6      <style>
7          .pp{
8              color: red;
9          }
10         .ppp{
11             color: green;
12         }
13         .pppp{
14             color: blue;
15         }
16         .ppppp{
17             color: yellow;
18         }
19         .para1{
20             font-size: 100px;
21         }
22         .para2{
23             font-style: italic;
24         }
25     </style>
26 </head>
27 <body>
28 <!--
29 1.什么是类选择器？
30 作用：根据指定的类名称找到对应的标签，然后设置属性
31
32 格式：
33 .类名{
34     属性:值;
35 }
36
37 注意点：
38 1.每个HTML标签都有一个属性叫做class，也就是说每个标签都可以设置类名
39 2.在同一个界面中class的名称是可以重复的
40 3.在编写class选择器时一定要在class名称前面加上.
41 4.类名的命名规范和id名称的命名规范一样
42 5.类名就是专门用来给CSS设置样式的
43 6.在HTML中每个标签可以同时绑定多个类名
```

```
44 格式：
45 <标签名称 class="类名1 类名2 ...">
46 错误的写法：
47 <p class="para1" class="para2">
48 -->
49 <p class="pp">迟到毁一生</p>
50 <p class="ppp">早退穷三代</p>
51 <p class="pppp">按时上下班</p>
52 <p class="ppppp">必成高富帅</p>
53
54 <p class="para1 para2">我是段落</p>
55 <p class="para1 para2">我是段落</p>
56
57 </body>
58 </html>
```

后代选择器

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>12-后代选择器</title>
6      <style>
7          /*
8          div p{
9              color: red;
10         }
11         */
12         /*
13         #identity p{
14             color: red;
15         }
16         */
17         /*
18         .para p{
19             color: blue;
20         }
21         */
22         /*
23         #identity #iii{
24             color: skyblue;
25         }
26         */
27         /*
28         #identity .ccc{
29             color: purple;
30         }
31         */
32         div ul li p{
33             color: red;
34         }
35     </style>
36 </head>
37 <body>
```

```

38 <!--
39 1. 什么是后代选择器？
40 作用：找到指定标签的所有特定的后代标签， 设置属性
41
42 格式：
43 标签名称1 标签名称2{
44     属性:值;
45 }
46 先找到所有名称叫做"标签名称1"的标签，然后再在这个标签下面去查找所有名称叫做"标签名称2"的标
   签，然后在设置属性
47 div p{}
48
49 注意点：
50 1. 后代选择器必须用空格隔开
51 2. 后代不仅仅是儿子，也包括孙子/重孙子，只要最终是放到指定标签中的都是后代
52 3. 后代选择器不仅仅可以使用标签名称，还可以使用其它选择器
53 4. 后代选择器可以通过空格一直延续下去
54 -->
55 <!--div ul li p-->
56 <p>我是段落</p>
57 <div id="identity" class="para">
58     <p>我是段落</p>
59     <p>我是段落</p>
60     <ul>
61         <li>
62             <!--<p id="iii" class="ccc">我是段落</p>-->
63             <p>我是段落</p>
64         </li>
65         <li>
66             <p>我是段落</p>
67         </li>
68     </ul>
69 </div>
70 <p>我是段落</p>
71 </body>
72 </html>

```

子元素选择器

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>13-子元素选择器</title>
6     <style>
7         /*
8         div>p{
9             color: red;
10        }
11        */
12
13        #identity>p{
14            color: blue;
15        }
16

```

```

17     /*
18     div>ul>li>p{
19         color: purple;
20     }
21     */
22 </style>
23 </head>
24 <body>
25 <!--
26 1. 什么是子元素选择器？
27 作用：找到指定标签中所有特定的直接子元素，然后设置属性
28
29 格式：
30 标签名称1>标签名称2{
31     属性:值;
32 }
33 先找到所有名称叫做"标签名称1"的标签，然后在这个标签中查找所有直接子元素名称叫做"标签名称
34 2"的元素
35
36 注意点：
37 1. 子元素选择器只会查找儿子，不会查找其他被嵌套的标签
38 2. 子元素选择器之间需要用>符号连接，并且不能有空格
39 3. 子元素选择器不仅仅可以使用标签名称，还可以使用其它选择器
40 4. 子元素选择器可以通过>符号一直延续下去
41 -->
42 <!--div>ul>li>p-->
43 <p>我是段落</p>
44 <div id="identity">
45     <p>我是段落</p>
46     <p>我是段落</p>
47     <ul>
48         <li><p>我是段落</p></li>
49     </ul>
50 </div>
51 <p>我是段落</p>
52 </body>
53 </html>

```

子代和后代选择器

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>14-后代选择器和子元素选择器</title>
6 </head>
7 <body>
8 <!--
9 1. 后代选择器和子元素选择器之间的区别？
10 1.1
11 后代选择器使用空格作为连接符号
12 子元素选择器使用>作为连接符号
13 1.2
14 后代选择器会选中指定标签中，所有的特定后代标签，也就是会选中儿子/孙子...，只要是被放到指
    定标签中的特定标签都会被选中

```

```
15 子元素选择器只会选中指定标签中，所有的特定的直接标签，也就是只会选中特定的儿子标签
16
17 2. 后代选择器和子元素选择器之间的共同点
18 2.1
19 后代选择器和子元素选择器都可以使用标签名称/id名称/class名称来作为选择器
20 2.2
21 后代选择器和子元素选择器都可以通过各自的连接符号一直延续下去
22 选择器1>选择器2>选择器3>选择器4{}
23
24 3. 在企业开发中如何选择
25 如果想选中指定标签中的所有特定的标签，那么就使用后代选择器
26 如果只想选中指定标签中的所有特定儿子标签，那么就使用子元素选择器
27
28 -->
29 </body>
30 </html>
```

交集选择器

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>15-交集选择器</title>
6      <style>
7          /*
8          p.para1{
9              color: red;
10         }
11         */
12         .para1#identity{
13             color: blue;
14         }
15     </style>
16 </head>
17 <body>
18 <!--
19 1. 什么是交集选择器？
20 作用：给所有选择器选中的标签中，相交的那部分标签设置属性
21
22 格式：
23 选择器1选择器2{
24     属性：值；
25 }
26
27 注意点：
28 1. 选择器和选择器之间没有任何的连接符号
29 2. 选择器可以使用标签名称/id名称/class名称
30 3. 交集选择器仅仅作为了解，企业开发中用的并不多
31 -->
32
33 <p>我是段落</p>
34 <p>我是段落</p>
35 <p class="para1" id="identity">我是段落</p>
36 <p class="para1">我是段落</p>
```



```
37 <p>我是段落</p>
38 </body>
39 </html>
```

并集选择器

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>16-并集选择器</title>
6   <style>
7     /*
8     .ht{
9       color: red;
10    }
11    .para{
12      color: red;
13    }
14    */
15    .ht,.para{
16      color: red;
17    }
18  </style>
19 </head>
20 <body>
21 <!--
22 1. 什么是并集选择器?
23 作用: 给所有选择器选中的标签设置属性
24
25 格式:
26 选择器1,选择器2{
27   属性:值;
28 }
29
30 注意点:
31 1. 并集选择器必须使用,来连接
32 2. 选择器可以使用标签名称/id名称/class名称
33 -->
34
35 <h1 class="ht">我是标题</h1>
36 <p class="para">我是段落</p>
37 <p>我是段落</p>
38 <p>我是段落</p>
39 </body>
40 </html>
```

兄弟选择器

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

```
4     <meta charset="UTF-8">
5     <title>17-兄弟选择器</title>
6     <style>
7         /*
8         h1+p{
9             color: red;
10        }
11        */
12        h1~p{
13            color: green;
14        }
15    </style>
16 </head>
17 <body>
18 <!--
19 1.相邻兄弟选择器 CSS2
20 作用：给指定选择器后面紧跟的那个选择器选中的标签设置属性
21
22 格式：
23 选择器1+选择器2{
24     属性:值;
25 }
26
27 注意点：
28 1.相邻兄弟选择器必须通过+连接
29 2.相邻兄弟选择器只能选中紧跟其后的那个标签，不能选中被隔开的标签
30
31 2.通用兄弟选择器 CSS3
32 作用：给指定选择器后面的所有选择器选中的所有标签设置属性
33
34 格式：
35 选择器1~选择器2{
36     属性:值;
37 }
38
39 注意点：
40 1.通用兄弟选择器必须用~连接
41 2.通用兄弟选择器选中的是指定选择器后面某个选择器选中的所有标签，无论有没有被隔开都可以选中
42 -->
43
44 <!--
45 <h1>我是标题</h1>
46 <a href="#">我是超链接</a>
47 <p>我是段落</p>
48 <p>我是段落</p>
49 <p>我是段落</p>
50 <h1>我是标题</h1>
51 <p>我是段落</p>
52 <p>我是段落</p>
53 <p>我是段落</p>
54 -->
55 <h1>我是标题</h1>
56 <a href="#">我是超链接</a>
57 <p>我是段落</p>
58 <p>我是段落</p>
59 <a href="#">我是超链接</a>
60 <p>我是段落</p>
61
```

```
62 <h1>我是标题</h1>
63 <p>我是段落</p>
64 <p>我是段落</p>
65 <p>我是段落</p>
66 </body>
67 </html>
```

序选择器

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>18-序选择器</title>
6   <style>
7     /*
8     p:first-child{
9       color: red;
10    }
11    */
12    /*
13    p:first-of-type{
14      color: blue;
15    }
16    */
17    /*
18    p:last-child{
19      color: red;
20    }
21    */
22    /*
23    p:last-of-type{
24      color: blue;
25    }
26    */
27    /*
28    p:nth-child(3){
29      color: red;
30    }
31    */
32    /*
33    p:nth-of-type(3){
34      color: blue;
35    }
36    */
37    /*
38    p:nth-last-child(2){
39      color: red;
40    }
41    */
42    /*
43    p:nth-last-of-type(2){
44      color: red;
45    }
46    */
```

```

47      /*
48      p:only-child{
49          color: purple;
50      }
51      */
52      /*
53      p:only-of-type {
54          color: red;
55      }
56      */
57      .para:only-of-type {
58          color: red;
59      }
60  </style>
61 </head>
62 <body>
63 <!--
64 CSS3中新增的选择器最具代表性的就是序选择器
65 1.同级别的第几个
66 :first-child 选中同级别中的第一个标签
67 :last-child 选中同级别中的最后一个标签
68 :nth-child(n) 选中同级别中的第n个标签
69 :nth-last-child(n) 选中同级别中的倒数第n个标签
70 :only-child 选中父元素中唯一的标签
71 注意点：不区分类型
72
73 2.同类型的第几个
74 :first-of-type 选中同级别中同类型的第一个标签
75 :last-of-type 选中同级别中同类型的最后一个标签
76 :nth-of-type(n) 选中同级别中同类型的第n个标签
77 :nth-last-of-type(n) 选中同级别中同类型的倒数第n个标签
78 :only-of-type 选中父元素中唯一类型的某个标签
79 -->
80 <!--
81 <h1>我是标题</h1>
82 <p>我是段落1</p>
83 <p>我是段落2</p>
84 <p>我是段落3</p>
85 <p>我是段落4</p>
86 <div>
87     <p>我是段落5</p>
88     <p>我是段落6</p>
89     <p>我是段落7</p>
90     <p>我是段落8</p>
91 </div>
92 -->
93 <p class="para">我是段落1</p>
94 <div>
95     <p class="para">我是段落2</p>
96     <p class="para">我是段落2</p>
97     <h1>我是标题</h1>
98 </div>
99 </body>
100 </html>

```

```

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

```

```

3 <head>
4   <meta charset="UTF-8">
5   <title>19-序选择器下</title>
6   <style>
7     /*
8     p:nth-child(odd){
9       color: red;
10    }
11    p:nth-child(even){
12      color: blue;
13    }
14    */
15    /*
16    p:nth-of-type(odd){
17      color: red;
18    }
19    p:nth-of-type(even){
20      color: blue;
21    }
22    */
23    /*
24    p:nth-child(2n+0){
25      color: red;
26    }
27    p:nth-child(2n+1){
28      color: blue;
29    }
30    */
31    p:nth-child(3n+0){
32      color: red;
33    }
34  </style>
35 </head>
36 <body>
37 <!--
38 :nth-child(odd) 选中同级别中的所有奇数
39 :nth-child(even) 选中同级别中的所有偶数
40
41 :nth-child(xn+y)
42 x和y是用户自定义的，而n是一个计数器，从0开始递增
43
44 -->
45 <p>我是项目</p>
46 <p>我是项目</p>
47 <p>我是项目</p>
48 <p>我是项目</p>
49 <p>我是项目</p>
50 <p>我是项目</p>
51 <p>我是项目</p>
52 </body>
53 </html>

```

属性选择器

```

2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>20-属性选择器上</title>
6      <style>
7          /*
8          p[id]{
9              color: red;
10         }
11         */
12         p[class=cc]{
13             color: blue;
14         }
15     </style>
16 </head>
17 <body>
18 <!--
19 1. 什么是属性选择器?
20 作用: 根据指定的属性名称找到对应的标签, 然后设置属性
21
22 格式:
23 [attribute]
24 作用: 根据指定的属性名称找到对应的标签, 然后设置属性
25
26 [attribute=value]
27 作用: 找到有指定属性, 并且属性的取值等于value的标签, 然后设置属性
28 最常见的应用场景, 就是用于区分input属性
29 input[type=password]{}
30 <input type="text" name="" id="">
31 <input type="password" name="" id="">
32 -->
33 <p id="identity1">我是段落1</p>
34 <p id="identity2" class="cc">我是段落2</p>
35 <p class="cc">我是段落3</p>
36 <p id="identity3" class="para">我是段落4</p>
37 <p>我是段落5</p>
38
39 </body>
40 </html>

```

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>21-属性选择器下</title>
6      <style>
7          /*
8          img[alt^=abc]{
9              color: red;
10         }
11         */
12         /*
13         img[alt|=abc]{
14             color: red;
15         }
16         img[alt$=abc]{
17             color: blue;

```

```

18     }
19     */
20     /*
21     img[alt*=abc]{
22         color: red;
23     }
24     */
25     img[alt~=abc]{
26         color: red;
27     }
28 </style>
29 </head>
30 <body>
31 <!--
32 1.属性的取值是以什么开头的
33 [attribute|=value] CSS2
34 [attribute^=value] CSS3
35 两者之间的区别：
36 CSS2中的只能找到value开头,并且value是被-和其它内容隔开的
37 CSS3中的只要是以value开头的都可以找到，无论有没有被-隔开
38
39 2.属性的取值是以什么结尾的
40 [attribute$=value] CSS3
41
42 3.属性的取值是否包含某个特定的值得
43 [attribute~=value] CSS2
44 [attribute*=value] CSS3
45 两者之间的区别：
46 CSS2中的只能找到独立的单词，也就是包含value,并且value是被空格隔开的
47 CSS3中的只要包含value就可以找到
48 -->
49
50 <!--
51 <img src="" alt="abcdef">
52 <img src="" alt="abc-www">
53 <img src="" alt="abc ppp">
54 <img src="" alt="defabc">
55 <img src="" alt="ppp abc">
56 <img src="" alt="www-abc">
57 <img src="" alt="qq">
58 <img src="" alt="yy">
59 -->
60
61 <img src="" alt="abcwwwmmm">
62 <img src="" alt="wwwmmmmabc">
63 <img src="" alt="wwwabcmmm">
64 <img src="" alt="www-abc-mmm">
65 <img src="" alt="www abc mmm">
66 <img src="" alt="qq">
67
68 </body>
69 </html>

```

通配符选择器

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>22-通配符选择器</title>
6      <style>
7          /*
8              .cc{
9                  color: red;
10             }
11             */
12             *{
13                 color: red;
14             }
15         </style>
16     </head>
17     <body>
18     <!--
19     1. 什么是通配符选择器?
20     作用：给当前界面上所有的标签设置属性
21
22     格式：
23     *{
24         属性:值;
25     }
26
27     注意点：
28     由于通配符选择器是设置界面上所有的标签的属性，所以在设置之前会遍历所有的标签，如果当前界面上的标签比较多，那么性能就会比较差，所以在企业开发中一般不会使用通配符选择器
29     -->
30     <!--
31     <h1 class="cc">我是标题</h1>
32     <p class="cc">我是段落</p>
33     <a href="#" class="cc">我是超链接</a>
34     -->
35     <h1>我是标题</h1>
36     <p>我是段落</p>
37     <a href="#">我是超链接</a>
38
39     </body>
40     </html>

```

css继承性

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>24-CSS三大特性之继承性</title>
6      <style>
7          div{
8              color: red;
9              text-decoration: none;
10             font-size: 30px;
11         }

```



```

12     </style>
13 </head>
14 <body>
15 <!--
16 1. 什么是继承性?
17 作用：给父元素设置一些属性，子元素也可以使用，这个我们就称之为继承性
18
19 注意点：
20 1. 并不是所有的属性都可以继承，只有以color/font-/text-/line-开头的属性才可以继承
21 2. 在CSS的继承中不仅仅是儿子可以继承，只要是后代都可以继承
22 3. 继承性中的特殊性
23 3.1a标签的文字颜色和下划线是不能继承的
24 3.2h标签的文字大小是不能继承的
25
26 应用场景：
27 一般用于设置网页上的一些共性信息，例如网页的文字颜色，字体,文字大小等内容
28 body{}
29 -->
30
31 <div>
32     <p>我是段落</p>
33 </div>
34
35 <div>
36     <ul>
37         <li>
38             <p>我是段落</p>
39         </li>
40     </ul>
41 </div>
42
43 <div>
44     <a href="#">我是超链接</a>
45 </div>
46
47 <div>
48     <h1>我是大标题</h1>
49 </div>
50 </body>
51 </html>

```

层叠性

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>25-CSS三大特性之层叠性</title>
6     <style>
7         p{
8             color: red;
9         }
10        .para{
11            color: blue;
12        }

```

```
13     </style>
14 </head>
15 <body>
16 <!--
17 1. 什么是层叠性?
18 作用：层叠性就是CSS处理冲突的一种能力
19
20 注意点：
21 层叠性只有在多个选择器选中"同一个标签"，然后又设置了"相同的属性"，才会发生层叠性
22
23 CSS全称 Cascading StyleSheet
24 -->
25 <p id="identity" class="para">我是段落</p>
26 </body>
27 </html>
```

优先级

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>26-CSS三大特性之优先级</title>
6     <style>
7         /*
8         li{
9             color: blue;
10        }
11        ul{
12            color: red;
13        }
14        */
15        /*
16        p{
17            color: blue;
18        }
19        p{
20            color: red;
21        }
22        */
23        #identity{
24            color: purple;
25        }
26        .para{
27            color: pink;
28        }
29        p{
30            color: green;
31        }
32        *{
33            color: blue;
34        }
35        li{
36            color: red;
37        }
```

```

38     </style>
39 </head>
40 <body>
41 <!--
42 1. 什么是优先级?
43 作用: 当多个选择器选中同一个标签, 并且给同一个标签设置相同的属性时, 如何层叠就由优先级来确
    定
44
45 2. 优先级判断的三种方式
46 2.1 间接选中就是指继承
47 如果是间接选中, 那么就是谁离目标标签比较近就听谁的
48 2.2 相同选择器(直接选中)
49 如果都是直接选中, 并且都是同类型的选择器, 那么就是谁写在后面就听谁的
50 2.3 不同选择器(直接选中)
51 如果都是直接选中, 并且不是相同类型的选择器, 那么就会按照选择器的优先级来层叠
52 id>类>标签>通配符>继承>浏览器默认
53 -->
54 <ul>
55     <li>
56         <p id="identity" class="para">我是段落</p>
57     </li>
58 </ul>
59 </body>
60 </html>

```

优先级之important

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>27-优先级之important</title>
6     <style>
7         #identity{
8             color: purple;
9             font-size: 50px;
10        }
11        .para{
12            color: pink ;
13        }
14        p{
15            color: green;
16        }
17        *{
18            color: blue !important;
19            font-size: 10px;
20        }
21        li{
22            color: red ;
23        }
24    </style>
25 </head>
26 <body>
27 <!--
28 1. 什么是!important

```

```
29 作用：用于提升某个直接选中标签的选择器中的某个属性的优先级的，可以将被指定的属性的优先级提升为最高
30
31 注意点：
32 1. !important只能用于直接选中，不能用于间接选中
33 2. 通配符选择器选中的标签也是直接选中的
34 3. !important只能提升被指定的属性的优先级，其它的属性的优先级不会被提升
35 4. !important必须写在属性值得分号前面
36 5. !important前面的感叹号不能省略
37
38 -->
39 <ul>
40     <li>
41         <p id="identity" class="para">我是段落</p>
42     </li>
43 </ul>
44 </body>
45 </html>
```

优先级中的权重问题

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>28-优先级之权重问题</title>
6      <style>
7          /*
8              #identity1 .box2{
9                  color: red;
10             }
11             .box1 .box2{
12                 color: green;
13             }
14             div ul li p{
15                 color: blue;
16             }
17             */
18             /*
19             .box1 .box2{
20                 color: blue;
21             }
22             div .box2{
23                 color: green;
24             }
25             */
26             /*
27             #identity1 ul li p{
28                 color: red;
29             }
30             #identity1 ul p{
31                 color: green;
32             }
33             */
34             /*
```

```

35     .box1 li #identity2{
36         color: blue;
37     }
38
39     #identity1 ul .box2{
40         color: red;
41     }
42     */
43
44     .box2{
45         color: red;
46     }
47     li{
48         color: blue;
49     }
50 </style>
51 </head>
52 <body>
53 <!--
54 1.什么是优先级的权重?
55 作用：当多个选择器混合在一起使用时，我们可以通过计算权重来判断谁的优先级最高
56
57 2.权重的计算规则
58 2.1首先计算选择器中有多少个id，id多的选择器优先级最高
59 2.2如果id的个数一样，那么再看类名的个数，类名个数多的优先级最高
60 2.3如果类名的个数一样，那么再看标签名称的个数，标签名称个数多的优先级最高
61 2.4如果id个数一样，类名个数也一样，标签名称个数也一样，那么就不会继续往下计算了，那么此时
    谁写在后面听谁的
62 也就是说优先级如果一样，那么谁写在后面听谁的
63
64 注意点：
65 1.只有选择器是直接选中标签的才需要计算权重，否则一定会听直接选中的选择器的
66 -->
67
68 <div id="identity1" class="box1">
69     <ul>
70         <li>
71             <p id="identity2" class="box2">我是段落</p>
72         </li>
73     </ul>
74 </div>
75 </body>
76 </html>

```

span和div标签

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>29-div和span标签</title>
6     <style>
7         .header{
8             width: 980px;
9             height: 100px;

```

```
10         background: red;
11         margin: auto;
12         margin-bottom: 10px;
13     }
14     .content{
15         width: 980px;
16         height: 500px;
17         background: green;
18         margin: auto;
19         margin-bottom: 10px;
20     }
21     .footer{
22         width: 980px;
23         height: 100px;
24         background: blue;
25         margin: auto;
26     }
27     .logo{
28         width: 200px;
29         height: 50px;
30         background: pink;
31         float: left;
32         margin: 20px;
33     }
34     .nav{
35         width: 600px;
36         height: 50px;
37         background: yellow;
38         float: right;
39         margin: 20px;
40     }
41     .aside{
42         width: 250px;
43         height: 460px;
44         background: purple;
45         float: left;
46         margin: 20px;
47     }
48     .article{
49         width: 650px;
50         height: 460px;
51         background: deepskyblue;
52         float: right;
53         margin: 20px;
54     }
55     span{
56         color: red;
57     }
58 </style>
59 </head>
60 <body>
61 <!--
62 1. 什么是div?
63 作用：一般用于配合css完成网页的基本布局
64
65 2. 什么是span?
66 作用：一般用于配合css修改网页中的一些局部信息
67
```

```
68 3.div和span有什么区别?
69 1.div会单独的占领一行,而span不会单独占领一行
70 2.div是一个容器级的标签, 而span是一个文本级的标签
71
72 4. 容器级的标签和文本级的标签的区别?
73 容器级的标签中可以嵌套其它所有的标签
74 文本级的标签中只能嵌套文字/图片/超链接
75
76 容器级的标签
77 div h ul ol dl li dt dd ...
78
79 文本级的标签
80 span p br strong em ins del ...
81
82 注意点:
83 哪些标签是文本级的哪些标签是容器级的, 我们不用刻意去记忆, 在企业开发中一般情况下要嵌套都是嵌套在div中, 或者按照组标签来嵌套
84 -->
85 <!--
86 <div class="header">
87     <div class="logo"></div>
88     <div class="nav"></div>
89 </div>
90 <div class="content">
91     <div class="aside"></div>
92     <div class="article"></div>
93 </div>
94 <div class="footer"></div>
95 -->
96
97 <!--
98 <p>努力到<span>无能为力</span>, 拼搏到<span>感动自己</span></p>
99 -->
100
101 <!--
102 <div>我是div</div>
103 <div>我是div</div>
104
105 <span>我是span</span>
106 <span>我是span</span>
107 -->
108
109 <!--
110 <div>
111     <div>
112         <ul>
113             <li>
114                 
115             </li>
116         </ul>
117     </div>
118 </div>
119 -->
120
121 <!--
122 <p>我是段落
123     <h1>我是标题</h1>
124 </p>
```

```

125  -->
126
127  <h1>我是标题
128      <p>我是段落</p>
129  </h1>
130  </body>
131  </html>

```

css元素的显示模式

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>30-CSS元素的显示模式</title>
6      <style>
7          div{
8              background: red;
9              width: 200px;
10             height: 200px;
11         }
12         span{
13             background: blue;
14             width: 200px;
15             height: 200px;
16         }
17         img{
18             width: 50px;
19             height: 50px;
20         }
21     </style>
22 </head>
23 <body>
24 <!--
25 在HTML中HTML将所有的标签分为两类，分别是容器级和文本级
26 在CSS中CSS也将所有的标签分为两类，分别是块级元素和行内元素
27
28 1. 什么是块级元素，什么是行内元素？
29 块级元素会独占一行
30 行内元素不会独占一行
31
32 容器级的标签
33 div h ul ol dl li dt dd ...
34 文本级的标签
35 span p b i s t o n g e m i n s d e l ...
36
37 块级元素
38 p div h ul ol dl li dt dd
39 行内元素
40 span b i s t o n g e m i n s d e l
41
42 2. 块级元素和行内元素的区别？
43 2.1 块级元素
44 独占一行
45 如果没有设置宽度，那么默认和父元素一样宽

```



```

46  如果设置了宽高，那么就按照设置的来显示
47
48  2.2行内元素
49  不会独占一行
50  如果没有设置宽度，那么默认和内容一样宽
51  行内元素是不可以设置宽度和高度的
52
53  2.3行内块级元素
54  为了能够让元素既能够不独占一行，又可以设置宽度和高度，那么就出现了行内块级元素
55  -->
56
57  <div>我是div</div>
58  <p>我是段落</p>
59  <h1>我是标题</h1>
60  <hr>
61  <span>我是span</span>
62  <b>我是加粗</b>
63  <strong>我是强调</strong>
64  <hr>
65  
66  
67  </body>
68  </html>

```

背景颜色

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>33-背景颜色</title>
6      <style>
7          div{
8              width: 500px;
9              height: 500px;
10         }
11         .box1{
12             background-color: red;
13         }
14         .box2{
15             background-color: rgb(0,255,0);
16         }
17         .box3{
18             background-color: rgba(0,0,255,0.7);
19         }
20         .box4{
21             background-color: #0ff;
22         }
23     </style>
24 </head>
25 <body>
26 <!--
27 1. 如何设置标签的背景颜色?
28 在CSS中有一个background-color:属性，就是专门用来设置标签的背景颜色的
29

```

```
30 取值：
31 具体单词
32 rgb
33 rgba
34 十六进制
35
36 快捷键：
37 bc background-color: #fff;
38
39 -->
40 <div class="box1"></div>
41 <div class="box2"></div>
42 <div class="box3"></div>
43 <div class="box4"></div>
44 </body>
45 </html>
```

背景图片

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>34-背景图片</title>
6      <style>
7          div{
8              width: 500px;
9              height: 500px;
10         }
11         .box1{
12             background-image: url(images/girl.jpg);
13             /*background-image:
14             url(http://img4.imgtn.bdimg.com/it/u=2278032206,4196312526&fm=21&gp=0.jpg);*
15             /
16         }
17     </style>
18 </head>
19 <body>
20 <!--
21 1. 如何设置背景图片？
22 在CSS中有一个叫做background-image: url();的属性，就是专门用于设置背景图片的
23
24 快捷键：
25 bi background-image: url();
26
27 注意点：
28 1. 图片的地址必须放在url()中，图片的地址可以是本地的地址，也可以是网络的地址
29 2. 如果图片的大小没有标签的大小大，那么会自动在水平和垂直方向平铺来填充
30 3. 如果网页上出现了图片，那么浏览器会再次发送请求获取图片
31 -->
32 <div class="box1"></div>
33 </body>
34 </html>
```

背景平铺属性

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>35-背景平铺属性</title>
6   <style>
7     /*
8     div{
9       width: 500px;
10      height: 500px;
11    }
12    .box1{
13      background-image: url(images/girl.jpg);
14      background-repeat: repeat-y;
15    }
16    */
17    /*
18    body{
19      background-image: url(images/bg2.jpg);
20    }
21    */
22    div{
23      width: 980px;
24      height: 60px;
25      background-image: url(images/1.png);
26    }
27  </style>
28 </head>
29 <body>
30 <!--
31 1. 如何控制背景图片的平铺方式？
32 在CSS中有一个background-repeat属性，就是专门用于控制背景图片的平铺方式的
33
34 取值：
35 repeat 默认，在水平和垂直都需要平铺
36 no-repeat 在水平和垂直都不需要平铺
37 repeat-x 只在水平方向平铺
38 repeat-y 只在垂直方向平铺
39
40 快捷键
41 bgr background-repeat:
42
43 应用场景：
44 可以通过背景图片的平铺来降低图片的大小，提升网页的访问速度
45 -->
46 <div class="box1"></div>
47 </body>
48 </html>
```

1. 如何控制背景图片的平铺方式？

在CSS中有一个background-repeat属性，就是专门用于控制背景图片的平铺方式的

取值：

- repeat 默认，在水平和垂直都需要平铺
- no-repeat 在水平和垂直都不需要平铺
- repeat-x 只在水平方向平铺
- repeat-y 只在垂直方向平铺

快捷键

bgr background-repeat:

应用场景：

可以通过背景图片的平铺来降低图片的大小，提升网页的访问速度

背景定位属性

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>36-背景定位属性</title>
6   <style>
7     div{
8       /*width: 500px;*/
9       height: 500px;
10    }
11    .box1{
12      /*background-color: red;*/
13      /*background-image: url(images/girl.jpg);*/
14      /*background-repeat: no-repeat;*/
15      /*background-position: left top;*/
16      /*background-position: right top;*/
17      /*background-position: right bottom;*/
18      /*background-position: left bottom;*/
19      /*background-position: center center;*/
20      /*background-position: left center;*/
21      /*background-position: center top;*/
22      /*background-position: 100px 0;*/
23      /*background-position: 100px 200px;*/
24      background-position: -100px -100px;
25      background-image: url(images/yx1m.jpg);
26      /*background-position: center top;*/
27    }
28  </style>
29 </head>
30 <body>
31 <!--
32 1.如何控制背景图片的位置?
33 在CSS中有一个叫做background-position:属性，就是专门用于控制背景图片的位置
34
35 2.格式:
36 background-position: 水平方向 垂直方向;
37
38 3.取值
39 2.1具体的方位名词
40 水平方向: left center right
41 垂直方向: top center bottom
42
43 2.2具体的像素
44 例如: background-position: 100px 200px;
45 记住一定要写单位，也就是一定要写px
46 记住具体的像素是可以接收负数的
47
48 快捷键:
49 bp background-position: 0 0;
50
51 注意点:
52 同一个标签可以同时设置背景颜色和背景图片，如果颜色和图片同时存在，那么图片会覆盖颜色
53 -->
54
55 <div class="box1">
56   <!---->
57 </div>
```

```
58 </body>
59 </html>
```

背景关联方式和背景缩写

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>37-背景缩写</title>
6   <style>
7     div{
8       /*width: 500px;*/
9       /*height: 500px;*/
10      /*
11       background-color: red;
12       background-image: url(images/girl.jpg);
13       background-repeat: no-repeat;
14       background-position: left bottom;
15       */
16      /*background: red url(images/girl.jpg) no-repeat left bottom;*/
17      /*background: red;*/
18      /*background: url(images/girl.jpg);*/
19    }
20
21    body{
22      background-image: url(images/girl.jpg);
23      background-repeat: no-repeat;
24      /*background-attachment: scroll;*/
25      background-attachment: fixed;
26    }
27  </style>
28 </head>
29 <body>
30 <!--
31 1. 背景属性缩写的格式
32 background: 背景颜色 背景图片 平铺方式 关联方式 定位方式;
33
34 快捷键:
35 bg+ background: #fff url() 0 0 no-repeat;
36
37 2. 注意点:
38 background属性中, 任何一个属性都可以被省略
39
40 3. 什么是背景关联方式?
41 默认情况下背景图片会随着滚动条的滚动而滚动, 如果不想让背景图片随着滚动条的滚动而滚动, 那
42 么我们就可以修改背景图片和滚动条的关联方式
43
44 4. 如何修改背景关联方式?
45 在CSS中有一个叫做background-attachment的属性, 这个属性就是专门用于修改关联方式的
46 格式
47 background-attachment: scroll;
48
49 取值:
```

```
50 scroll 默认值，会随着滚动条的滚动而滚动
51 fixed 不会随着滚动条的滚动而滚动
52
53 快捷键：
54 ba background-attachment;;
55 -->
56
57 <div></div>
58 我是文字
59 </body>
60 </html>
```

背景图片和插入图片的区别

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>38-背景图片和插入图片区别</title>
6     <style>
7         div{
8             /*width: 150px;*/
9             /*height: 170px;*/
10            width: 300px;
11            height: 300px;
12            background-color: red;
13        }
14        .box1{
15            background-image: url(images/girl.jpg);
16            background-repeat:no-repeat;
17            background-position: right bottom;
18        }
19    </style>
20 </head>
21 <body>
22 <!--
23 1. 背景图片和插入图片区别？
24 1.1
25 背景图片仅仅是一个装饰，不会占用位置
26 插入图片会占用位置
27
28 1.2
29 背景图片有定位属性，所以可以很方便的控制图片的位置
30 插入图片没有定位属性，所有控制图片的位置不太方便
31
32 1.3
33 插入图片的语义比背景图片的语义要强，所以在企业开发中如果你的图片想被搜索引擎收录，那么推荐
    使用插入图片
34 -->
35 <div class="box1">我是文字</div>
36 <div class="box2">
37     
38     我是文字
39 </div>
40 </body>
```

背景图片练习

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>39-背景图片练习</title>
6      <style>
7          .box1{
8              width: 1024px;
9              height: 768px;
10             background-image: url(images/bybg.jpg);
11         }
12         .box2{
13             /*background-color: red;*/
14             width: 1024px;
15             height: 768px;
16             background-image: url(images/bybottom.png);
17             background-position: center bottom;
18             background-repeat: no-repeat;
19             /*background-position: 100px 200px;*/
20         }
21     </style>
22 </head>
23 <body>
24     <div class="box1">
25         <div class="box2"></div>
26     </div>
27 </body>
28 </html>
```

css精灵图

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>40-CSS精灵图</title>
6      <style>
7          .box{
8              width: 86px;
9              height: 28px;
10             background-image: url(images/weibo.png);
11             background-position: -425px -200px;
12         }
13     </style>
14 </head>
15 <body>
16     <!--
17     1. 什么是CSS精灵图
```

```
18 CSS精灵图是一种图像合成技术
19
20 2.CSS精灵图作用
21 可以减少请求的次数，以及可以降低服务器处理压力
22
23 3.如何使用CSS精灵图
24 CSS的精灵图需要配合背景图片和背景定位来使用
25 -->
26 <div class="box"></div>
27 </body>
28 </html>
```

精灵图练习

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>41-CSS精灵图练习</title>
6     <style>
7         div{
8             background-image: url(images/a-z.jpg);
9             display: inline-block;
10        }
11        .box1{
12            width: 110px;
13            height: 143px;
14            background-position: -560px -216px;
15        }
16        .box2{
17            width: 127px;
18            height: 140px;
19            background-position: -834px -217px;
20        }
21        .box3{
22            width: 124px;
23            height: 144px;
24            background-position: -264px -215px;
25        }
26
27     </style>
28 </head>
29 <body>
30 <div class="box1"></div>
31 <div class="box2"></div>
32 <div class="box3"></div>
33 </body>
34 </html>
```

边框属性

```
1 <!DOCTYPE html>
```



```

2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>42-边框属性上</title>
6   <style>
7     .box{
8       width: 100px;
9       height: 100px;
10      background-color: red;
11      /*border: 5px solid blue;*/
12      /*border: 5px solid;*/
13      /*border: 5px blue;*/
14      /*border: solid blue;*/
15
16      border-top:5px solid blue;
17      border-right:10px dashed green;
18      border-bottom:15px dotted purple;
19      border-left:20px double pink;
20    }
21  </style>
22 </head>
23 <body>
24 <!--
25 1.什么边框?
26 边框就是环绕在标签宽度和高度周围的线条
27
28 2.边框属性的格式
29 2.1连写(同时设置四条边的边框)
30 border: 边框的宽度 边框的样式 边框的颜色;
31
32 快捷键:
33 bd+ border: 1px solid #000;
34
35 注意点:
36 1.连写格式中颜色属性可以省略, 省略之后默认就是黑色
37 2.连写格式中样式不能省略, 省略之后就看不到边框了
38 3.连写格式中宽度可以省略, 省略之后还是可以看到边框
39
40 2.2连写(分别设置四条边的边框)
41 border-top: 边框的宽度 边框的样式 边框的颜色;
42 border-right: 边框的宽度 边框的样式 边框的颜色;
43 border-bottom: 边框的宽度 边框的样式 边框的颜色;
44 border-left: 边框的宽度 边框的样式 边框的颜色;
45
46 快捷键:
47 bt+ border-top: 1px solid #000;
48 br+
49 bb+
50 bl+
51 -->
52 <div class="box"></div>
53 </body>
54 </html>

```

```

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

```

```

4      <meta charset="UTF-8">
5      <title>43-边框属性下</title>
6      <style>
7          .box{
8              width: 500px;
9              height: 500px;
10             background-color: red;
11             /*border-width: 5px 10px 15px 20px;*/
12             /*border-style: solid dashed dotted double;*/
13             /*border-color: blue green purple pink;*/
14             /*border-color: blue green purple;*/
15             /*border-color: blue green;*/
16             /*border-color: blue;*/
17
18             border-top-width: 5px;
19             border-top-style: solid;
20             border-top-color: blue;
21
22             border-right-width: 10px;
23             border-right-style: dashed;
24             border-right-color: green;
25
26             border-bottom-width: 15px;
27             border-bottom-style: dotted;
28             border-bottom-color: purple;
29
30             border-left-width: 20px;
31             border-left-style: double;
32             border-left-color: pink;
33         }
34     </style>
35 </head>
36 <body>
37 <!--
38 2.3连写(分别设置四条边的边框)
39 border-width: 上 右 下 左;
40 border-style: 上 右 下 左;
41 border-color: 上 右 下 左;
42
43 注意点:
44 1.这三个属性的取值是按照顺时针来赋值, 也就是按照上右下左来赋值, 而不是按照日常生活中的上下
   左右
45 2.这三个属性的取值省略时的规律
46 2.1上 右 下 左 > 上 右 下 > 左边的取值和右边的一样
47 2.2上 右 下 左 > 上 右 > 左边的取值和右边的一样 下边的取值和上边一样
48 2.3上 右 下 左 > 上 > 右下左边取值和上边一样
49
50
51 3.非连写(方向+要素)
52 border-left-width: 20px;
53 border-left-style: double;
54 border-left-color: pink;
55 -->
56 <div class="box"></div>
57 </body>
58 </html>

```

内边距属性

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>45-内边距属性</title>
6   <style>
7     div{
8       width: 98px;
9       height: 90px;
10      border: 1px solid #000;
11      background-color: red;
12    }
13    .box1{
14      padding-top: 20px;
15    }
16    .box2{
17      padding-right:40px;
18    }
19    .box3{
20      padding-bottom:80px;
21    }
22    .box4{
23      padding-left:160px;
24    }
25    .box5{
26      /*padding:20px 40px 80px 160px;*/
27      /*padding:20px 40px 80px;*/
28      /*padding:20px 40px;*/
29      padding:20px;
30    }
31  </style>
32 </head>
33 <body>
34 <!--
35 1.什么是内边距?
36 边框和内容之间的距离就是内边距
37
38 2.格式
39 2.1非连写
40 padding-top: ;
41 padding-right: ;
42 padding-bottom: ;
43 padding-left: ;
44
45 2.2连写
46 padding: 上 右 下 左;
47
48 2.这三个属性的取值省略时的规律
49 2.1上 右 下 左 > 上 右 下 > 左边的取值和右边的一样
50 2.2上 右 下 左 > 上 右 > 左边的取值和右边的一样 下边的取值和上边一样
51 2.3上 右 下 左 > 上 > 右下左边取值和上边一样
52
53 注意点:
```

```

54 1. 给标签设置内边距之后， 标签占有的宽度和高度会发生变化
55 2. 给标签设置内边距之后， 内边距也会有背景颜色
56
57 -->
58 <div class="box1">我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字</div>
59 <hr>
60 <div class="box2">我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字</div>
61 <hr>
62 <div class="box3">我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字</div>
63 <hr>
64 <div class="box4">我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字</div>
65 <hr>
66 <div class="box5">我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字</div>
67 </body>
68 </html>

```

外边距属性

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>46-外边距属性</title>
6      <style>
7          *{
8              padding:0;
9              margin:0;
10         }
11         span{
12             display: inline-block;
13             width: 100px;
14             height: 100px;
15             border: 1px solid #000;
16             background-color: red;
17         }
18         div{
19             height: 100px;
20             border: 1px solid #000;
21         }
22         .box1{
23             /*
24             margin-top:20px;
25             margin-right:40px;
26             margin-bottom:80px;
27             margin-left:160px;
28             */
29             /*margin:20px 40px 80px 160px;*/
30             /*margin:20px 40px 80px;*/
31             /*margin:20px 40px;*/
32             margin:20px;
33         }
34
35     </style>
36 </head>
37 <body>

```

```

38 <!--
39 1.什么是外边距?
40 标签和标签之间的距离就是外边距
41
42 2.格式
43 2.1非连写
44 margin-top: ;
45 margin-right: ;
46 margin-bottom: ;
47 margin-left: ;
48
49 2.2连写
50 margin: 上 右 下 左;
51
52 2.这三个属性的取值省略时的规律
53 2.1上 右 下 左 > 上 右 下 > 左边的取值和右边的一样
54 2.2上 右 下 左 > 上 右 > 左边的取值和右边的一样 下边的取值和上边一样
55 2.3上 右 下 左 > 上 > 右下左边取值和上边一样
56
57 注意点:
58 外边距的那一部分是没有背景颜色的
59 -->
60 <span class="box1">我是span</span><span class="box2">我是span</span><span
  class="box3">我是span</span><div class="box4"></div>
61
62 </body>
63 </html>

```

垂直方向外边距的合并现象

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>47-外边距合并现象</title>
6   <style>
7     span{
8       display: inline-block;
9       width: 100px;
10      height: 100px;
11      border: 1px solid #000;
12    }
13    div{
14      height: 100px;
15      border: 1px solid #000;
16    }
17    .hezi1{
18      margin-right:50px;
19    }
20    .hezi2{
21      margin-left:100px;
22    }
23    .box1{
24      margin-bottom:50px;
25    }

```

```

26         .box2{
27             margin-top:100px;
28         }
29     </style>
30 </head>
31 <body>
32 <!--
33 在默认布局的垂直方向上，默认情况下外边距是不会叠加的，会出现合并现象，谁的外边距比较大就听
    谁的
34 -->
35 <span class="hezi1">我是span</span><span class="hezi2">我是span</span>
36 <div class="box1">我是div</div>
37 <div class="box2">我是div</div>
38 </body>
39 </html>

```

css盒子模型

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>48-CSS盒子模型</title>
6      <style>
7          span,a,b,strong{
8              display: inline-block;
9              width: 100px;
10             height: 100px;
11             border: 6px solid #000;
12             padding: 20px;
13             margin: 20px;
14         }
15
16     </style>
17 </head>
18 <body>
19 <!--
20 1. 什么是CSS盒子模型?
21 CSS盒子模型仅仅是一个形象的比喻，HTML中所有的标签都是盒子
22
23 结论
24 1. 在HTML中所有的标签都可以设置
25 宽度/高度 == 指定可以存放内容的区域
26 内边距 == 填充物
27 边框 == 手机盒子自己
28 外边距 == 盒子和盒子之间的间隙
29 -->
30 <span>我是span</span>
31 <a href="#">我是超链接</a>
32 <b>我是加粗</b>
33 <strong>我是强调</strong>
34
35 </body>
36 </html>

```

盒子模型的宽度和高度

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>49-盒子模型宽度和高度</title>
6  </head>
7  <body>
8      <!--
9      1. 内容的宽度和高度
10     就是通过width/height属性设置的宽度和高度
11
12     2. 元素的宽度和高度
13     宽度 = 左边框 + 左内边距 + width + 右内边距 + 右边框
14     高度 同理可证
15
16     3. 元素空间的宽度和高度
17     宽度 = 左外边距 + 左边框 + 左内边距 + width + 右内边距 + 右边框 + 右外边距
18     高度 同理可证
19     -->
20 </body>
21 </html>
```

盒子模型宽高练习

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>50-盒子宽高练习</title>
6      <style>
7          /*判断是否是内容宽高为100的盒子
8          只需要看width/height是否为100即可*/
9          .box1{
10              width: 50px;
11              height: 50px;
12              padding: 25px;
13              background-color: green;
14          }
15          .box2{
16              width: 96px;
17              height: 96px;
18              border: 2px solid #000;
19          }
20          .box3{
21              width: 100px;
22              height: 100px;
23              background-color: yellow;
24          }
25          /*判断是否是元素宽高为200的盒子
26          只需要看 边框 + 内边距 + 内容宽度/内容高度 时候等于200即可*/
```

```

27     .box4{
28         width: 100px;
29         height: 100px;
30         margin: 50px;
31         background-color: red;
32     }
33     .box5{
34         /*25+25+150 = 200*/
35         width: 150px;
36         height: 150px;
37         border: 25px solid #000;
38     }
39     .box6{
40         /*25+25+150 = 200*/
41         /*5+5+20+20+150=200*/
42         width: 150px;
43         height: 150px;
44         /*padding: 25px;*/
45         padding:20px;
46         border: 5px solid #000;
47         background-color: blue;
48     }
49     /*判断是否是元素空间宽高为300的盒子
50     只需要看 外边距+边框+内边距+内容宽度/内容高度*/
51     .box7{
52         width: 300px;
53         height: 300px;
54         background-color: purple;
55     }
56     .box8{
57         /*
58         50+50+25+25+25+25+100
59         100 +50 +50 + 100
60         300
61         */
62         width: 100px;
63         height: 100px;
64         padding: 25px;
65         border: 25px solid #000;
66         margin: 50px;
67     }
68     /*现有如下盒子模型，要求增加padding属性为25之后仍然保持元素宽高为200
69
70     元素宽高 = 边框 + 内边距 + 内容宽高
71             = 0 + 0 + 200 = 200
72             = 0 + 25 + 25 + 200 = 250
73             = 0 + 25 + 25 + 150 = 200
74
75     规律：
76     1.增加了padding之后元素的宽高也会发生变化
77     2.如果增加了padding之后还想保持元素的宽高，那么就必须减去内容的宽高
78     */
79     .box9{
80         width: 150px;
81         height: 150px;
82         padding: 25px;
83         background-color: red;
84     }
85     /*现有如下盒子模型，要求增加border属性为20之后仍然保持元素宽高为200

```



```

85     元素宽高 = 边框 + 内边距 + 内容宽高
86             = 0 + 0 + 200 = 200
87             = 0 + 20 + 20 + 200 = 240
88             = 0 + 20 + 20 + 160 = 200
89     规律：
90     1. 增加了border之后元素的宽高也会发生变化
91     2. 如果增加了border之后还想保持元素的宽高，那么就必须减去内容的宽高
92     */
93     .box10{
94         width: 160px;
95         height: 160px;
96         border: 20px solid #000;
97         background-color: deepskyblue;
98     }
99     </style>
100 </head>
101 <body>
102 <!--<div class="box1">box1</div>-->
103 <!--<div class="box2">box2</div>-->
104 <!--<div class="box3">box3</div>-->
105 <!--<hr>-->
106 <!--<div class="box4">box4</div>-->
107 <!--<div class="box5">box5</div>-->
108 <!--<div class="box6">box6</div>-->
109 <!--<hr>-->
110 <!--<div class="box7">box7</div>-->
111 <!--<div class="box8">box8</div>-->
112 <!--<hr>-->
113 <div class="box9">box9</div>
114 <div class="box10">box10</div>
115 </body>
116 </html>

```

box-sizing属性

```

1  <!--<!DOCTYPE html>-->
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>51-盒子box-sizing属性</title>
6      <style>
7          .content{
8              width: 300px;
9              height: 300px;
10             background-color: red;
11         }
12         .aside{
13             width: 100px;
14             height: 200px;
15             background-color: green;
16             float: left;
17         }
18         .article{
19             /*

```

```

20      1.CSS3中新增了一个box-sizing属性，这个属性可以保证我们给盒子新增padding
    和border之后，盒子元素的宽度和高度不变
21      2.box-sizing属性是如何保证增加padding和border之后，盒子元素的宽度和高度
    不变
22      和我们前面学习的原理一样，增加padding和border之后要想保证盒子元素的宽高不
    变，那么就必须减去一部分内容的宽度和高度
23      3.box-sizing取值
24      3.1content-box
25      元素的宽高 = 边框 + 内边距 + 内容宽高
26      3.2border-box
27      元素的宽高 = width/height的宽高
28      */
29      box-sizing: border-box;
30      width: 200px;
31      height: 200px;
32      background-color: blue;
33      float: right;
34      border: 20px solid #000;
35      padding: 20px;
36    }
37  </style>
38 </head>
39 <body>
40
41 <div class="content">
42   <div class="aside"></div>
43   <div class="article"></div>
44 </div>
45
46 </body>
47 </html>

```

盒子居中和内容居中

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>53-盒子居中和内容居中</title>
6    <style>
7      .father{
8        width: 800px;
9        height: 500px;
10       background-color: red;
11       /*text-align: center;*/
12       margin:0 auto;
13     }
14     .son{
15       width: 100px;
16       height: 100px;
17       background-color: blue;
18     }
19   </style>
20 </head>
21 <body>

```

```

22 <!--
23 1.text-align:center;和margin:0 auto;区别
24 text-align: center;作用
25 设置盒子中存储的文字/图片水平居中
26
27 margin:0 auto;作用
28 让盒子自己水平居中
29 -->
30 <div class="father">
31     我是文字<br/>
32     
33     <!--<div class="son"></div>-->
34 </div>
35 </body>
36 </html>

```

清空默认边距

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>54-清空默认边距</title>
6     <style>
7         /*
8         *{
9             margin: 0;
10            padding: 0;
11        }
12        */
13
14        body,div,d1,dt,dd,u1,o1,li,h1,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend,
15        input,textarea,p,blockquote,th,td{
16            margin:0;padding:0
17        }
18        p{
19            width: 610px;
20            height: 110px;
21            background-color: #cdcdcd;
22            border: 1px solid #000000;
23        }
24    </style>
25 </head>
26 <body>
27 <!--
28 1.为什么要清空默认边距(外边距和内边距)
29 在企业开发中为了更好的控制盒子的宽高和计算盒子的宽高等等，所以在企业开发中，编写代码之前第一件事情就是清空默认的边距
30
31 2.如何清空默认的边距
32 格式
33 *{
34     margin: 0;
35     padding: 0;
36 }

```

```
35
36 3. 注意点
37 通配符选择器会找到(遍历)当前界面中所有的标签，所以性能不好
38 企业开发中可以从这个网址中拷贝
39 http://yui.yahooapis.com/3.18.1/build/cssreset/cssreset-min.css
40
41 -->
42 <p>葬爱:非主流文化的常用词，是当今网络流行术语。且流行于非主流杀马特之中。葬，即埋葬，爱，
    即爱情，翻译成外语就bury love</p>
43 </body>
44 </html>
```

行高和字号

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>54-清空默认边距</title>
6     <style>
7
8         body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend,
            input,textarea,p,blockquote,th,td{
9             margin:0;padding:0
10        }
11        p{
12            width: 610px;
13            height: 110px;
14            background-color: #cdcdcd;
15            border: 1px solid #000000;
16        }
17        div{
18            box-sizing: border-box;
19            width: 100px;
20            height: 80px;
21            border: 1px solid #000;
22            line-height: 20px;
23            /*line-height: 80px;*/
24            padding-top:20px;
25            padding-bottom:20px;
26        }
27    </style>
28 </head>
29 <body>
30 <!--
31 1. 什么是行高?
32 在CSS中所有的行都有自己的行高
33
34 注意点:
35 行高和盒子高不是同一个概念
36 行高指的是每行内容的高度
37 盒子高指的是元素的高度
38
39 规律:
40 1. 文字在行高中默认是垂直居中的
```

```

40
41 2.在企业开发中我们经常将盒子的高度和行高设置为一样，那么这样就可以保证一行文字在盒子的高度
42 中是垂直居中的
43 简而言之就是：要想一行文字在盒子中垂直居中，那么只需要设置这行文字的"行高等于盒子的高"即可
44
45 3.在企业开发中如果一个盒子中有多行文字，那么我们就不能使用设置行高等于盒子高来实现让文字垂
46 直居中，只能通过设置padding来让文字居中
47 -->
48
49 <!--<p>葬爱:非主流文化的常用词，是当今网络流行术语。且流行于非主流杀马特之中。葬，即埋葬，
50 爱，即爱情，翻译成外语就bury love</p>-->
51 <!--<div>我是文字我是文字我是文字</div>-->
52 <!--<div>我是文字</div>-->
53 <div>我是文字我是文字我是文字</div>
54 </body>
55 </html>

```

还原字体和字号

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>54-清空默认边距</title>
6      <style>
7
8          body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend,
9          input,textarea,p,blockquote,th,td{
10              margin:0;padding:0
11          }
12          p{
13              box-sizing: border-box;
14              width: 610px;
15              height: 110px;
16              background-color: #cdcdcd;
17              border: 1px solid #000000;
18              font-family:"黑体";
19              font-size: 20px;
20              line-height: 40px;
21              color: #67676d;
22              padding-left: 10px;
23              padding-right: 10px;
24              padding-top: 10px;
25          }
26      </style>
27 </head>
28 <body>
29 <!--
30 注意点：
31 1.在企业开发中，如果一个盒子中存储的是文字，那么一般情况下我们会以盒子左边的内边距为基准，
32 不会以右边的内边距为基准，因为这个右边的内边距有误差
33
34 2.右边内边距的误差从何而来？因为右边如果放不下一个文字，那么文字就会换行显示，所以文字和内
35 边距之间的距离就有了误差
36

```

```
33 3.顶部的内边距并不是边框到文字顶部的距离，而是边框到行高顶部的距离
34 -->
35
36 <p>葬爱:非主流文化的常用词，是当今网络流行术语.且流行于非主流杀马特之中。葬，即埋葬，爱，
    即爱情，翻译成外语就bury love</p>
37
38 </body>
39 </html>
```

文章界面

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>57-文章界面</title>
6      <style>
7
8          body,div,d1,dt,dd,u1,o1,li,h1,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend,
          input,textarea,p,blockquote,th,td{
9              margin:0;padding:0
10         }
11         body{
12             background-color: #efefef;
13         }
14         div{
15             margin:0 auto;
16             box-sizing: border-box;
17             width: 372px;
18             height: 232px;
19             border: 1px solid #666;
20             padding: 15px;
21         }
22         h1{
23             font-family:"微软雅黑";
24             font-size: 18px;
25             border-bottom: 1px solid #666;
26             padding-bottom: 10px;
27         }
28         span{
29             font-size: 14px;
30         }
31         ul{
32             list-style: none;
33             margin-top: 10px;
34         }
35         ul li{
36             line-height: 30px;
37             border-bottom: 1px dashed #666;
38             font-family:"微软雅黑";
39             font-size: 12px;
40             color: #242424;
41             padding-left: 20px;
42         }
43     </style>
```

```

43 </head>
44 <body>
45 <!--
46 1.清空所有的边距
47 2.从外向内， 从上至下的编写网页
48 -->
49 <div>
50   <h1>最新文章<span>/New Articles</span></h1>
51   <ul>
52     <li>问中国：经济新活力篇 治国理政 长征</li>
53     <li>问中国：民生保障不是"清谈馆"国平 国庆节</li>
54     <li>李克强不忘棚户区改造10日至12日将视察澳门
55     </li>
56     <li>王岐山监督中央部委五大抓手：关键少数要加码</li>
57     <li>李南江老师免费公开H5+跨平台开发教学视频</li>
58   </ul>
59 </div>
60 </body>
61 </html>

```

网页的布局方式

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>58-网页的布局方式</title>
6    <style>
7      /*
8      div,h1,p{
9        border: 1px solid #000;
10     }
11     span,strong,b{
12       border: 1px solid #000;
13     }
14     */
15     *{
16       margin: 0;
17       padding: 0;
18     }
19     .box1{
20       width: 100px;
21       height: 100px;
22       background-color: red;
23       /*display: inline-block;*/
24       float: left;
25     }
26     .box2{
27       width: 100px;
28       height: 100px;
29       background-color: blue;
30       /*display: inline-block;*/
31       /*margin-left:930px;*/
32       float: right;
33     }

```

```

34     </style>
35 </head>
36 <body>
37 <!--
38 1. 什么是网页的布局方式?
39 网页的布局方式其实就是指浏览器是如何对网页中的元素进行排版的
40
41 1. 标准流(文档流/普通流)排版方式
42 1.1 其实浏览器默认的排版方式就是标准流的排版方式
43 1.2 在CSS中将元素分为三类， 分别是块级元素/行内元素/行内块级元素
44 1.3 在标准流中有两种排版方式， 一种是垂直排版， 一种是水平排版
45 垂直排版， 如果元素是块级元素， 那么就会垂直排版
46 水平排版， 如果元素是行内元素/行内块级元素， 那么就会水平排版
47
48 2. 浮动流排版方式
49 2.1 浮动流是一种"半脱离标准流"的排版方式
50 2.2 浮动流只有一种排版方式， 就是水平排版。 它只能设置某个元素左对齐或者右对齐
51
52 注意点：
53 1. 浮动流中没有居中对齐， 也就是没有center这个取值
54 2. 在浮动流中是不可以使用margin: 0 auto;
55
56 特点：
57 1. 在浮动流中是不区分块级元素/行内元素/行内块级元素的
58 无论是级元素/行内元素/行内块级元素都可以水平排版
59 2. 在浮动流中无论是块级元素/行内元素/行内块级元素都可以设置宽高
60 3. 综上所述， 浮动流中的元素和标准流中的行内块级元素很像
61
62 3. 定位流排版方式
63 -->
64 <!--
65 <div>我是div</div>
66 <h1>我是标题</h1>
67 <p>我是段落</p>
68 -->
69 <!--
70 <span>我是span</span>
71 <strong>我是强调</strong>
72 <b>我是加粗</b>
73 -->
74 <span class="box1"></span>
75 <span class="box2"></span>
76 </body>
77 </html>

```

浮动元素的脱标

```

1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8">
6     <title>59-浮动元素的脱标</title>
7     <style>
8         .box1{

```



```

9         float: left;
10        width: 100px;
11        height: 100px;
12        background-color: red;
13    }
14    .box2{
15        width: 150px;
16        height: 150px;
17        background-color: blue;
18    }
19    </style>
20 </head>
21 <body>
22 <!--
23 1.什么是浮动元素的脱标?
24 脱标: 脱离标准流
25 当某一个元素浮动之后, 那么这个元素看上去就像被从标准流中删除了一样, 这个就是浮动元素的脱标
26
27 2.浮动元素脱标之后会有什么影响?
28 如果前面一个元素浮动了, 而后面一个元素没有浮动, 那么这个时候前面一个元就会盖住后面一个元
   素
29 -->
30 <div class="box1"></div>
31 <div class="box2"></div>
32 </body>
33 </html>

```

浮动元素的排序规则

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>60-浮动元素排序规则</title>
6      <style>
7          .box1{
8              float: left;
9              width: 50px;
10             height: 50px;
11             background-color: red;
12         }
13         .box2{
14             width: 100px;
15             height: 100px;
16             background-color: pink;
17         }
18         .box3{
19             float: left;
20             width: 150px;
21             height: 150px;
22             background-color: yellow;
23         }
24         .box4{
25             float: left;
26             width: 200px;

```

```

27         height: 200px;
28         background-color: tomato;
29     }
30 </style>
31 </head>
32 <body>
33 <!--
34 1. 浮动元素排序规则
35 1.1 相同方向上的浮动元素，先浮动的元素会显示在前面，后浮动的元素会显示在后面
36 1.2 不同方向上的浮动元素，左浮动会找左浮动，右浮动会找右浮动
37 1.3 浮动元素浮动之后的位置，由浮动元素浮动之前在标准流中的位置来确定
38 -->
39 <div class="box1">1</div>
40 <div class="box2">2</div>
41 <div class="box3">3</div>
42 <div class="box4">4</div>
43 </body>
44 </html>

```

浮动元素的贴靠现象

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>61-浮动元素贴靠现象</title>
6     <style>
7         .father{
8             width: 400px;
9             height: 400px;
10            border: 1px solid #000;
11        }
12        .box1{
13            float: left;
14            width: 50px;
15            height: 300px;
16            background-color: red;
17        }
18        .box2{
19            float: left;
20            width: 50px;
21            height: 100px;
22            background-color: green;
23        }
24        .box3{
25            float: left;
26            width: 350px;
27            height: 100px;
28            background-color: blue;
29        }
30    </style>
31 </head>
32 <body>
33 <!--
34 1. 什么是浮动元素贴靠现象？

```

```
35 1.如果父元素的宽度能够显示所有浮动元素，那么浮动的元素会并排显示
36 2.如果父元素的宽度不能显示所有浮动元素，那么会从最后一个元开始往前贴靠
37 3.如果贴靠了前面所有浮动元素之后都不能显示，最终会贴靠到父元素的左边或者右边
38 -->
39 <div class="father">
40     <div class="box1"></div>
41     <div class="box2"></div>
42     <div class="box3"></div>
43 </div>
44 </body>
45 </html>
```

浮动元素的自围现象

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>62-浮动元素字围现象</title>
6      <style>
7          div{
8              float: left;
9              width: 100px;
10             height: 100px;
11             /*background-color: red;*/
12             border: 1px solid #000;
13         }
14         p{
15             width: 500px;
16             height: 500px;
17             background-color: yellow;
18         }
19         img{
20             float: left;
21         }
22     </style>
23 </head>
24 <body>
25 <!--
26 1.什么是浮动元素字围现象?
27 浮动元素不会挡住没有浮动元素中的文字，没有浮动的文字会自动给浮动的元素让位置,这个就是浮动元
   素字围现象
28 -->
29
30 <!--<div></div>-->
31 
32 <p>1999年-2002年范冰冰一共出演了《青春出动》、《小李飞刀》、《秦始皇》、《中关村风云》、
   《少年包青天2》《尘埃落定》等十七部电视剧[12]，以及《河东狮吼》等三部电影[13]。
33     2003年在由梁羽生小说改编的电视剧《萍踪侠影》中饰演女主角云蕾。同年在冯小刚执导的贺岁档
   电影《手机》中饰演女主角武月[14]。
34     2004年凭借电影《手机》获得第27届大众电影百花奖最佳女主角奖，同时《手机》也获得大众电
   影百花奖最佳故事片奖。[2] 9月，出演根据古龙小说改编的古装剧《小鱼儿与花无缺》，饰演女主角
   铁心兰[15]。同年出演《大唐芙蓉园》中的杨玉环等五部电视剧，以及电影《千机变2》和《情癫大
   圣》[16]。
```

35	2005年发行首张个人专辑《刚刚开始》，这张处女大碟由圈内多位音乐人联袂制作，包含了多种风格迥异的音乐元素。[3] 同年主演由张之亮执导的古装片《墨攻》，饰演女主角逸悦。[17]
36	2006年出演电视剧《封神榜之凤鸣岐山》，饰演女主角苏妲己。之后接连拍摄《苹果》、《导火线》、《心中有鬼》等六部电影[18] 。
37	2007年2月，主演的电影《苹果》入围第57届柏林国际电影节主
38	范冰冰
39	范冰冰(23张)
40	竞赛单元，导演李玉，女主演范冰冰，男主演佟大为共同出席本届电影节。[19] 6月，范冰冰与华谊约满，自组工作室，投资拍摄民国剧《胭脂雪》，并首次担当制片人，同时饰演女主角文玉禾[20] 。10月，凭借电影《苹果》获得第四届欧亚国际电影节最佳女演员奖。[21] 同年出演《合约情人》、《精舞门》、《新宿事件》等五部电影[22] 。同年凭借电影《心中有鬼》获得第44届台湾电影金马奖最佳女配角[4] 。
41	
42	</body>
43	</html>

浮动练习

1	<!DOCTYPE html>
2	<html lang="en">
3	<head>
4	<meta charset="UTF-8">
5	<title>63-浮动练习</title>
6	<style>
7	
8	body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend
9	,input,textarea,p,blockquote,th,td{
10	margin:0;padding:0
11	}
12	.header{
13	width: 980px;
14	height: 100px;
15	/*background-color: red;*/
16	margin:0 auto;
17	}
18	.header .logo{
19	width: 250px;
20	height: 100px;
21	background-color: pink;
22	float: left;
23	}
24	.header .language{
25	width: 150px;
26	height: 50px;
27	background-color: skyblue;
28	float: right;
29	}
30	.header .nav{
31	width: 650px;
32	height: 50px;
33	background-color: purple;
34	float: right;
35	}
36	.content{
37	width: 980px;

```
36         height: 400px;
37         /*background-color: green;*/
38         margin:0 auto;
39         margin-top: 10px;
40     }
41     .content .aside{
42         width: 320px;
43         height: 400px;
44         background-color: yellow;
45         float: left;
46     }
47     .content .article{
48         width: 650px;
49         height: 400px;
50         /*background-color: pink;*/
51         float: right;
52     }
53     .content .article .articleTop{
54         width: 650px;
55         height: 350px;
56         /*background-color: blue;*/
57     }
58     .content .article .articleTop .articleTopLeft{
59         width: 400px;
60         height: 350px;
61         /*background-color: red;*/
62         float: left;
63     }
64     .content .article .articleTop .articleTopLeft .new1{
65         width: 400px;
66         height: 200px;
67         background-color: deepskyblue;
68     }
69     .content .article .articleTop .articleTopLeft .new2{
70         width: 400px;
71         height: 140px;
72         background-color: tomato;
73         margin-top: 10px;
74     }
75     .content .article .articleTop .articleTopRight{
76         width: 240px;
77         height: 350px;
78         background-color: green;
79         float: right;
80     }
81     .content .article .articleBottom{
82         width: 650px;
83         height: 40px;
84         background-color: darkgoldenrod;
85         margin-top: 10px;
86     }
87     .footer{
88         width: 980px;
89         height: 40px;
90         background-color: tomato;
91         margin:0 auto;
92         margin-top:10px;
93     }
```

```

94     </style>
95 </head>
96 <body>
97 <!--
98 1.企业开发中什么时候使用标准流什么时候使用浮动流?
99 垂直方向使用标准流，水平方向使用浮动流
100
101 2.拿到一个很复杂的界面如何入手?
102 2.1从上至下布局
103 2.2从外向内布局
104 2.3水平方向可以先划分为一左一右再对左边或者右边进行进一步布局
105 -->
106 <div class="header">
107     <div class="logo"></div>
108     <div class="language"></div>
109     <div class="nav"></div>
110 </div>
111 <div class="content">
112     <div class="aside"></div>
113     <div class="article">
114         <div class="articleTop">
115             <div class="articleTopLeft">
116                 <div class="new1"></div>
117                 <div class="new2"></div>
118             </div>
119             <div class="articleTopRight"></div>
120         </div>
121         <div class="articleBottom"></div>
122     </div>
123 </div>
124 <div class="footer"></div>
125 </body>
126 </html>

```

浮动元素的高度

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>64-浮动元素高度问题</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        div{
12            border: 1px solid #000;
13        }
14        p{
15            float: left;
16            width: 50px;
17            height: 50px;
18            background-color: red;
19        }

```

```

20     </style>
21 </head>
22 <body>
23 <!--
24 1. 在标准流中内容的高度可以撑起父元素的高度
25 2. 在浮动流中浮动的元素是不可以撑起父元素的高度的
26 -->
27 <div>
28     <p></p>
29 </div>
30 </body>
31 </html>

```

清除浮动方式一

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>65-清除浮动方式一</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         .box1{
12             height: 20px;
13             background-color: red;
14         }
15         .box2{
16             background-color: green;
17         }
18         .box1 p{
19             width: 100px;
20             background-color: blue;
21         }
22         .box2 p{
23             width: 100px;
24             background-color: yellow;
25         }
26         p{
27             float: left;
28         }
29     </style>
30 </head>
31 <body>
32 <!--
33 1. 清除浮动的第一种方式
34 给前面一个父元素设置高度
35
36 注意点：
37 在企业开发中，我们能不写高度就不写高度，所以这种方式用得很少
38 -->
39 <div class="box1">
40     <p>我是文字1</p>

```

```
41     <p>我是文字1</p>
42     <p>我是文字1</p>
43 </div>
44
45 <div class="box2">
46     <p>我是文字2</p>
47     <p>我是文字2</p>
48     <p>我是文字2</p>
49 </div>
50 </body>
51 </html>
```

清除浮动方式二

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>66-清除浮动方式二</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         body{
12             border: 1px solid #000;
13         }
14         .box1{
15             background-color: red;
16         }
17         .box2{
18             background-color: green;
19             clear: both;
20             margin-top: 28px;
21         }
22         .box1 p{
23             width: 100px;
24             background-color: blue;
25         }
26         .box2 p{
27             width: 100px;
28             background-color: yellow;
29         }
30         p{
31             float: left;
32         }
33         /*
34         .dd{
35             width: 500px;
36             height: 500px;
37             background-color: red;
38             border: 1px solid #000;
39         }
40         .ddd{
41             width: 200px;
```



```

42         height: 200px;
43         background-color: blue;
44         margin-top: 100px;
45     }
46     */
47 </style>
48 </head>
49 <body>
50 <!--
51 1. 清除浮动的第二种方式
52 给后面的盒子添加clear属性
53
54 clear属性取值：
55 none：默认取值，按照浮动元素的排序规则来排序（左浮动找左浮动，右浮动找右浮动）
56 left：不要找前面的左浮动元素
57 right：不要找前面的右浮动元素
58 both：不要找前面的左浮动元素和右浮动元素
59
60 注意点：
61 当我们给某个元素添加clear属性之后，那么这个属性的margin属性就会失效
62 -->
63
64 <div class="box1">
65     <p>我是文字1</p>
66     <p>我是文字1</p>
67     <p>我是文字1</p>
68 </div>
69
70 <div class="box2">
71     <p>我是文字2</p>
72     <p>我是文字2</p>
73     <p>我是文字2</p>
74 </div>
75
76 <!--
77 <div class="dd">
78     <div class="ddd"></div>
79 </div>
80 -->
81 </body>
82 </html>

```

清除浮动方式三

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>67-清除浮动方式三</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        .box1{

```

```

12         background-color: red;
13         /*margin-bottom: 10px;*/
14     }
15     .box2{
16         background-color: green;
17         /*margin-top: 10px;*/
18     }
19     .box1 p{
20         width: 100px;
21         background-color: blue;
22     }
23     .box2 p{
24         width: 100px;
25         background-color: yellow;
26     }
27     p{
28         float: left;
29     }
30     .wall{
31         clear: both;
32     }
33     .h20{
34         height: 20px;
35         background-color: skyblue;
36     }
37     </style>
38 </head>
39 <body>
40 <!--
41 1. 清除浮动的第三种方式
42 隔墙法
43
44 2. 外墙法
45 2.1 在两个盒子中间添加一个额外的块级元素
46 2.2 给这个额外添加的块级元素设置clear: both;属性
47
48 注意点：
49 外墙法它可以第二个盒子使用margin-top属性
50 外墙法不可以第一个盒子使用margin-bottom属性
51
52 3. 内墙法
53 3.1 在第一个盒子中所有子元素最后添加一个额外的块级元素
54 3.2 给这个额外添加的块级元素设置clear: both;属性
55
56 注意点：
57 内墙法它可以第二个盒子使用margin-top属性
58 内墙法它可以第一个盒子使用margin-bottom属性
59
60 4. 外墙法和内墙法区别？
61 外墙法不能撑起第一个盒子的高度，而内墙法可以撑起第一个盒子的高度
62
63 5. 在企业开发中不常用隔墙法来清除浮动
64 -->
65 <div class="box1">
66     <p>我是文字1</p>
67     <p>我是文字1</p>
68     <p>我是文字1</p>
69     <div class="wall h20"></div>

```

```

70 </div>
71
72 <!--<div class="wall h20"></div>-->
73
74 <div class="box2">
75     <p>我是文字2</p>
76     <p>我是文字2</p>
77     <p>我是文字2</p>
78 </div>
79 </body>
80 </html>

```

伪元素选择器

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>68-伪元素选择器</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 200px;
13             height: 200px;
14             background-color: red;
15         }
16         /*
17         p{
18             width: 50px;
19             height: 50px;
20             background-color: pink;
21         }
22         */
23
24         div::before{
25             content: "爱你";
26             width: 50px;
27             height: 50px;
28             background-color: pink;
29             display: block;
30         }
31         div::after{
32             /*指定添加的子元素中存储的内容*/
33             content: "么么哒";
34             /*指定添加的子元素的宽度和高度*/
35             width: 50px;
36             /*height: 50px;*/
37             /*内容是可以超出标签的范围的，所以高度为0依然可以看见内容*/
38             height:0;
39             background-color: pink;
40             /*指定添加的子元素的显示模式*/
41             display: block;

```

```

42      /*隐藏添加的子元素*/
43      visibility: hidden;
44  }
45
46  </style>
47 </head>
48 <body>
49 <!--
50 1. 什么是伪元素选择器?
51 伪元素选择器作用就是给指定标签的内容前面添加一个子元素或者给指定标签的内容后面添加一个子元素
52
53 2. 格式:
54 标签名称::before{
55     属性名称:值;
56 }
57 给指定标签的内容前面添加一个子元素
58
59 标签名称::after{
60     属性名称:值;
61 }
62 给指定标签的内容后面添加一个子元素
63
64 -->
65 <div>
66     <!--<p>爱你</p>-->
67     我是文字
68     <!--<p>么么哒</p>-->
69 </div>
70
71 </body>
72 </html>

```

清除浮动方式四

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>69-清除浮动方式四</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         .box1{
12             background-color: red;
13             /*margin-bottom: 10px;*/
14         }
15         .box2{
16             background-color: green;
17             /*margin-top: 10px;*/
18         }
19         .box1 p{
20             width: 100px;

```

```

21         background-color: blue;
22     }
23     .box2 p{
24         width: 100px;
25         background-color: yellow;
26     }
27     p{
28         float: left;
29     }
30     .box1::after{
31         /*设置添加的子元素的内容为空*/
32         content: "";
33         /*设置添加的子元素为块级元素*/
34         display: block;
35         /*设置添加的子元素的高度为0*/
36         height: 0;
37         /*设置添加的子元素看不见*/
38         visibility: hidden;
39         /*给添加的子元素设置clear: both;*/
40         clear: both;
41     }
42     .box1{
43         /*兼容IE6*/
44         *zoom:1;
45     }
46 </style>
47 </head>
48 <body>
49 <!--
50 1. 清除浮动的第四种方式
51 利用伪元素选择器清除浮动
52 本质上就是内墙法，只不过是直接通过CSS代码添加了内墙，其它特性和内墙法都一样
53
54 注意点：
55 IE6中不支持这种方式，为了兼容IE6必须给前面的盒子添加*zoom:1;属性
56 -->
57 <div class="box1">
58     <p>我是文字1</p>
59     <p>我是文字1</p>
60     <p>我是文字1</p>
61 </div>
62
63 <div class="box2">
64     <p>我是文字2</p>
65     <p>我是文字2</p>
66     <p>我是文字2</p>
67 </div>
68 </body>
69 </html>

```

清除浮动方式五

```

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

```

```

4      <meta charset="UTF-8">
5      <title>70-清除浮动方式五</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         /*
12         div{
13             width: 100px;
14             height: 100px;
15             background-color: red;
16             overflow: hidden;
17         }
18         */
19         /*
20         .box1{
21             background-color: red;
22             overflow: hidden;
23             *zoom:1;
24         }
25         .box2{
26             background-color: green;
27         }
28         .box1 p{
29             width: 100px;
30             background-color: blue;
31         }
32         .box2 p{
33             width: 100px;
34             background-color: yellow;
35         }
36         p{
37             float: left;
38         }
39         */
40         .box1{
41             width: 200px;
42             height: 200px;
43             background-color: red;
44             /*border: 1px solid #000;*/
45             overflow: hidden;
46         }
47         .box2{
48             width: 100px;
49             height: 100px;
50             background-color: blue;
51             margin-top: 20px;
52         }
53     </style>
54 </head>
55 <body>
56 <!--
57 1.overflow: hidden;作用
58 1.1可以将超出标签范围的内容裁剪掉
59 1.2清除浮动
60 1.3可以通过overflow: hidden;让里面的盒子设置margin-top之后，外面的盒子不被顶下来
61 -->

```

```

62
63 <!--
64 <div>我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是文字我是
  文字我是文字我是文字我是文字我是文字我是文字</div>
65 -->
66
67 <!--
68 <div class="box1">
69     <p>我是文字1</p>
70     <p>我是文字1</p>
71     <p>我是文字1</p>
72
73 </div>
74
75 <div class="box2">
76     <p>我是文字2</p>
77     <p>我是文字2</p>
78     <p>我是文字2</p>
79 </div>
80 -->
81 <div class="box1">
82     <div class="box2"></div>
83 </div>
84 </body>
85 </html>

```

相对定位

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>75-定位流</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 100px;
13             height: 100px;
14         }
15         .box1{
16             background-color: red;
17         }
18         .box2{
19             background-color: green;
20             position: relative;
21             top: 20px;
22             left: 20px;
23             /*right: 20px;*/
24             /*margin-bottom: 20px;*/
25             margin-top: 20px;
26         }
27         .box3{

```

```

28         background-color: blue;
29     }
30     span{
31         position: relative;
32         width: 100px;
33         height: 100px;
34         background-color: red;
35     }
36     input{
37         width: 200px;
38         height: 50px;
39     }
40     img{
41         width: 100px;
42         height: 50px;
43         position: relative;
44         top: 20px;
45     }
46     </style>
47 </head>
48 <body>
49 <!--
50 1. 定位流分类
51 1.1 相对定位
52 1.2 绝对定位
53 1.3 固定定位
54 1.4 静态定位
55
56 2. 什么是相对定位?
57 相对定位就是相对于自己以前在标准流中的位置来移动
58
59 3. 相对定位注意点
60 3.1 相对定位是不脱离标准流的，会继续在标准流中占用一份空间
61 3.2 在相对定位中同一个方向上的定位属性只能使用一个
62 3.3 由于相对定位是不脱离标准流的，所以在相对定位中是区分块级元素/行内元素/行内块级元素
63 3.4 由于相对定位是不脱离标准流的，并且相对定位的元素会占用标准流中的位置，所以当给相对定位
    的元素设置margin/padding等属性的时会影响到标准流的布局
64
65 4. 相对定位应用场景
66 4.1 用于对元素进行微调
67 4.2 配合后面学习的绝对定位来使用
68 -->
69
70 <!--
71 <div class="box1"></div>
72 <div class="box2"></div>
73 <div class="box3"></div>
74 -->
75 <span>我是span</span>
76
77
78 <input type="text" name="" id="">
79 
80 </body>
81 </html>

```


绝对定位

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>76-定位流-绝对定位</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 100px;
13             height: 100px;
14         }
15         .box1{
16             background-color: red;
17         }
18         .box2{
19             background-color: green;
20             position: absolute;
21             /*float: left;*/
22             left: 0;
23             /*right: 0;*/
24             /*top: 0;*/
25             bottom: 0;
26         }
27         .box3{
28             background-color: blue;
29         }
30         span{
31             position: absolute;
32             width: 100px;
33             height: 100px;
34             background-color: yellow;
35         }
36     </style>
37 </head>
38 <body>
39 <!--
40 1. 什么是绝对定位?
41 绝对定位就是相对于body来定位
42
43 2. 绝对定位注意点
44 2.1 绝对定位的元素是脱离标准流的
45 2.2 绝对定位的元素是不区分块级元素/行内元素/行内块级元素
46 -->
47 <div class="box1"></div>
48 <div class="box2"></div>
49 <div class="box3"></div>
50 <span>我是span</span>
51 
52 </body>
53 </html>
```

绝对定位的参考点

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>77-绝对定位-参考点</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    .box1{
12      width: 300px;
13      height: 300px;
14      background-color: red;
15      position: relative;
16    }
17    .box2{
18      width: 200px;
19      height: 200px;
20      background-color: green;
21      position: relative;
22    }
23    .box3{
24      width: 100px;
25      height: 100px;
26      background-color: blue;
27      position: absolute;
28      left: 0;
29      bottom: 0;
30    }
31  </style>
32 </head>
33 <body>
34 <!--
35 1. 规律
36 1. 默认情况下所有的绝对定位的元素，无论有没有祖先元素，都会以body作为参考点
37
38 2. 如果一个绝对定位的元素有祖先元素，并且祖先元素也是定位流，那么这个绝对定位的元素就会以定
   位流的那个祖先元素作为参考点
39 2.1 只要是这个绝对定位元素的祖先元素都可以
40 2.2 指的定位流是指绝对定位/相对定位/固定定位
41 2.3 定位流中只有静态定位不行
42
43 3. 如果一个绝对定位的元素有祖先元素，并且祖先元素也是定位流，而且祖先元素中有多个元素都是定
   位流，那么这个绝对定位的元素会以离它最近的那个定位流的祖先元素为参考点
44 -->
45 <div class="box1">
46   <div class="box2">
47     <div class="box3"></div>
48   </div>
49 </div>
50 </body>
51 </html>
```

绝对定位的注意事项

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>78-绝对定位-注意点</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         /*
12         .box1{
13             width: 100px;
14             height: 100px;
15             background-color: red;
16             position: absolute;
17             right: 0;
18             bottom: 0;
19         }
20         .box2{
21             width: 2000px;
22             height: 100px;
23             background-color: green;
24         }
25         .box3{
26             width: 100px;
27             height: 2000px;
28             background-color: blue;
29         }
30         */
31         .box1{
32             width: 300px;
33             height: 300px;
34             background-color: red;
35             border: 10px solid #000;
36             padding: 30px;
37             position: relative;
38         }
39         .box2{
40             width: 100px;
41             height: 100px;
42             background-color: green;
43             position: absolute;
44             left: 0;
45             top: 0;
46         }
47     </style>
48 </head>
49 <body>
50 <!--
51 1. 如果一个绝对定位的元素是以body作为参考点，那么其实是以网页首屏的宽度和高度作为参考点，
   而不是以整个网页的宽度和高度作为参考点
52
```

```

53 2. 一个绝对定位的元素会忽略祖先元素的padding
54 -->
55
56 <!--
57 <div class="box1"></div>
58 <div class="box2"></div>
59 <div class="box3"></div>
60 -->
61
62 <div class="box1">
63     <div class="box2"></div>
64 </div>
65 </body>
66 </html>

```

子绝父相

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>79-绝对定位-子绝父相</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             list-style: none;
13             width: 800px;
14             height: 50px;
15             margin: 0 auto;
16             margin-top: 100px;
17             background-color: red;
18         }
19         ul li{
20             float: left;
21             width: 150px;
22             line-height: 50px;
23             text-align: center;
24             background-color: #ccc;
25         }
26         ul li:nth-of-type(4){
27             background-color: yellow;
28             position: relative;
29         }
30         ul li img{
31             /*
32             相对定位弊端：
33             相对定位不会脱离标准流，会继续在标准流中占用一份空间，所以不利于布局界面
34             */
35             /*
36             position: relative;
37             left: -42px;
38             top: -18px;

```

```

39      */
40      /*
41      绝对定位弊端：
42      默认情况下绝对定位的元素会以body作为参考点，所以会随着浏览器的宽度高度的变化
    而变化
43      */
44      /*
45      position: absolute;
46      left: 526px;
47      top: 90px;
48      */
49
50      /*
51      子绝父相
52      子元素用绝对定位，父元素用相对定位
53      */
54      position: absolute;
55      /*left: 40px;*/
56      left: 50%;
57      margin-left: -12px;
58      top: -10px;
59    }
60  </style>
61 </head>
62 <body>
63 <ul>
64   <li>服装城</li>
65   <li>美妆馆</li>
66   <li>京东超市</li>
67   <li>全球购
68     
69   </li>
70   <li>闪购</li>
71   <li>团购</li>
72   <li>拍卖</li>
73   <li>金融</li>
74 </ul>
75 </body>
76 </html>

```

绝对定位实现水平居中

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>80-绝对定位-水平居中</title>
6    <style>
7      *{
8        margin: 0;
9        padding: 0;
10     }
11     div{
12       width: 300px;
13       /*width: 50%;*/

```

```

14         height: 50px;
15         background-color: red;
16         /*margin: 0 auto;*/
17         position: absolute;
18         left: 50%;
19         margin-left: -150px;
20     }
21     </style>
22 </head>
23 <body>
24 <!--
25 1.如何让绝对定位的元素水平居中
26 只需要设置绝对定位元素的left:50%;
27 然后再设置绝对定位元素的 margin-left: -元素宽度的一半px;
28 -->
29 <div></div>
30 </body>
31 </html>

```

定位练习一

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>81-定位练习1</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        div{
12            width: 300px;
13            height: 300px;
14            border: 2px solid #ccc;
15            margin: 0 auto;
16            margin-top: 100px;
17            position: relative;
18        }
19        div img{
20            width: 300px;
21            height: 200px;
22        }
23        div .hot{
24            width: 45px;
25            height: 44px;
26            background: url("images/tuangou.png") no-repeat 0px -280px;
27            /*display: inline-block;*/
28            position: absolute;
29            left: 0;
30            top: 0;
31        }
32        div .price{
33            width: 134px;
34            height: 42px;

```

```

35         background: url("images/tuangou.png") no-repeat 0px -362px;
36         /*display: inline-block;*/
37         position: absolute;
38         left: -7px;
39         top: 163px;
40     }
41 </style>
42 </head>
43 <body>
44 <div>
45     
46     <span class="hot"></span>
47     <span class="price"></span>
48     <p>【2店通用】Love Taste爱味道牛排生活馆 双人套餐，提供免费WiFi</p>
49 </div>
50 </body>
51 </html>

```

定位练习二

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>82-定位练习2</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 520px;
13             height: 280px;
14             border: 2px solid gold;
15             margin: 0 auto;
16             margin-top: 100px;
17             position: relative;
18         }
19         div span{
20             /*margin-top: 10px;*/
21             /*display: block;*/
22             width: 40px;
23             height: 80px;
24             background-color: rgba(0,0,0,0.3);
25             font-size: 50px;
26             color: white;
27             text-align: center;
28             line-height: 80px;
29         }
30         div .leftArrow{
31             position: absolute;
32             left: 0px;
33             top: 100px;
34         }
35         div .rightArrow{

```

```

36         position: absolute;
37         right: 0px;
38         top: 100px;
39     }
40     ol{
41         list-style: none;
42         width: 200px;
43         height: 40px;
44         background-color: rgba(255,255,255,0.7);
45         position: absolute;
46         right: 10px;
47         bottom: 10px;
48     }
49     ol li{
50         width: 40px;
51         /*height: 40px;*/
52         line-height: 40px;
53         text-align: center;
54         border: 1px solid gold;
55         box-sizing: border-box;
56         float: left;
57     }
58     </style>
59 </head>
60 <body>
61 <div>
62     
63     <span class="leftArrow">&lt;</span>
64     <span class="rightArrow">&gt;</span>
65     <ol>
66         <li>1</li>
67         <li>2</li>
68         <li>3</li>
69         <li>4</li>
70         <li>5</li>
71     </ol>
72 </div>
73 </body>
74 </html>

```

固定定位

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>83-定位流-固定定位</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         /*
12         div{
13             width: 200px;

```



```

14         height: 2000px;
15         border: 1px solid #000;
16         background-image: url(images/girl.jpg);
17         background-repeat: no-repeat;
18         background-attachment: fixed;
19     }
20     */
21     div{
22         width: 100px;
23         height: 100px;
24     }
25     .box1{
26         background-color: red;
27     }
28     .box2{
29         background-color: green;
30         position: fixed;
31     }
32     .box3{
33         background-color: blue;
34     }
35     /*
36     span{
37         width: 100px;
38         height: 100px;
39         background-color: yellow;
40         position: fixed;
41     }
42     */
43     .box4{
44         height: 2000px;
45         background-color: yellow;
46     }
47     </style>
48 </head>
49 <body>
50 <!--<div></div>-->
51
52 <!--
53 1. 什么是固定定位?
54 固定定位和前面学习的背景关联方式很像，背景定位可以让背景图片不随着滚动条的滚动而滚动，而固定定位可以让某个盒子不随着滚动条的滚动而滚动
55
56 注意点：
57 1. 固定定位的元素是脱离标准流的，不会占用标准流中的空间
58 2. 固定定位和绝对定位一样不区分行内/块级/行内块级
59 3. IE6不支持固定定位
60 -->
61 <div class="box1"></div>
62 <div class="box2"></div>
63 <div class="box3"></div>
64 <div class="box4"></div>
65 <!--<span>我是span</span>-->
66 </body>
67 </html>

```

定位流的z-index

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>85-定位流-z-index属性</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         /*
12         div{
13             width: 100px;
14             height: 100px;
15         }
16         .box1{
17             background-color: red;
18             position: relative;
19             left: 0;
20             top: 0;
21         }
22         .box2{
23             background-color: green;
24             position: absolute;
25             left: 50px;
26             top: 50px;
27         }
28         .box3{
29             background-color: blue;
30             position: fixed;
31             left: 100px;
32             top: 100px;
33         }
34         */
35         .father1{
36             width: 200px;
37             height: 200px;
38             background-color: red;
39             position: relative;
40             z-index: 2;
41         }
42         .father2{
43             width: 200px;
44             height: 200px;
45             background-color: green;
46             position: relative;
47             z-index: 3;
48         }
49         .son1{
50             width: 100px;
51             height: 100px;
52             background-color: blue;
53             position: absolute;
54             left: 200px;
55             top: 200px;
```

```

56         z-index: 1;
57     }
58     .son2{
59         width: 100px;
60         height: 100px;
61         background-color: yellow;
62         position: absolute;
63         left: 250px;
64         top: 50px;
65         z-index: 2;
66     }
67 </style>
68 </head>
69 <body>
70 <!--
71 1.什么是z-index属性?
72 默认情况下所有的元素都有一个默认的z-index属性，取值是0.
73 z-index属性的作用是专门用于控制定位流元素的覆盖关系的
74
75 1.默认情况下定位流的元素会盖住标准流的元素
76 2.默认情况下定位流的元素后面编写的会盖住前面编写的
77 3.如果定位流的元素设置了z-index属性，那么谁的z-index属性比较大，谁就会显示在上面
78
79 注意点：
80 1.从父现象
81 1.1如果两个元素的父元素都没有设置z-index属性，那么谁的z-index属性比较大谁就显示在上面
82 1.2如果两个元素的父元素设置了z-index属性，那么子元素的z-index属性就会失效，也就是说谁的
    父元素的z-index属性比较大谁就会显示在上面
83 -->
84 <!--
85 <div class="box1"></div>
86 <div class="box2"></div>
87 <div class="box3"></div>
88 -->
89 <div class="father1">
90     <div class="son1"></div>
91 </div>
92 <div class="father2">
93     <div class="son2"></div>
94 </div>
95 </body>
96 </html>

```

a标签的伪类选择

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>86-a标签的伪类选择器</title>
6     <style>
7         /*
8         a:link{
9             color: tomato;
10        }

```

```

11     a:visited{
12         color: green;
13     }
14     a:hover{
15         color: orange;
16     }
17     a:active{
18         color: pink;
19     }
20     */
21     a{
22         // 简写格式
23         color: green;
24     }
25     /*a:link{*/
26         /*color: green;*/
27     /*}*/
28     /*a:visited{*/
29         /*color: green;*/
30     /*}*/
31     a:hover{
32         color: orange;
33     }
34     a:active{
35         color: pink;
36     }
37 </style>
38 </head>
39 <body>
40 <!--
41 1. 通过我们的观察发现a标签存在一定的状态
42 1.1 默认状态，从未被访问过
43 1.2 被访问过的状态
44 1.3 鼠标长按状态
45 1.4 鼠标悬停在a标签上状态
46
47 2. 什么是a标签的伪类选择器？
48 a标签的伪类选择器是专门用来修改a标签不同状态的样式的
49
50 3. 格式
51 :link 修改从未被访问过状态下的样式
52 :visited 修改被访问过的状态下的样式
53 :hover 修改鼠标悬停在a标签上状态下的样式
54 :active 修改鼠标长按状态下的样式
55
56 4. 注意点
57 4.1 a标签的伪类选择器可以单独出现也可以一起出现
58 4.2 a标签的伪类选择器如果一起出现，那么有严格的顺序要求
59 编写的顺序必须要遵守爱恨原则 love hate
60 4.3 如果默认状态的样式和被访问过状态的样式一样，那么可以缩写
61 -->
62 <a href="http://www.taobao.com">taobao</a>
63 <a href="http://www.jd.com">jd</a>
64 </body>
65 </html>

```

过渡模块

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>88-过渡模块</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    div{
12      width: 100px;
13      height: 50px;
14      background-color: red;
15      /*告诉系统哪个属性需要执行过渡效果*/
16      transition-property: width, background-color;
17      /*告诉系统过渡效果持续的时长*/
18      transition-duration: 5s, 5s;
19
20      /*transition-property: background-color;*/
21      /*transition-duration: 5s;*/
22    }
23    /*:hover这个伪类选择器除了可以用在a标签上，还可以用在其它的任何标签上*/
24    div:hover{
25      width: 300px;
26      background-color: blue;
27    }
28  </style>
29 </head>
30 <body>
31 <!--
32 1, 过渡三要素
33 1.1 必须要有属性发生变化
34 1.2 必须告诉系统哪个属性需要执行过渡效果
35 1.3 必须告诉系统过渡效果持续时长
36
37 2. 注意点
38 当多个属性需要同时执行过渡效果时用逗号隔开即可
39 transition-property: width, background-color;
40 transition-duration: 5s, 5s;
41 -->
42 <div></div>
43 </body>
44 </html>
```

过渡模块的其他属性

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>89-过渡模块-其它属性</title>
```

```
6 <style>
7   *{
8     margin: 0;
9     padding: 0;
10  }
11  div {
12    width: 100px;
13    height: 50px;
14    background-color: red;
15    transition-property: width;
16    transition-duration: 5s;
17    /*告诉系统延迟多少秒之后才开始过渡动画*/
18    /*transition-delay: 2s;*/
19  }
20  div:hover{
21    width: 300px;
22  }
23  ul{
24    width: 800px;
25    height: 500px;
26    margin: 0 auto;
27    background-color: pink;
28    border: 1px solid #000;
29  }
30  ul li{
31    list-style: none;
32    width: 100px;
33    height: 50px;
34    margin-top: 50px;
35    background-color: blue;
36    transition-property: margin-left;
37    transition-duration: 10s;
38  }
39  ul:hover li{
40    margin-left: 700px;
41  }
42  ul li:nth-child(1){
43    /*告诉系统过渡动画的运动的的速度*/
44    transition-timing-function: linear;
45  }
46  ul li:nth-child(2){
47    transition-timing-function: ease;
48  }
49  ul li:nth-child(3){
50    transition-timing-function: ease-in;
51  }
52  ul li:nth-child(4){
53    transition-timing-function: ease-out;
54  }
55  ul li:nth-child(5){
56    transition-timing-function: ease-in-out;
57  }
58 </style>
59 </head>
60 <body>
61 <!--<div></div>-->
62 <ul>
63   <li>linear</li>
```

```
64     </li>ease</li>
65     </li>ease-in</li>
66     </li>ease-out</li>
67     </li>ease-in-out</li>
68 </ul>
69 </body>
70 </html>
```

过渡模块的连写

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>90-过渡模块-连写</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div {
12             width: 100px;
13             height: 50px;
14             background-color: red;
15             /*transition-property: width;*/
16             /*transition-duration: 5s;*/
17             /*transition: width 5s linear 0s,background-color 5s linear
0s;*/
18             /*transition: background-color 5s linear 0s;*/
19             /*transition: width 5s,background-color 5s,height 5s;*/
20             transition: all 5s;
21         }
22         div:hover{
23             width: 300px;
24             height: 300px;
25             background-color: blue;
26         }
27     </style>
28 </head>
29 <body>
30 <!--
31 1.过渡连写格式
32 transition: 过渡属性 过渡时长 运动速度 延迟时间;
33
34 2.过渡连写注意点
35 2.1和分开写一样，如果想给多个属性添加过渡效果也是用逗号隔开即可
36 2.2连写的时可以省略后面的两个参数，因为只要编写了前面的两个参数就已经满足了过渡的三要素
37 2.3如果多个属性运动的速度/延迟的时间/持续时间都一样，那么可以简写为
38 transition:all 0s;
39 -->
40 <div></div>
41 </body>
42 </html>
```

过渡实现弹性效果

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>91-过渡模块-弹性效果</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    div{
12      height: 100px;
13      background-color: red;
14      margin-top: 100px;
15      text-align: center;
16      line-height: 100px;
17    }
18    div span{
19      font-size: 80px;
20      /*transition-property: margin;*/
21      /*transition-duration: 3s;*/
22      transition: margin 3s;
23    }
24    div:hover span{
25      margin: 0 20px;
26    }
27  </style>
28 </head>
29 <body>
30 <!--
31 1. 编写过渡套路
32 1.1 不要管过渡，先编写基本界面
33 1.2 修改我们认为需要修改的属性
34 1.3 再回过头去给被修改属性的那个元素添加过渡即可
35 -->
36 <div>
37   <span>江</span>
38   <span>哥</span>
39   <span>带</span>
40   <span>你</span>
41   <span>狂</span>
42   <span>虐</span>
43   <span>H</span>
44   <span>5</span>
45 </div>
46 </body>
47 </html>
```

过渡实现手风琴效果


```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>92-过渡模块-手风琴效果</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 960px;
13             height: 300px;
14             margin: 100px auto;
15             border: 1px solid #000;
16             overflow: hidden;
17         }
18         ul li{
19             list-style: none;
20             width: 160px;
21             height: 300px;
22             background-color: red;
23             float: left;
24             /*border: 1px solid #000;*/
25             /*box-sizing: border-box;*/
26             /*transition-property: width;*/
27             /*transition-duration: 0.5s;*/
28             transition: width 0.5s;
29         }
30         ul: hover li{
31             width: 100px;
32         }
33         ul li: hover{
34             width: 460px;
35         }
36     </style>
37 </head>
38 <body>
39 <ul>
40     <li></li>
41     <li></li>
42     <li></li>
43     <li></li>
44     <li></li>
45     <li></li>
46 </ul>
47 </body>
48 </html>

```

2D转换模块

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">

```

```
5 <title>93-2D转换模块</title>
6 <style>
7     *{
8         margin: 0;
9         padding: 0;
10    }
11    ul{
12        width: 800px;
13        height: 500px;
14        border: 1px solid #000;
15        margin: 0 auto;
16    }
17    ul li{
18        list-style: none;
19        width: 100px;
20        height: 50px;
21        background-color: red;
22        margin: 0 auto;
23        margin-top: 50px;
24        text-align: center;
25        line-height: 50px;
26    }
27    ul li:nth-child(2){
28        /*其中deg是单位，代表多少度*/
29        transform: rotate(45deg);
30    }
31    ul li:nth-child(3){
32        /*
33        第一个参数:水平方向
34        第二个参数:垂直方向
35        */
36        transform: translate(100px, 0px);
37    }
38    ul li:nth-child(4){
39        /*
40        第一个参数:水平方向
41        第二个参数:垂直方向
42        注意点:
43        如果取值是1，代表不变
44        如果取值大于1，代表需要放大
45        如果取值小于1，代表需要缩小
46        如果水平和垂直缩放都一样，那么可以简写为一个参数
47        */
48        /*transform: scale(0.5, 0.5);*/
49        transform: scale(1.5);
50    }
51    ul li:nth-child(5){
52        /*
53        注意点:
54        1.如果需要进行多个转换，那么用空格隔开
55        2.2D的转换模块会修改元素的坐标系，所以旋转之后再平移就不是水平平移的
56        */
57        transform: rotate(45deg) translate(100px, 0px) scale(1.5, 1.5);
58        /*transform: translate(100px, 0px);*/
59    }
60 </style>
61 </head>
62 <body>
```

```

63 <ul>
64     <li>正常的</li>
65     <li>旋转的</li>
66     <li>平移的</li>
67     <li>缩放的</li>
68     <li>综合的</li>
69 </ul>
70 </body>
71 </html>

```

形变中心

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>94-2D转换模块-形变中心点</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 200px;
13             height: 200px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             position: relative;
17         }
18         ul li{
19             list-style: none;
20             width: 200px;
21             height: 200px;
22             position: absolute;
23             left: 0;
24             top: 0;
25             /*
26             第一个参数:水平方向
27             第二个参数:垂直方向
28             注意点
29             取值有三种形式
30             具体像素
31             百分比
32             特殊关键字
33             */
34             /*transform-origin: 200px 0px;*/
35             /*transform-origin: 50% 50%;*/
36             /*transform-origin: 0% 0%;*/
37             /*transform-origin: center center;*/
38             transform-origin: left top;
39         }
40         ul li:nth-child(1){
41             /*
42             默认情况下所有的元素都是以自己的中心点作为参考来旋转的，我们可以通过形变中心
             点属性来修改它的参考点

```

```

43         */
44         background-color: red;
45         transform: rotate(30deg);
46     }
47     ul li:nth-child(2){
48         background-color: green;
49         transform: rotate(50deg);
50     }
51     ul li:nth-child(3){
52         background-color: blue;
53         transform: rotate(70deg);
54     }
55     </style>
56 </head>
57 <body>
58 <ul>
59     <li></li>
60     <li></li>
61     <li></li>
62 </ul>
63 </body>
64 </html>

```

旋转轴向

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>95-2D转换模块-旋转轴向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 800px;
13             height: 500px;
14             margin: 0 auto;
15         }
16         ul li{
17             list-style: none;
18             width: 200px;
19             height: 200px;
20             margin: 0 auto;
21             margin-top: 50px;
22             border: 1px solid #000;
23             /*
24             1.什么是透视
25             近大远小
26             2.注意点
27             一定要注意，透视属性必须添加到需要呈现近大远小效果的元素的父元素上面
28             */
29             perspective: 500px;
30

```

```

31     }
32     ul li img{
33         width: 200px;
34         height: 200px;
35         /*perspective: 500px;*/
36     }
37     ul li:nth-child(1){
38         /*默认情况下所有元素都是围绕Z轴进行旋转*/
39         transform: rotateZ(45deg);
40     }
41     ul li:nth-child(2) img{
42         transform: rotateX(45deg);
43     }
44     ul li:nth-child(3) img{
45         /*
46         总结：
47         想围绕哪个轴旋转，那么只需要在rotate后面加上哪个轴即可
48         */
49         transform: rotateY(45deg);
50     }
51     </style>
52 </head>
53 <body>
54 <ul>
55     <li></li>
56     <li></li>
57     <li></li>
58 </ul>
59 </body>
60 </html>

```

转换模块的练习

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>96-2D转换模块-练习</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 310px;
13             height: 438px;
14             border: 1px solid #000;
15             background-color: skyblue;
16             margin: 100px auto;
17             perspective: 500px;
18         }
19         div img{
20             transform-origin: center bottom;
21             transition: transform 1s;
22         }

```

```

23         div:hover img{
24             transform: rotateX(80deg);
25         }
26     </style>
27 </head>
28 <body>
29 <div>
30     
31 </div>
32 </body>
33 </html>

```

相片墙案例

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>97-2D转换模块-相片墙</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             height: 400px;
13             border: 1px solid #000;
14             background-color: skyblue;
15             margin-top: 100px;
16             text-align: center;
17         }
18         ul li{
19             list-style: none;
20             width: 150px;
21             height: 200px;
22             background-color: red;
23             display: inline-block;
24             margin-top: 100px;
25             transition: all 1s;
26             position: relative;
27             box-shadow: 0 0 10px;
28         }
29         ul li:nth-child(1){
30             transform: rotate(30deg);
31         }
32         ul li:nth-child(2){
33             transform: rotate(-40deg);
34         }
35         ul li:nth-child(3){
36             transform: rotate(10deg);
37         }
38         ul li:nth-child(4){
39             transform: rotate(45deg);
40         }
41         ul li img{

```

```

42         width: 150px;
43         height: 200px;
44         border: 5px solid #fff;
45         box-sizing: border-box;
46     }
47     ul li:hover{
48         /*transform: rotate(0deg);*/
49         /*transform: none;*/
50         transform: scale(1.5);
51         z-index: 998;
52     }
53     </style>
54 </head>
55 <body>
56 <ul>
57     <li></li>
58     <li></li>
59     <li></li>
60     <li></li>
61 </ul>
62 </body>
63 </html>

```

盒子阴影和文字阴影

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>98-盒子阴影和文字阴影</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         .box1{
12             width: 200px;
13             height: 200px;
14             background-color: red;
15             margin: 100px auto;
16             text-align: center;
17             line-height: 200px;
18             /*box-shadow: 10px 10px 10px 10px skyblue;*/
19             /*box-shadow: 10px 10px 10px 10px skyblue inset;*/
20             box-shadow: 10px 10px 10px;
21             color: yellow;
22         }
23         .box2{
24             width: 200px;
25             height: 200px;
26             margin: 0 auto;
27             background-color: pink;
28             text-align: center;
29             line-height: 200px;
30             font-size: 40px;

```

```

31         /*text-shadow: 10px 10px 10px black;*/
32         text-shadow: 10px 10px 10px;
33         color: purple;
34     }
35 </style>
36 </head>
37 <body>
38 <!--
39 1.如何给盒子添加阴影
40 box-shadow: 水平偏移 垂直偏移 模糊度 阴影扩展 阴影颜色 内外阴影;
41
42 2.注意点
43 2.1盒子的阴影分为内外阴影，默认情况下就是外阴影
44 2.2快速添加阴影只需要编写三个参数即可
45 box-shadow: 水平偏移 垂直偏移 模糊度;
46 默认情况下阴影的颜色和盒子内容的颜色一致
47
48 3.如何给文字添加阴影
49 text-shadow: 水平偏移 垂直偏移 模糊度 阴影颜色 ;
50 -->
51 <div class="box1">我是盒子</div>
52 <div class="box2">我是盒子</div>
53 </body>
54 </html>

```

翻转菜单

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>99-翻转菜单-综合练习</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        .nav{
12            width: 400px;
13            height: 40px;
14            margin: 0 auto;
15            margin-top: 100px;
16            background-color: black;
17        }
18        .nav>li{
19            list-style: none;
20            float: left;
21            width: 120px;
22            height: 40px;
23            background-color: green;
24            margin-left: 10px;
25            text-align: center;
26            line-height: 40px;
27            position: relative;
28        }

```



```
29     .sub{
30         /*不显示元素*/
31         /*display: none;*/
32         width: 120px;
33         position: absolute;
34         left: 0;
35         top: 40px;
36     }
37     .sub li{
38         list-style: none;
39         background-color: deeppink;
40         transform: rotateY(180deg);
41         transition: all 1s;
42         opacity: 0;
43     }
44     /*
45     .nav>li:hover .sub{
46         display: block;
47     }
48     */
49     .nav>li:hover .sub li{
50         transform: none;
51         opacity: 1;
52     }
53     .nav>li:hover .sub li:nth-child(1){
54         transition-delay: 0ms;
55     }
56     .nav>li:hover .sub li:nth-child(2){
57         transition-delay: 200ms;
58     }
59     .nav>li:hover .sub li:nth-child(3){
60         transition-delay: 400ms;
61     }
62     .nav>li:hover .sub li:nth-child(4){
63         transition-delay: 600ms;
64     }
65     .nav>li:hover .sub li:nth-child(5){
66         transition-delay: 800ms;
67     }
68
69     .nav>li .sub li:nth-child(5){
70         transition-delay: 0ms;
71     }
72     .nav>li .sub li:nth-child(4){
73         transition-delay: 200ms;
74     }
75     .nav>li .sub li:nth-child(3){
76         transition-delay: 400ms;
77     }
78     .nav>li .sub li:nth-child(2){
79         transition-delay: 600ms;
80     }
81     .nav>li .sub li:nth-child(1){
82         transition-delay: 800ms;
83     }
84     div{
85         width: 400px;
86         height: 300px;
```

[illegible]

动画模块

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>100-动画模块</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        div{
12            width: 100px;
```

```

13         height: 50px;
14         background-color: red;
15         /*transition-property: margin-left;*/
16         /*transition-duration: 3s;*/
17
18         /*1.告诉系统需要执行哪个动画*/
19         animation-name: lnj;
20         /*3.告诉系统动画持续的时长*/
21         animation-duration: 3s;
22     }
23     /*2.告诉系统我们需要自己创建一个名称叫做lnj的动画*/
24     @keyframes lnj {
25         from{
26             margin-left: 0;
27         }
28         to{
29             margin-left: 500px;
30         }
31     }
32
33     /*div:hover{*/
34         /*margin-left: 500px;*/
35     /*}*/
36 </style>
37 </head>
38 <body>
39 <!--
40 1.过渡和动画之间的异同
41 1.1不同点
42 过渡必须人为的触发才会执行动画
43 动画不需要人为的触发就可以执行动画
44
45 1.2相同点
46 过渡和动画都是用来给元素添加动画的
47 过渡和动画都是系统新增的一些属性
48 过渡和动画都需要满足三要素才会有动画效果
49 -->
50 <div></div>
51 </body>
52 </html>

```

动画模块的其他属性

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>101-动画模块-其它属性上</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        div {
12            width: 100px;

```

```

13     height: 50px;
14     background-color: red;
15     animation-name: sport;
16     animation-duration: 2s;
17     /*告诉系统多少秒之后开始执行动画*/
18     /*animation-delay: 2s;*/
19     /*告诉系统动画执行的速度*/
20     animation-timing-function: linear;
21     /*告诉系统动画需要执行几次*/
22     animation-iteration-count: 3;
23     /*告诉系统是否需要执行往返动画
24     取值:
25     normal, 默认的取值, 执行完一次之后回到起点继续执行下一次
26     alternate, 往返动画, 执行完一次之后往回执行下一次
27     */
28     animation-direction: alternate;
29 }
30 @keyframes sport {
31     from{
32         margin-left: 0;
33     }
34     to{
35         margin-left: 500px;
36     }
37 }
38 div:hover{
39     /*
40     告诉系统当前动画是否需要暂停
41     取值:
42     running: 执行动画
43     paused: 暂停动画
44     */
45     animation-play-state: paused;
46 }
47 </style>
48 </head>
49 <body>
50 <div class="box1"></div>
51 </body>
52 </html>

```

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>102-动画模块-其它属性下</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        .box1 {
12            width: 100px;
13            height: 50px;
14            background-color: red;
15            position: absolute;
16            left: 0;

```

```

17         top: 0;
18         animation-name: sport;
19         animation-duration: 5s;
20     }
21     @keyframes sport {
22         0%{
23             left: 0;
24             top: 0;
25         }
26         25%{
27             left: 300px;
28             top: 0;
29         }
30         50%{
31             left: 300px;
32             top: 300px;
33         }
34         75%{
35             left: 0;
36             top: 300px;
37         }
38         100%{
39             left: 0;
40             top: 0;
41         }
42     }
43
44     .box2{
45         width: 200px;
46         height: 200px;
47         background-color: blue;
48         margin: 100px auto;
49         animation-name: myRotate;
50         animation-duration: 5s;
51         animation-delay: 2s;
52         /*
53         通过我们的观察，动画是有一定的状态的
54         1.等待状态
55         2.执行状态
56         3.结束状态
57         */
58         /*
59         animation-fill-mode作用：
60         指定动画等待状态和结束状态的样式
61         取值：
62         none：不做任何改变
63         forwards：让元素结束状态保持动画最后一帧的样式
64         backwards：让元素等待状态的时候显示动画第一帧的样式
65         both：让元素等待状态显示动画第一帧的样式，让元素结束状态保持动画最后一帧的
66         样式
67         */
68         /*animation-fill-mode: backwards;*/
69         /*animation-fill-mode: forwards;*/
70         animation-fill-mode: both;
71     }
72     @keyframes myRotate {
73         0%{
74             transform: rotate(10deg);

```

```

74         }
75         50%{
76             transform: rotate(50deg);
77         }
78         100%{
79             transform: rotate(70deg);
80         }
81     }
82 </style>
83 </head>
84 <body>
85 <div class="box1"></div>
86 <div class="box2"></div>
87 </body>
88 </html>

```

动画简写

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>103-动画模块-连写</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div {
12             width: 100px;
13             height: 50px;
14             background-color: red;
15             /*animation: move 3s linear 2s 1 normal;*/
16             animation: move 3s;
17         }
18         @keyframes move {
19             from{
20                 margin-left: 0;
21             }
22             to{
23                 margin-left: 500px;
24             }
25         }
26     </style>
27 </head>
28 <body>
29 <!--
30 1. 动画模块连写格式
31 animation:动画名称 动画时长 动画运动速度 延迟时间 执行次数 往返动画;
32
33 2. 动画模块连写格式的简写
34 animation:动画名称 动画时长;
35 -->
36 <div></div>
37 </body>

```

云层效果

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>104-动画模块-云层效果</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             height: 400px;
13             background-color: skyblue;
14             margin-top: 100px;
15             animation: change 5s linear 0s infinite alternate;
16             position: relative;
17             overflow: hidden;
18         }
19         ul li{
20             list-style: none;
21             width: 400%;
22             height: 100%;
23             position: absolute;
24             left: 0;
25             top: 0;
26         }
27         ul li:nth-child(1){
28             background-image: url("images/cloud_one.png");
29             animation: one 30s linear 0s infinite alternate;
30         }
31         ul li:nth-child(2){
32             background-image: url("images/cloud_two.png");
33             animation: two 30s linear 0s infinite alternate;
34         }
35         ul li:nth-child(3){
36             background-image: url("images/cloud_three.png");
37             animation: three 30s linear 0s infinite alternate;
38         }
39         @keyframes change {
40             from{
41                 background-color: skyblue;
42             }
43             to{
44                 background-color: black;
45             }
46         }
47         @keyframes one {
48             from{
49                 margin-left: 0;
50             }
51             to{
```

```

52         margin-left: -100%;
53     }
54 }
55 @keyframes two {
56     from{
57         margin-left: 0;
58     }
59     to{
60         margin-left: -200%;
61     }
62 }
63 @keyframes three {
64     from{
65         margin-left: 0;
66     }
67     to{
68         margin-left: -300%;
69     }
70 }
71 </style>
72 </head>
73 <body>
74 <ul>
75     <li></li>
76     <li></li>
77     <li></li>
78 </ul>
79 </body>
80 </html>

```

无线滚动

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>105-动画模块-无限滚动</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 600px;
13             height: 188px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             overflow: hidden;
17         }
18
19         ul{
20             width: 2000px;
21             height: 188px;
22             background-color: black;
23             animation: move 10s linear 0s infinite normal;

```



```

24     }
25     ul li{
26         float: left;
27         list-style: none;
28         width: 300px;
29         height: 188px;
30         background-color: red;
31         border: 1px solid #000;
32         box-sizing: border-box;
33     }
34     ul:hover{
35         /*动画添加给谁，就让谁停止*/
36         animation-play-state: paused;
37     }
38     ul:hover li{
39         opacity: 0.5;
40     }
41     ul li:hover{
42         opacity: 1;
43     }
44     @keyframes move {
45         from{
46             margin-left: 0;
47         }
48         to{
49             margin-left: -1200px;
50         }
51     }
52 </style>
53 </head>
54 <body>
55 <div>
56     <ul>
57         <li></li>
58         <li></li>
59         <li></li>
60         <li></li>
61         <li></li>
62         <li></li>
63     </ul>
64 </div>
65 </body>
66 </html>

```

3D模块

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>106-3D转换模块</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;

```

```

10     }
11     .father{
12         width: 200px;
13         height: 200px;
14         background-color: red;
15         border: 1px solid #000;
16         margin: 100px auto;
17         perspective: 500px;
18         /*
19         1. 什么是2D和3D
20         2D就是一个平面，只有宽度和高度，没有厚度
21         3D就是一个立体，有宽度和高度，还有厚度
22         默认情况下所有的元素都是呈2D展现的
23         2. 如何让某个元素呈3D展现
24         和透视一样，想看到某个元素的3d效果，只需要给他的父元素添加一个transform-
style属性，然后设置为preserve-3d即可
25         */
26         transform-style: preserve-3d;
27         transform: rotateY(0deg);
28     }
29     .son{
30         width: 100px;
31         height: 100px;
32         background-color: blue;
33         border: 1px solid #000;
34         margin: 0 auto;
35         margin-top: 50px;
36         transform: rotateY(45deg);
37     }
38     </style>
39 </head>
40 <body>
41 <div class="father">
42     <div class="son"></div>
43 </div>
44 </body>
45 </html>

```

3D-正方体

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>108-3D转换模块-正方体终极</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        ul{
12            width: 200px;
13            height: 200px;
14            border: 1px solid #000;
15            box-sizing: border-box;

```

```

16         margin: 100px auto;
17         position: relative;
18         transform: rotateY(0deg) rotateX(0deg);
19         transform-style: preserve-3d;
20     }
21     ul li{
22         list-style: none;
23         width: 200px;
24         height: 200px;
25         font-size: 60px;
26         text-align: center;
27         line-height: 200px;
28         position: absolute;
29         left: 0;
30         top: 0;
31     }
32     ul li:nth-child(1){
33         background-color: red;
34         transform: rotateX(90deg) translateZ(100px);
35     }
36     ul li:nth-child(2){
37         background-color: green;
38         transform: rotateX(180deg) translateZ(100px);
39     }
40     ul li:nth-child(3){
41         background-color: blue;
42         transform: rotateX(270deg) translateZ(100px);
43     }
44     ul li:nth-child(4){
45         background-color: yellow;
46         transform: rotateX(360deg) translateZ(100px);
47     }
48     ul li:nth-child(5){
49         background-color: purple;
50         transform: translateX(-100px) rotateY(90deg);
51     }
52     ul li:nth-child(6){
53         background-color: pink;
54         transform: translateX(100px) rotateY(90deg);
55     }
56
57     </style>
58 </head>
59 <body>
60 <ul>
61     <li>1</li>
62     <li>2</li>
63     <li>3</li>
64     <li>4</li>
65     <li>5</li>
66     <li>6</li>
67 </ul>
68 </body>
69 </html>

```

长方体

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>109-3D转换模块-长方体</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    /*
12    .father{
13      width: 200px;
14      height: 200px;
15      border: 1px solid #000;
16      margin: 100px auto;
17    }
18    .son{
19      width: 200px;
20      height: 200px;
21      background-color: rgba(255,0,0,0.3);
22      transform: scale(1.5, 1);
23    }
24    */
25    ul{
26      width: 200px;
27      height: 200px;
28      border: 1px solid #000;
29      box-sizing: border-box;
30      margin: 100px auto;
31      position: relative;
32      transform: rotateY(0deg) rotateX(0deg);
33      transform-style: preserve-3d;
34    }
35    ul li{
36      list-style: none;
37      width: 200px;
38      height: 200px;
39      font-size: 60px;
40      text-align: center;
41      line-height: 200px;
42      position: absolute;
43      left: 0;
44      top: 0;
45    }
46    ul li:nth-child(1){
47      background-color: red;
48      transform: rotateX(90deg) translateZ(100px) scale(2, 1);
49    }
50    ul li:nth-child(2){
51      background-color: green;
52      transform: rotateX(180deg) translateZ(100px) scale(2, 1);
53    }
54    ul li:nth-child(3){
55      background-color: blue;
```

```

56         transform: rotateX(270deg) translateZ(100px) scale(2, 1);
57     }
58     ul li:nth-child(4){
59         background-color: yellow;
60         transform: rotateX(360deg) translateZ(100px) scale(2, 1);
61     }
62     ul li:nth-child(5){
63         background-color: purple;
64         transform: translateX(-200px) rotateY(90deg);
65     }
66     ul li:nth-child(6){
67         background-color: pink;
68         transform: translateX(200px) rotateY(90deg);
69     }
70     </style>
71 </head>
72 <body>
73 <!--
74 <div class="father">
75     <div class="son"></div>
76 </div>
77 -->
78 <ul>
79     <li>1</li>
80     <li>2</li>
81     <li>3</li>
82     <li>4</li>
83     <li>5</li>
84     <li>6</li>
85 </ul>
86 </body>
87 </html>

```

3D模块的练习

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>110-3D转换模块-练习</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        body{
12            /*想看到整个立方的近大远小效果，就给ul的父元素添加透视*/
13            perspective: 500px;
14        }
15        ul{
16            width: 200px;
17            height: 200px;
18            box-sizing: border-box;
19            margin: 100px auto;
20            position: relative;

```

```

21         transform: rotateY(0deg) rotateX(0deg);
22         transform-style: preserve-3d;
23         animation: sport 5s linear 0s infinite normal;
24     }
25     ul li{
26         list-style: none;
27         width: 200px;
28         height: 200px;
29         font-size: 60px;
30         text-align: center;
31         line-height: 200px;
32         position: absolute;
33         left: 0;
34         top: 0;
35     }
36     ul li:nth-child(1){
37         background-color: red;
38         transform: rotateX(90deg) translateZ(100px) scale(2, 1);
39     }
40     ul li:nth-child(2){
41         background-color: green;
42         transform: rotateX(180deg) translateZ(100px) scale(2, 1);
43     }
44     ul li:nth-child(3){
45         background-color: blue;
46         transform: rotateX(270deg) translateZ(100px) scale(2, 1);
47     }
48     ul li:nth-child(4){
49         background-color: yellow;
50         transform: rotateX(360deg) translateZ(100px) scale(2, 1);
51     }
52     ul li:nth-child(5){
53         background-color: purple;
54         transform: translateX(-200px) rotateY(90deg);
55     }
56     ul li:nth-child(6){
57         background-color: pink;
58         transform: translateX(200px) rotateY(90deg);
59     }
60     ul li img{
61         /*
62         注意点：
63         只要父元素被拉伸了，子元素也会被拉伸
64         */
65         width: 200px;
66         height: 200px;
67     }
68     @keyframes sport {
69         from{
70             transform: rotateX(0deg);
71         }
72         to{
73             transform: rotateX(360deg);
74         }
75     }
76 </style>
77 </head>
78 <body>

```

```

79 <ul>
80   <li></li>
81   <li></li>
82   <li></li>
83   <li></li>
84   <li></li>
85   <li></li>
86 </ul>
87 </body>
88 </html>

```

3D播放器的效果

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>111-3D播放器下</title>
6    <style>
7      *{
8        margin: 0;
9        padding: 0;
10     }
11     body{
12       background: url("images/jacky/bg.jpg") no-repeat;
13       background-size:cover;
14       overflow: hidden;
15     }
16     ul{
17       width: 200px;
18       height: 200px;
19       /*background-color: red;*/
20       position: absolute;
21       bottom: 100px;
22       left: 50%;
23       margin-left:-100px;
24       transform-style: preserve-3d;
25       /*transform: rotateX(-10deg);*/
26       animation: sport 6s linear 0s infinite normal;
27     }
28     ul li{
29       list-style: none;
30       width: 200px;
31       height: 200px;
32       font-size: 60px;
33       text-align: center;
34       line-height: 200px;
35       position: absolute;
36       left: 0;
37       top: 0;
38       background-color: black;
39     }
40     ul li:nth-child(1){
41       transform: rotateY(60deg) translateZ(200px);
42     }

```

```

43     ul li:nth-child(2){
44         transform: rotateY(120deg) translateZ(200px);
45     }
46     ul li:nth-child(3){
47         transform: rotateY(180deg) translateZ(200px);
48     }
49     ul li:nth-child(4){
50         transform: rotateY(240deg) translateZ(200px);
51     }
52     ul li:nth-child(5){
53         transform: rotateY(300deg) translateZ(200px);
54     }
55     ul li:nth-child(6){
56         transform: rotateY(360deg) translateZ(200px);
57     }
58     ul li img{
59         width: 200px;
60         height: 200px;
61         border: 5px solid skyblue;
62         box-sizing: border-box;
63     }
64     ul:hover{
65         animation-play-state: paused;
66     }
67     ul:hover li img{
68         opacity: 0.5;
69     }
70     ul li:hover img{
71         opacity: 1;
72     }
73     @keyframes sport {
74         from{
75             /*
76             注意点:
77             1.动画中如果有和默认样式中同名的属性，会覆盖默认样式中同名的属性
78             2.在编写动画的时候，固定不变的值写在前面，需要变化的值写在后面
79             */
80             transform: rotateX(-10deg) rotateY(0deg);
81         }
82         to{
83             transform: rotateX(-10deg) rotateY(360deg);
84         }
85     }
86     .heart{
87         width: 173px;
88         height: 157px;
89         position: absolute;
90         left: 100px;
91         bottom: 100px;
92         animation: move 10s linear 0s infinite normal;
93     }
94     @keyframes move {
95         0%{
96             left: 100px;
97             bottom: 100px;
98             opacity: 1;
99         }
100        20%{

```



```

101         left: 300px;
102         bottom: 300px;
103         opacity: 0;
104     }
105     40%{
106         left: 500px;
107         bottom: 500px;
108         opacity: 1;
109     }
110     60%{
111         left: 800px;
112         bottom: 300px;
113         opacity: 0;
114     }
115     80%{
116         left: 1200px;
117         bottom: 100px;
118         opacity: 1;
119     }
120     100%{
121         left: 800px;
122         bottom: -200px;
123     }
124 }
125 </style>
126 </head>
127 <body>
128 <ul>
129     <li></li>
130     <li></li>
131     <li></li>
132     <li></li>
133     <li></li>
134     <li></li>
135 </ul>
136 
137 <audio src="images/jacky/江哥最爱的歌.mp3" autoplay="autoplay" loop="loop">
</audio>
138 </body>
139 </html>

```

背景尺寸

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>112-背景尺寸属性</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        ul{
12            width: 800px;

```

```
13         height: 500px;
14         margin: 0 auto;
15     }
16     ul li{
17         list-style: none;
18         float: left;
19         width: 200px;
20         height: 200px;
21         margin: 30px 30px;
22         border: 1px solid #000;
23         text-align: center;
24         line-height: 200px;
25     }
26     ul li:nth-child(1){
27         background: url("images/dog.jpg") no-repeat;
28     }
29     ul li:nth-child(2){
30         background: url("images/dog.jpg") no-repeat;
31         /*
32         第一个参数:宽度
33         第二个参数:高度
34         */
35         background-size:200px 100px;
36     }
37     ul li:nth-child(3){
38         background: url("images/dog.jpg") no-repeat;
39         /*
40         第一个参数:宽度
41         第二个参数:高度
42         */
43         background-size:100% 80%;
44     }
45     ul li:nth-child(4){
46         background: url("images/dog.jpg") no-repeat;
47         /*
48         第一个参数:宽度
49         第二个参数:高度
50         */
51         background-size:auto 100px;
52     }
53     ul li:nth-child(5){
54         background: url("images/dog.jpg") no-repeat;
55         /*
56         第一个参数:宽度
57         第二个参数:高度
58         */
59         background-size:100px auto;
60     }
61     ul li:nth-child(6){
62         background: url("images/dog.jpg") no-repeat;
63         /*
64         cover含义:
65         1.告诉系统图片需要等比拉伸
66         2.告诉系统图片需要拉伸到宽度和高度都填满元素
67         */
68         background-size:cover;
69     }
70     ul li:nth-child(7){
```

```

71         background: url("images/dog.jpg") no-repeat;
72         /*
73         cover含义:
74         1.告诉系统图片需要等比拉伸
75         2.告诉系统图片需要拉伸到宽度或高度都填满元素
76         */
77         background-size:contain;
78     }
79 </style>
80 </head>
81 <body>
82 <!--
83 1.什么是背景尺寸属性
84 背景尺寸属性是CSS3中新增的一个属性，专门用于设置背景图片大小
85 -->
86 <ul>
87     <li>默认</li>
88     <li>具体像素</li>
89     <li>百分比</li>
90     <li>宽度等比拉伸</li>
91     <li>高度等比拉伸</li>
92     <li>cover</li>
93     <li>contain</li>
94 </ul>
95 </body>
96 </html>

```

背景图片定位区域

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>113-背景图片定位区域属性</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul li{
12             list-style: none;
13             float: left;
14             width: 100px;
15             height: 100px;
16             text-align: center;
17             line-height: 100px;
18             border: 20px dashed #000;
19             padding: 50px;
20             margin-left: 20px;
21             background: url("images/dog.jpg") no-repeat;
22         }
23         ul li:nth-child(2){
24             /*
25             告诉系统背景图片从什么区域开始显示，
26             默认情况下就是从padding区域开始显示

```

```

27         */
28         background-origin: padding-box;
29     }
30     ul li:nth-child(3){
31         background-origin: border-box;
32     }
33     ul li:nth-child(4){
34         background-origin: content-box;
35     }
36 </style>
37 </head>
38 <body>
39 <ul>
40     <li>默认</li>
41     <li>padding</li>
42     <li>border</li>
43     <li>content</li>
44 </ul>
45 </body>
46 </html>

```

背景绘制区域

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>114-背景绘制区域属性</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        ul li{
12            list-style: none;
13            float: left;
14            width: 100px;
15            height: 100px;
16            text-align: center;
17            line-height: 100px;
18            border: 20px dashed #000;
19            padding: 50px;
20            margin-left: 20px;
21            background: red url("images/dog.jpg") no-repeat;
22        }
23        ul li:nth-child(2){
24            /*
25            背景绘制区域属性是专门用于指定从哪个区域开始绘制背景的，默认情况下会从
border区域开始绘制背景
26            */
27            background-clip: padding-box;
28        }
29        ul li:nth-child(3){
30            background-clip: border-box;
31        }

```

```

32         ul li:nth-child(4){
33             background-clip: content-box;
34         }
35     </style>
36 </head>
37 <body>
38 <ul>
39     <li>默认</li>
40     <li>padding</li>
41     <li>border</li>
42     <li>content</li>
43 </ul>
44 </body>
45 </html>

```

多重背景图片

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>115-多重背景图片</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 500px;
13             height: 500px;
14             border: 1px solid #000;
15             margin: 0 auto;
16             /*
17              多张背景图片之间用逗号隔开即可
18              注意点：
19              先添加的背景图片会盖住后添加的背景图片
20              建议在编写多重背景时拆开编写
21              */
22             /*background: url("images/animal1.png") no-repeat left
top,url("images/animal2.png") no-repeat right top,url("images/animal3.png")
no-repeat left bottom,url("images/animal4.png") no-repeat right
bottom,url("images/animal5.png") no-repeat center center;*/
23             background-image:
url("images/animal1.png"),url("images/animal2.png"),url("images/animal3.png"
);
24             background-repeat: no-repeat, no-repeat, no-repeat;
25             background-position: left top, right top, left bottom;
26         }
27     </style>
28 </head>
29 <body>
30 <div></div>
31 </body>
32 </html>

```

多重背景图片练习

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>116-多重背景图片-练习</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    div{
12      width: 600px;
13      height: 190px;
14      border: 1px solid #000;
15      margin: 100px auto;
16      background-image: url("images/bg-plane.png"),url("images/bg-
17      sun.png"), url("images/bg-clouds.png");
18      background-repeat: no-repeat, no-repeat, no-repeat;
19      background-size: 50px 50px, 50px 50px, auto auto;
20      background-position: 50px 150px, 400px 50px, 0px 0px;
21      animation: move 10s linear 0s infinite normal;
22    }
23    @keyframes move {
24      from{
25        background-position: 50px 150px, 400px 50px, 0px 0px;
26      }
27      to{
28        background-position: 500px -150px, 400px 50px, -600px 0px;
29      }
30    }
31  </style>
32 </head>
33 <body>
34 <div></div>
35 </body>
36 </html>
```

css编写格式

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>117-CSS编写格式</title>
6
7   <!--<link rel="stylesheet" href="117-abc.css">-->
8
9   <style>
10     @import "117-abc.css";
11   </style>
```

```

12
13     <style>
14         div{
15             color: blue;
16         }
17     </style>
18
19
20
21
22 </head>
23 <body>
24 <!--
25 1. 行内样式
26 2. 内嵌样式
27 3. 外链样式  -- 企业开发用外链样式
28 4. 导入样式
29
30
31 外链样式和导入样式区别：
32 共同点： 都是将CSS代码写到了一个单独的文件中
33 不同点：
34 外链样式， 在显示网页时， 会先加载CSS文件， 再显示页面
35 导入样式， 在显示网页时， 会先显示界面， 再加载CSS文件
36
37 外链样式是通过一个HTML标签引入CSS的
38 而导入样式是通过@import引入CSS的， 而@import是CSS2.1推出， 所以导入样式存在兼容问题
39
40
41 优先级问题
42 行内样式的优先级最高
43 其它写法谁写在后面就听谁的
44 -->
45 <!--<div style="color: red">我是div</div>-->
46 <div>我是div</div>
47
48 </body>
49 </html>

```

边框圆角

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>118-边框圆角基本概念</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         .father{
12             width: 200px;
13             height: 200px;
14             border: 1px solid #000;

```

```

15         margin: 100px auto;
16     }
17     .father>.son{
18         width: 200px;
19         height: 200px;
20         background-color: pink;
21         /*
22         边框圆角的作用：
23         将原始的直角变为圆角
24         格式：
25         border-radius: 100px 0px 0px 0px;
26         第一个参数：指定左上角的半径
27         按照左上/右上/右下/左下的顺序
28         */
29         /*border-radius: 0px 0px 0px 100px;*/
30         /*如果省略一个参数，会变成对角的值*/
31         /*border-radius: 100px 80px 40px;*/
32         /*如果省略两个参数，会变成对角的值*/
33         /*border-radius: 100px 80px;*/
34         /*如果省略三个参数，其它角会和它一样*/
35         /*border-radius: 100px;*/
36         /*如果指定的半径是当前元素宽高的一半，那么就是一个圆形*/
37         /*border-radius: 100px;*/
38         border-radius: 50%;
39     }
40 </style>
41 </head>
42 <body>
43 <div class="father">
44     <div class="son"></div>
45 </div>
46 </body>
47 </html>

```

单独指定圆角和椭圆

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        .father{
12            width: 200px;
13            /*height: 200px;*/
14            height: 100px;
15            border: 1px solid #000;
16            margin: 100px auto;
17        }
18        .father>.son {
19            width: 200px;

```



```

20      /*height: 200px;*/
21      height: 100px;
22      background-color: pink;
23      /*单独指定某一个角的半径*/
24      /*border-top-left-radius: 100px;*/
25      /*border-top-right-radius: 100px;*/
26      /*border-bottom-left-radius: 100px;*/
27      /*border-bottom-right-radius: 100px;*/
28
29      /*如果传递了两个参数，那么第一个参数代表水平方向的半径，第二个参数代表垂直方
    向的半径*/
30      /*
31      border-top-left-radius: 100px 50px;
32      border-top-right-radius: 100px 50px;
33      border-bottom-left-radius: 100px 50px;
34      border-bottom-right-radius: 100px 50px;
35      */
36
37      /*border-top-left-radius: 50%;*/
38      /*border-top-right-radius: 50%;*/
39      /*border-bottom-left-radius: 50%;*/
40      /*border-bottom-right-radius: 50%;*/
41
42      /*
43      斜杠之前的代表左上/右上/右下/左下的水平方向的半径
44      斜杠之后的代表左上/右上/右下/左下的垂直方向的半径
45      */
46      /*border-radius: 100px 100px 100px 100px/50px 50px 50px 50px;*/
47      border-radius: 100px/50px;
48  }
49  </style>
50 </head>
51 <body>
52 <div class="father">
53     <div class="son"></div>
54 </div>
55 </body>
56 </html>

```

线性渐变

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>125-线性渐变</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         div{
12             width: 500px;
13             height: 200px;
14             border: 1px solid #000;

```

```

15         margin: 50px auto;
16         /*
17         渐变不是一个新的属性，而是一个取值
18         默认情况下线性渐变是从上至下的渐变
19         我们可以通过在颜色的前面添加to xxx修改渐变的方向
20         to top
21         to left
22         to right
23
24         */
25         /*background-image: linear-gradient(to top right, red, blue);*/
26
27         /*除了可以通过关键字控制渐变的方向以外，还可以通过角度来控制渐变的方向，角度
28         是按照顺时针旋转*/
29         /*background-image: linear-gradient(200deg, red, blue);*/
30
31         /*30%以前不渐变，从30%开始慢慢渐变到下一个颜色*/
32         /*background-image: linear-gradient(red 30%, blue);*/
33         /*background-image: linear-gradient(red 0%, red 30%,blue 30%,
34         blue 60%, green 60%, green 100%);*/
35         /*background-image: linear-gradient(to left ,red 0%, red
36         30%,blue 30%, blue 60%, green 60%, green 100%);*/
37     }
38 </style>
39 </head>
40 <body>
41 <div></div>
42 </body>
43 </html>

```

径向渐变

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>126-径向渐变</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        div{
12            width: 200px;
13            height: 200px;
14            margin: 100px auto;
15            border: 1px solid #000;
16            /*可以在颜色前面添加at和关键字来指定从什么位置开始渐变*/
17            /*background-image: radial-gradient(at left top,red, blue);*/
18            /*除了可以通过关键字指定从什么位置开始渐变以外，还可以通过指定具体像素来指定
19            从什么位置开始渐变*/
20            /*background-image: radial-gradient(at 50px 50px,red, blue);*/
21
22            /*也可以在颜色前面直接加上一个具体的像素来指定渐变的范围*/

```

```

22      /*background-image: radial-gradient(150px ,red, blue);*/
23
24      /*
25      编写规范：
26      如果既要指定扩散的范围，又要指定起始位置，那么把扩散的范围写在前面，而起始位
置写在后面
27      */
28      /*background-image: radial-gradient(50px at right bottom ,red,
blue);*/
29    }
30  </style>
31 </head>
32 <body>
33
34 <div></div>
35 </body>
36 </html>

```

伸缩布局

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>127-伸缩布局</title>
6    <style>
7      *{
8        margin: 0;
9        padding: 0;
10     }
11     .one{
12       border: 1px solid #000;
13       margin: 100px auto;
14       list-style: none;
15       overflow: hidden;
16     }
17     .one>li{
18       float: left;
19       width: 100px;
20       height: 100px;
21       background-color: pink;
22       margin: 20px;
23       text-align: center;
24       line-height: 100px;
25       font-size: 35px;
26     }
27     .two{
28       border: 1px solid #000;
29       margin: 0 auto;
30       list-style: none;
31
32       display: flex;
33     }
34     .two>li{
35       width: 100px;

```

```

36         height: 100px;
37         background-color: pink;
38         margin: 20px;
39         text-align: center;
40         line-height: 100px;
41         font-size: 35px;
42     }
43 </style>
44 </head>
45 <body>
46 <ul class="one">
47     <li>1</li>
48     <li>2</li>
49     <li>3</li>
50 </ul>
51
52 <ul class="two">
53     <li>1</li>
54     <li>2</li>
55     <li>3</li>
56 </ul>
57 </body>
58 </html>

```

伸缩布局-主轴方向

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             border: 1px solid #000;
13             margin: 100px auto;
14             list-style: none;
15             /*
16              在伸缩布局中，默认伸缩项是从左至右的排版的
17             */
18             display: flex;
19             /*
20              主轴的排版的方向默认就是row，默认就是从左至右
21             */
22             /*flex-direction: row;*/
23             /*
24              修改主轴的排版方向为从右至左
25             */
26             /*flex-direction: row-reverse;*/
27             /*
28              告诉系统把主轴的方向改为垂直方向
29              注意点：

```

```

30      1.默认情况下主轴是水平方向的，但是也可以修改为垂直方向.只要看到flex-
direction: column/column-reverse就代表主轴被修改为了垂直方向
31      2.如果将主轴修改为了垂直方向，那么侧轴就会自动从垂直方向转换为水平方向
32      */
33      /*
34      修改主轴的排版方向为从上至下
35      */
36      /*flex-direction: column;*/
37
38      /*
39      修改主轴的排版方向为从下至上
40      */
41      flex-direction: column-reverse;
42  }
43  ul>li{
44      width: 100px;
45      height: 100px;
46      background-color: pink;
47      text-align: center;
48      line-height: 100px;
49      font-size: 35px;
50      margin: 20px;
51  }
52  </style>
53  </head>
54  <body>
55
56  <ul>
57      <li>1</li>
58      <li>2</li>
59      <li>3</li>
60  </ul>
61  </body>
62  </html>

```

伸缩布局-主轴对齐

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 600px;
13             height: 600px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             list-style: none;
17             display: flex;
18             /*

```

```

19         justify-content:
20         用于设置伸缩项主轴上的对齐方式
21         如果设置为flex-start，代表告诉系统伸缩项和主轴的起点对齐
22         */
23         /*justify-content: flex-start;*/
24         /*justify-content: flex-end;*/
25         /*居中对齐*/
26         /*justify-content: center;*/
27         /*两端对齐*/
28         /*justify-content: space-between;*/
29         /*环绕对齐*/
30         justify-content: space-around;
31     }
32     ul>li{
33         width: 100px;
34         height: 100px;
35         background-color: pink;
36         text-align: center;
37         line-height: 100px;
38         font-size: 35px;
39     }
40     ul>li:nth-child(2){
41         background-color: skyblue;
42     }
43     ul>li:nth-child(3){
44         background-color: yellow;
45     }
46 </style>
47 </head>
48 <body>
49
50 <ul>
51     <li>1</li>
52     <li>2</li>
53     <li>3</li>
54 </ul>
55 </body>
56 </html>

```

伸缩布局-侧轴对齐

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 600px;
13             height: 600px;
14             border: 1px solid #000;

```

```

15         margin: 100px auto;
16         list-style: none;
17         display: flex;
18         /*
19         通过align-items可以修改侧轴的对齐方式
20         默认情况下是以侧轴的起点对齐
21         */
22         /*align-items: flex-start;
23         /*align-items: flex-end;*/
24         /*align-items: center;*/
25         /*
26         注意点：
27         和主轴不同的是，侧轴没有两端对齐和环绕对齐
28         */
29         /*基线对齐*/
30         /*align-items: baseline;*/
31         /*拉伸对齐*/
32         align-items: stretch;
33
34     }
35     ul>li{
36         width: 100px;
37         /*height: 100px;*/
38         background-color: pink;
39         text-align: center;
40         line-height: 100px;
41         font-size: 35px;
42     }
43     ul>li:nth-child(1){
44         /*padding-top: 50px;*/
45     }
46     ul>li:nth-child(2){
47         background-color: skyblue;
48     }
49     ul>li:nth-child(3){
50         /*padding-top: 100px;*/
51         background-color: yellow;
52     }
53 </style>
54 </head>
55 <body>
56
57 <ul>
58     <li>j</li>
59     <li>y</li>
60     <li>z</li>
61 </ul>
62 </body>
63 </html>

```

align-self

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>128-伸缩布局-主轴方向</title>

```

```

6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 600px;
13             height: 600px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             list-style: none;
17             display: flex;
18         }
19         ul>li{
20             width: 100px;
21             height: 100px;
22             background-color: pink;
23             text-align: center;
24             line-height: 100px;
25             font-size: 35px;
26         }
27         ul>li:nth-child(1){
28             /*padding-top: 50px;*/
29             /*
30              如果在伸缩容器中通过 align-items 修改侧轴的对齐方式，是修改所有伸缩项侧轴
31              的对齐方式
32              如果是在伸缩项中通过 align-self 修改侧轴的对齐方式，是单独修改当前伸缩项侧
33              轴的对齐方式
34              align-self属性的取值和align-items一样
35              */
36             align-self: flex-end;
37         }
38         ul>li:nth-child(2){
39             background-color: skyblue;
40             align-self: center;
41         }
42         ul>li:nth-child(3){
43             /*padding-top: 100px;*/
44             background-color: yellow;
45             align-self: flex-end;
46         }
47     </style>
48 </head>
49 <body>
50     <ul>
51         <li>j</li>
52         <li>y</li>
53         <li>z</li>
54     </ul>
55 </body>
56 </html>

```

伸缩布局-主轴侧轴的方向问题


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>128-伸缩布局-主轴方向</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        ul{
12            width: 600px;
13            height: 600px;
14            border: 1px solid #000;
15            margin: 100px auto;
16            list-style: none;
17            display: flex;
18            /*
19             默认情况下主轴是水平方向，侧轴是垂直方向
20             默认情况下主轴的起点在伸缩容器的最左边，主轴的终点在伸缩容器的最右边
21             默认情况下侧轴的起点在伸缩容器的最顶部，侧轴的终点在伸缩容器的最底部
22             我们可以通过flex-direction属性修改主轴的方向
23             如果flex-direction属性的取值是column/column-reverse
24             就代表将主轴的方向修改为了垂直方向，那么侧轴就会立刻变为水平方向
25            */
26            /*justify-content: space-around;*/
27            /*flex-direction: column;*/
28            /*align-items: center;*/
29        }
30        ul>li{
31            width: 100px;
32            height: 100px;
33            background-color: pink;
34            text-align: center;
35            line-height: 100px;
36            font-size: 35px;
37        }
38        ul>li:nth-child(1){
39
40        }
41        ul>li:nth-child(2){
42            background-color: skyblue;
43        }
44        ul>li:nth-child(3){
45            background-color: yellow;
46        }
47    </style>
48 </head>
49 <body>
50
51 <ul>
52     <li>j</li>
53     <li>y</li>
54     <li>z</li>
55 </ul>
56 </body>
57 </html>
```

伸缩布局-设置换行

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>128-伸缩布局-主轴方向</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    ul{
12      width: 400px;
13      height: 600px;
14      border: 1px solid #000;
15      margin: 100px auto;
16      list-style: none;
17      display: flex;
18      /*
19        在伸缩布局中，如果伸缩容器的宽度不够，系统会自动压缩伸缩项的宽度，保证所有的
        伸缩项都能放在伸缩容器中
20        如果当伸缩容器宽度不够时，不想让伸缩项被压缩，也可以让系统换行flex-wrap:
        wrap;
21
22        默认情况下如果伸缩容器的高度比换行之后所有伸缩项的高度还要高，那么系统会自动
        将剩余空间平分之后添加给每一行
23        */
24        /*宽度不够也不换行，默认取值*/
25        /*flex-wrap: nowrap;*/
26        flex-wrap: wrap;
27        /*flex-wrap: wrap-reverse;*/
28      }
29      ul>li{
30        width: 200px;
31        height: 200px;
32        background-color: pink;
33        text-align: center;
34        line-height: 200px;
35        font-size: 35px;
36      }
37      ul>li:nth-child(1){
38
39      }
40      ul>li:nth-child(2){
41        background-color: skyblue;
42      }
43      ul>li:nth-child(3){
44        background-color: yellow;
45      }
46    </style>
47 </head>
48 <body>
49
50 <ul>
```

```
51     <li>1</li>
52     <li>2</li>
53     <li>3</li>
54 </ul>
55 </body>
56 </html>
```

伸缩布局-换行对齐模式

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 400px;
13             height: 600px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             list-style: none;
17             display: flex;
18             flex-wrap: wrap;
19             /*
20             如果伸缩容器中的伸缩项换行了，那么我们就可以通过align-content来设置行与行
                之间的对齐方式
21             */
22             /*align-content: flex-start;*/
23             /*align-content: flex-end;*/
24             /*align-content: center;*/
25             /*align-content: space-between;*/
26             /*align-content: space-around;*/
27             /*
28             默认情况下换行就是就是拉伸对齐
29             一定要注意：在换行中的拉伸对齐是指，所有行的高度总和要和伸缩容器的高度一样
30             所以会将多余的空间平分之后添加给每一行
31             */
32             align-content: stretch;
33         }
34         ul>li{
35             width: 200px;
36             height: 200px;
37             background-color: pink;
38             text-align: center;
39             line-height: 200px;
40             font-size: 35px;
41         }
42         ul>li:nth-child(1){
43
44         }
45         ul>li:nth-child(2){
```

```

46         background-color: skyblue;
47     }
48     ul>li:nth-child(3){
49         background-color: yellow;
50     }
51 </style>
52 </head>
53 <body>
54
55 <ul>
56     <li>1</li>
57     <li>2</li>
58     <li>3</li>
59 </ul>
60 </body>
61 </html>

```

伸缩布局-伸缩项排序

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 400px;
13             height: 400px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             list-style: none;
17             display: flex;
18         }
19         ul>li{
20             width: 100px;
21             height: 100px;
22             background-color: pink;
23             text-align: center;
24             line-height: 100px;
25             font-size: 35px;
26         }
27         ul>li:nth-child(1){
28             /*
29             如果想调整伸缩布局中伸缩项的顺序，那么我们可以通过修改伸缩项的order属性来实
30             现
31             默认情况下order的取值是0
32             如果我们设置了order属性的值，那么系统就会按照设置的值从小到大的排序
33             */
34             order:-1;
35         }
36         ul>li:nth-child(2){

```

```

36         background-color: skyblue;
37         order: 2;
38     }
39     ul>li:nth-child(3){
40         background-color: yellow;
41         order: 1;
42     }
43 </style>
44 </head>
45 <body>
46
47 <ul>
48     <li>1</li>
49     <li>2</li>
50     <li>3</li>
51 </ul>
52 </body>
53 </html>

```

伸缩布局-伸缩项放大比

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 400px;
13             height: 400px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             list-style: none;
17             display: flex;
18         }
19         ul>li{
20             width: 100px;
21             height: 100px;
22             background-color: pink;
23             text-align: center;
24             line-height: 100px;
25             font-size: 35px;
26         }
27         ul>li:nth-child(1){
28             /*
29             1.flex-grow作用:
30             当所有伸缩项宽度的总和没有伸缩容器宽度大的时，我们可以通过flex-grow让系统调
31             整伸缩项的宽度，以便于让所有伸缩项的宽度的总和等于伸缩容器的宽度
32
33             2.flex-grow计算公式
34             2.1计算剩余的空间

```

```

34      伸缩容器宽度 - 所有伸缩项的宽度
35      400 - 300 = 100
36      2.2计算每份剩余空间的宽度
37      剩余空间 / 需要的份数
38      100 / 6 = 16.66
39      2.3计算每个伸缩项最终的宽度
40      伸缩项的宽度 + 需要的份数 * 每份的宽度
41
42      注意点：
43      flex-grow默认值是0
44      */
45      flex-grow: 1;
46      /*100 + 1 * 16.66 = 116.66*/
47  }
48  ul>li:nth-child(2){
49      background-color: skyblue;
50      flex-grow: 2;
51      /*100 + 2 * 16.66 = 133.3*/
52  }
53  ul>li:nth-child(3){
54      background-color: yellow;
55      flex-grow: 3;
56      /*100 + 3 * 16.66 = 149.8*/
57  }
58  </style>
59  </head>
60  <body>
61
62  <ul>
63      <li>1</li>
64      <li>2</li>
65      <li>3</li>
66  </ul>
67  </body>
68  </html>

```

伸缩布局-伸缩项缩小比

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 400px;
13             height: 400px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             list-style: none;
17             display: flex;

```

```

18     }
19     ul>li{
20         width: 200px;
21         height: 200px;
22         background-color: pink;
23         text-align: center;
24         line-height: 200px;
25         font-size: 35px;
26     }
27     ul>li:nth-child(1){
28         /*
29         1.flex-shrink作用:
30         当所有伸缩项宽度的总和比伸缩容器宽度大的时，我们可以通过flex-shrink让系
统调整伸缩项的宽度，以便于让所有伸缩项的宽度的总和等于伸缩容器的宽度
31
32         2.计算每个伸缩项需要压缩的宽度
33         2.1计算溢出的宽度
34         伸缩容器的宽度 - 所有伸缩项的宽度总和
35         400 - 600 = -200
36         2.2计算总权重
37         每个伸缩项需要的份数 * 每个伸缩项的宽度
38         1 * 200 + 2 * 200 + 3 * 200 = 1200
39         2.3计算每个伸缩项需要压缩的宽度
40         溢出的宽度 * 需要的份数 * 每个伸缩项的宽度 / 总权重
41         -200 * 1 * 200 / 1200 = -33.33
42
43         注意点:
44         flex-shrink: 默认值是1
45         */
46         flex-shrink: 1;
47     }
48     ul>li:nth-child(2){
49         background-color: skyblue;
50         flex-shrink: 2;
51     }
52     ul>li:nth-child(3){
53         background-color: yellow;
54         flex-shrink: 3;
55     }
56 </style>
57 </head>
58 <body>
59
60 <ul>
61     <li>1</li>
62     <li>2</li>
63     <li>3</li>
64 </ul>
65 </body>
66 </html>

```

伸缩布局-伸缩项自适应

```

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

```

```

3 <head>
4   <meta charset="UTF-8">
5   <title>128-伸缩布局-主轴方向</title>
6   <style>
7     *{
8       margin: 0;
9       padding: 0;
10    }
11    ul{
12      width: 400px;
13      height: 400px;
14      border: 1px solid #000;
15      margin: 100px auto;
16      list-style: none;
17      display: flex;
18    }
19    ul>li{
20      width: 200px;
21      height: 200px;
22      background-color: pink;
23      text-align: center;
24      line-height: 200px;
25      font-size: 35px;
26    }
27    ul>li:nth-child(1){
28      /*
29      如果有伸缩项没有设置flex-grow，那么系统会保持原有的宽度
30      而会将多余的宽度等分之后，按照每个伸缩项需要的份数添加给它们
31      如果想让某个伸缩项不缩小，那么需要将它flex-shrink设置为0
32      */
33      flex-shrink: 0;
34    }
35    ul>li:nth-child(2){
36      background-color: skyblue;
37      flex-grow: 1;
38    }
39    ul>li:nth-child(3){
40      background-color: yellow;
41      flex-grow: 4;
42    }
43  </style>
44 </head>
45 <body>
46
47
48 <ul>
49   <li>1</li>
50   <li>2</li>
51   <li>3</li>
52 </ul>
53 </body>
54 </html>

```

伸缩布局-伸缩项宽度


```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>128-伸缩布局-主轴方向</title>
6     <style>
7         *{
8             margin: 0;
9             padding: 0;
10        }
11        ul{
12            width: 800px;
13            height: 400px;
14            border: 1px solid #000;
15            margin: 100px auto;
16            list-style: none;
17            display: flex;
18        }
19        ul>li{
20            /*width: 200px;*/
21            height: 200px;
22            background-color: pink;
23            text-align: center;
24            line-height: 200px;
25            font-size: 35px;
26        }
27        ul>li:nth-child(1){
28            /*
29            如果是伸缩布局，除了可以通过元素的width属性来设置宽度以外，还可以通过flex-
30            basis属性来设置伸缩项的宽度
31
32            注意点：
33            1.width属性可以设置宽度/flex-basis也可以设置宽度
34            那么如果两者同时存在，系统会听flex-basis的
35            2.flex-basis属性是专门提供给伸缩布局使用的
36            3.如果同时通过这两个属性设置了宽度，但是其中一个是auto，那么系统会按照具体
37            值来设置
38
39            */
40            /*flex-basis: 100px;*/
41            /*width: 200px;*/
42            /*width: auto;*/
43            flex-basis: auto;
44        }
45        ul>li:nth-child(2){
46            background-color: skyblue;
47            flex-basis: 200px;
48        }
49        ul>li:nth-child(3){
50            background-color: yellow;
51            flex-basis: 300px;
52        }
53    </style>
54 </head>
55 <body>
56 <ul>
57     <li>1</li>
58     <li>2</li>

```

```
57     <li>3</li>
58 </ul>
59 </body>
60 </html>
```

伸缩布局-伸缩项属性简写

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>128-伸缩布局-主轴方向</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         ul{
12             width: 800px;
13             height: 400px;
14             border: 1px solid #000;
15             margin: 100px auto;
16             list-style: none;
17             display: flex;
18         }
19         ul>li{
20             width: 200px;
21             height: 200px;
22             background-color: pink;
23             text-align: center;
24             line-height: 200px;
25             font-size: 35px;
26         }
27         ul>li:nth-child(1){
28             /*
29              flex: flex-grow flex-shrink flex-basis;
30              默认值:
31              flex: 0 1 auto;
32
33              注意点:
34              在连写格式中, flex-shrink flex-basis是可以省略的
35              */
36             /*flex: 1 1 200px;*/
37             flex: 1;
38         }
39         ul>li:nth-child(2){
40             background-color: skyblue;
41             /*flex: 1 2 200px;*/
42             flex: 2;
43         }
44         ul>li:nth-child(3){
45             background-color: yellow;
46             /*flex: 1 3 200px;*/
47             flex: 3;
48         }
49     </style>
50 </head>
51 <body>
52     <ul>
53         <li>1</li>
54         <li>2</li>
55         <li>3</li>
56     </ul>
```

```

49     </style>
50 </head>
51 <body>
52
53 <ul>
54     <li>1</li>
55     <li>2</li>
56     <li>3</li>
57 </ul>
58 </body>
59 </html>

```

媒体查询

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>143-媒体查询</title>
6      <style>
7          *{
8              margin: 0;
9              padding: 0;
10         }
11         /*
12         需求:
13         PC显示小牛(大于等于980)
14         pad显示小猪(小于980, 并且大于等于620)
15         phone显示小马(小于620)
16         */
17
18         /*如果浏览器的宽度大于等于980, 那么就执行后面{}中的代码*/
19         @media(min-width: 980px) {
20             body{
21                 background: url("images/animal1.png");
22             }
23         }
24         /*如果浏览器的宽度大于等于620, 并且宽度小于980*/
25         /*注意点: and的两边必须添加空格*/
26         @media(min-width: 620px) and (max-width: 979px) {
27             body{
28                 background: url("images/animal2.png");
29             }
30         }
31         @media(max-width: 619px) {
32             body{
33                 background: url("images/animal3.png");
34             }
35         }
36     </style>
37 </head>
38 <body>
39
40 </body>
41 </html>

```

媒体查询-外链式

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>144-媒体查询-外链式</title>
6     <link rel="stylesheet" href="144-style-pc.css" media="(min-
width:980px)">
7     <link rel="stylesheet" href="144-style-pad.css" media="(min-width:620px)
and (max-width:979px)">
8     <link rel="stylesheet" href="144-style-phone.css" media="(max-
width:619px)">
9
10 </head>
11 <body>
12
13 </body>
14 </html>
```

grid布局

<https://enen.me/wp-content/uploads/2021/11/grid.pdf>

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Document</title>
7     <style>
8         .main {
9             display: grid;
10            width: 600px;
11            height: 600px;
12            border: 10px solid skyblue;
13            grid-template-columns: 100px 100px 100px;
14            /* 等价于 */
15            grid-template-columns: repeat(3,100px);
16            /* auto-fill的意思是如果一行能放下多少就放下多少，放不下的就换行*/
17            grid-template-columns: repeat(auto-fill,100px);
18            /* 为了方便表示比例关系，网格布局提供了fr关键字（fraction 的缩写，意为"片
段"）*/
19            grid-template-columns: repeat(3,1fr);
20            /* minmax(), 函数产生一个长度范围，表示长度就在这个范围之内，它接受两个参
数，分别为最小值和最大值 */
21            grid-template-columns: 1fr minmax(150px,1fr);
22            /* auto, 表示由浏览器自己决定长度 */
23            grid-template-columns: 100px auto 100px;
24            /* 网格线，可以用方括号定义网格线名称，方便以后的引用 */
25            grid-template-columns: [c1] 100px [c2] 100px [c3] 100px [c4];
```

```

26      /* 间隔 */
27      row-gap: 20px;
28      column-gap: 20px;
29      gap: 20px 20px;
30      /* 一个区域由单个或多个单元格组成，由你决定（具体使用，需要在项目属性里面设置） */
31      grid-template-areas: 'a b c'
32                           'd e f'
33                           'g h i';
34      grid-template-areas: 'a a a'
35                           'b b b'
36                           'c c c';
37      grid-template-areas: 'a . c'
38                           'd . f'
39                           'g . i';
40      /* 划分网格以后，容器的子元素会按照顺序，自动放置在每一个网格。默认的放置顺序是“先行后列”，即先填满第一行，再开始放入第二行 */
41      grid-auto-flow: row;
42      grid-auto-flow: column;
43      grid-auto-flow: row dense;
44      /* 容器属性 justify-items(水平方向) / align-items (垂直方向) */
45      /* justify-items: start | end | center | stretch; */
46      /* align-items: start | end | center | stretch; */
47      /* place-items属性是align-items属性和justify-items属性的合并简写形式 */
48      /* 容器属性 justify-content (水平方向) / align-content (垂直方向) */
49      /* justify-content: start | end | center | stretch | space-around
| space-between | space-evenly; */
50      /* align-content: start | end | center | stretch | space-around |
space-between | space-evenly; */
51
52      /* grid-auto-columns / grid-auto-rows 用来设置多出来的项目宽和高 */
53      grid-auto-rows: 50px;
54
55      /* grid-column-start / grid-column-end / grid-row-start / grid-
row-end一句话解释，用来指定item的具体位置，根据在哪根网格线 */
56      /* 列方向4根网格线，第一个项目占用了从第一根到第三根 */
57      grid-column-start: 1;
58      grid-column-end: 3;
59      /* 简写 */
60      grid-column: 1 / 3;
61      grid-column-start: span 2;
62      /* grid-area指定项目放在哪一个区域 配合grid-template-areas使用 */
63      grid-area: b;
64
65      /* grid-area属性还可用作grid-row-start、grid-column-start、grid-
row-end、grid-column-end的合并简写形式，直接指定项目的位置
66      grid-area: <row-start> / <column-start> / <row-end> / <column-
end>; */
67      grid-area: 1 / 1 / 3 / 3;
68
69      /* justify-self / align-self / place-self 设置项目自己 */
70
71      grid-template-rows: 100px 100px 100px 100px;
72
73  }
74  .item {
75      background-color: red;

```

```
76     }
77     </style>
78 </head>
79 <body>
80     <div class="mian">
81         <div class="item item1"></div>
82         <div class="item item2"></div>
83         <div class="item item3"></div>
84         <div class="item item4"></div>
85         <div class="item item5"></div>
86         <div class="item item6"></div>
87         <div class="item item7"></div>
88         <div class="item item8"></div>
89         <div class="item item9"></div>
90         <div class="item item10"></div>
91     </div>
92 </body>
93 </html>
```

毛玻璃

```
1 | -webkit-backdrop-filter: blur(25px)
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Document</title>
7     <style>
8         body {
9             background: url("../images/bg.jpg");
10        }
11        div {
12            width: 300px;
13            height: 500px;
14            background: transparent;
15            margin: 0 auto;
16            box-shadow: 0 25px 50px rgba(0, 0, 0, .2);
17            backdrop-filter: blur(50px);
18        }
19    </style>
20 </head>
21 <body>
22     <div></div>
23 </body>
24 </html>
```

垂直居中

- 1 单行文字:
- 2 `line-height`和`height`设置一样

```
1 多行文字:
2  div {
3      background: blue;
4      height: 210px;
5      width: 170px;
6      display: flex;
7      flex-direction: column;
8      justify-content: center;
9      text-align: center;
10 }
```

```
1  grid布局:
2  div {
3      display: grid;
4      grid-template-columns: repeat(6,1fr);
5      align-items: center;
6      justify-content: center;
7  }
```

- 1 绝对定位

```
1  vertical-align加伪元素
2  div.search {
3      display: inline-block;
4      vertical-align: middle;
5  }
6  div.main::after {
7      content: "",
8      display: inline-block;
9      background-color: gold;
10     height: 100%;
11     width: 0;
12     vertical-align: middle;
13 }
```

水平居中

- 1 行内元素
- 2 `text-align: center;`
- 3 块级元素
- 4 `margin: 0 auto;`

水平垂直居中

```
1 1.绝对定位transform
2 2.绝对定位:left 0 right 0 bottom 0 top 0
3 3.flex
4 div {
5     display: flex;
6     align-items: center;
7     justify-content: center;
8 }
9 4.伪元素和vertical-align
```

文字显示两行

```
1 p {
2     overflow: hidden;
3     text-overflow: ellipsis;
4     display: -webkit-box;
5     -webkit-line-clamp: 2;
6     -webkit-box-orient: vertical;
7 }
```

BFC 定义

BFC(Block formatting context)直译为"块级格式化上下文"。它是一个独立的渲染区域，只有Block-level box参与，它规定了内部的Block-level Box如何布局，并且与这个区域外部毫不相干。

在解释什么是BFC之前，我们需要先知道Box、Formatting Context的概念。

Box: css布局的基本单位

Box 是 CSS 布局的对象和基本单位，直观点来说，就是一个页面是由很多个 Box 组成的。元素的类型和 display 属性，决定了这个 Box 的类型。不同类型的 Box，会参与不同的 Formatting Context（一个决定如何渲染文档的容器），因此Box内的元素会以不同的方式渲染。让我们看看有哪些盒子：

block-level box:display 属性为 block, list-item, table 的元素，会生成 block-level box。并且参与 block formatting context;

inline-level box:display 属性为 inline, inline-block, inline-table 的元素，会生成 inline-level box。并且参与 inline formatting context;

run-in box: css3 中才有，这儿先不讲了。

Formatting Context

Formatting context 是 W3C CSS2.1 规范中的一个概念。它是页面中的一块渲染区域，并且有一套渲染规则，它决定了其子元素将如何定位，以及和其他元素的关系和相互作用。最常见的 Formatting context 有 Block formatting context (简称BFC)和 Inline formatting context (简称IFC)。

BFC是一个独立的布局环境，其中的元素布局是不受外界的影响，并且在一个BFC中，块盒与行盒（行盒由一行中所有的内联元素所组成）都会垂直的沿着其父元素的边框排列。

BFC的布局规则

内部的Box会在垂直方向，一个接一个地放置。

Box垂直方向的距离由margin决定。属于同一个BFC的两个相邻Box的margin会发生重叠。

每个盒子（块盒与行盒）的margin box的左边，与包含块border box的左边相接触(对于从左往右的格式化，否则相反)。即使存在浮动也是如此。

BFC的区域不会与float box重叠。

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。

计算BFC的高度时，浮动元素也参与计算。

如何创建BFC

- 1、float的值不是none。
- 2、position的值不是static或者relative。
- 3、display的值是inline-block、table-cell、flex、table-caption或者inline-flex
- 4、overflow的值不是visible

BFC的作用

- 1.利用BFC避免margin重叠。
- 2.自适应两栏布局(一个浮动，一个overflow:hidden)
- 3.清楚浮动。

总结

以上例子都体现了：

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。

因为BFC内部的元素和外部的元素绝对不会互相影响，因此，当BFC外部存在浮动时，它不应该影响BFC内部Box的布局，BFC会通过变窄，而不与浮动有重叠。同样的，当BFC内部有浮动时，为了不影响外部元素的布局，BFC计算高度时会包括浮动的高度。避免margin重叠也是这样的道理。

inline和block的区别

display:block特点

- 1、独占一行，多个block元素另起一行，默认情况下，block元素宽度自动填满其父元素宽度
- 2、block元素可以设置width,height属性。块元素即使设置了宽度,仍然是独占一行。
- 3、block元素可以设置margin和padding属性。

display:inline特点

- 1、inline元素不会独占一行，多个相邻的行内元素会排列在同一行里，直到一行排列不下，才会新换一行，其宽度随元素的内容而变化
- 2、inline元素设置width,height属性无效。
- 3、inline元素的margin和padding属性，水平方向的padding-left, padding-right, margin-left, margin-right都产生边距效果；但垂直方向的padding-top, padding-bottom, margin-top, margin-bottom不会产生边距效果。

什么是文档流

文档流指的是文档中的[标签](#)在排列时所占用的位置。将窗体自上而下分成一行行，并在每行中按从左至右的顺序排放标签，即为文档流。也就是说在文档流中标签默认会紧贴到上一个标签的右边，如果右边不足以放下标签，标签则会另起一行，在新的一行中继续从左至右摆放。

目前常见的会影响元素脱离文档流的css属性有：

- ①float浮动。
- ②position的absolute和fixed定位。

