

CSS

主讲:石小俊

一.介绍css

1.什么是css

Cascading Style sheet

层叠样式表

为页面增加外观

2.作用

- 实现内容与样式的分离
- 便于样式的复用,有利于网站后期维护
- 使得页面精准控制,让页面更加美观

3.W3C建议

- 页面的内容,即数据,由html实现
- 页面的效果,即外观,由css实现
- 页面的行为,即功能,由javascript实现

二.css引入方式

1.行内式

为当前的某个元素添加样式

```
1  <标签名 style="属性名:属性值;属性名:属性值;..."></标签名>
```

2.内联样式

为某一种共性的多个元素进行统一的样式的控制

这些元素必须是当前页面的

```
1  <style>
2      选择器 {
3          属性名:属性值;
4          属性名:属性值;
5          ...
6      }
7  </style>
```

3.link引入

引入一个外部的css样式

该css样式可以用于当前工程中任意一个页面

为引入的页面中某一种共性的多个元素进行统一的样式控制

引入的css样式文件必须是一个 *.css 的文件

```
1 <link rel="stylesheet" href="*.css">
```

4.@import引入

该方式必须在style标签中使用

```
1 <style>
2     @import *.css
3 </style>
```

5.引入方式的优先级

- 行内式的优先级最高
- 内联样式与外部引入的优先级一致
 - 由于html加载顺序是从上往下,从左往右
 - 因此后面的代码会覆盖前面的代码
 - 因此:处于后面的会执行

6.破坏优先级

!important

在样式的属性值后面加上关键字 !important 会破坏当前样式的优先级

由 !important 标注的样式优先级最高

三.css选择器

1.选择器语法

选择器必须在style标签中或者某个css文件中进行编写

用于查找拥有某些共性的元素

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

2.基本选择器

- id选择器
 - 匹配指定的id值所对应的元素
 - 在一个文档中,id是唯一的
 - 在css中,如果存在相同的id值的元素,css中可以识别(但是这种语法是错误的)
 - 语法: #id值
- class选择器
 - 匹配所有拥有相同的class属性的值的元素
 - 匹配的是一个元素的集合
 - 语法: .class属性值
- 标签选择器
 - 匹配所有拥有指定标签的元素
 - 匹配的是一个元素的集合
 - 语法: 标签名
- * 选择器
 - 匹配当前文档中的所有元素
 - 语法: *

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>css基本选择器</title>
6      <style>
7          #d1{
8              width: 200px;
9              height: 100px;
10             background: pink;
11         }
12         .c1{
13             width: 200px;
14             height: 100px;
15             background: red;
16         }
17         p{
18             color: red;
19         }
20         *{
21             font-size: 20px;
22             margin: 0px;
23             padding: 0px;
24         }
25     </style>
26 </head>
27 <body>
28 <div id="d1"></div>
29 <div class="c1"></div>
30 <hr/>
31 <div class="c1"></div>
32 <hr/>
33 <div class="c1"></div>
34 <p>这是一个段落</p>
35 <p>这是一个段落</p>
36 <p>这是一个段落</p>
37 <p>这是一个段落</p>
38 <span>这是一个span</span><span>这是一个span</span><span>这是一个span</span><span>这是一个
   一个span</span>
39 </body>
40 </html>

```

3.混合选择器

不同的选择器可以通过不同的方式进行组合,匹配到合适的元素

- 多个选择器以逗号分隔
 - 匹配到这些选择器所匹配的所有元素
 - 类似于或者的意思
 - 语法: 选择器1, 选择器2, 选择器3...
- 多个选择器以空格分隔

- 先匹配到前面一个选择器所匹配到的所有元素
- 在用后一个选择器对已经匹配到的元素进行适配
- 匹配这些元素中匹配后一个选择器的所有元素
- 匹配所有的后代元素
- 所谓的后代元素,即包含子元素,子元素的子元素,子元素的子元素的子元素...
- 语法: 选择器1 选择器2 选择器3 ...
- 多个选择器以 > 分隔
 - 表示先匹配到前面的选择器所匹配的元素
 - 在这些元素的基础上通过后一个选择器来匹配这些元素的子元素中符合条件的元素
 - 类似于以空格分隔
 - 不包含后代元素,只匹配子元素
 - 语法: 选择器1>选择器2>选择器3...
- 多个选择器以 + 分隔
 - 先匹配符合第一个选择器的所有元素
 - 然后在这些元素的基础上找到紧跟着这些元素的后一个元素
 - 通过后一个选择器匹配这些元素
 - 匹配到符合后一个选择器规则的所有元素
 - 只能同级匹配,不考虑子标签
 - 语法: 选择器1+选择器2+选择器3...
- 多个选择器以 ~ 分隔
 - 先匹配符合的第一个选择器的所有元素
 - 在这些元素的后面找到所有符合第二个选择器的所有元素
 - 不需要紧跟着,只需要在后面就行

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>混合选择器</title>
6      <style>
7          #d1,#d3{
8              width: 100px;
9              height: 100px;
10             background: red;
11         }
12         .c1 .c2{
13             color: red;
14         }
15         #d4>.c3{
16             color: green;
17         }
18         #d5{
19             color:red;
20         }
21         .c5+span{
22             color: red;
23         }
24         #d6~span{
25             color:green;
26         }
27     </style>
28 </head>
29 <body>
30 <div id="d1">这是第一个div</div>
31 <hr/>
32 <div id="d2">这是第二个div</div>
33 <hr/>
34 <div id="d3">这是第三个div</div>
35
36 <div class="c1">这是div
37     <div class="c2">这是c1中的c2</div>
38     <div class="c2">这是c1中的c2</div>
39 </div>
40 <div class="c2"></div>
41 <div class="c1">这是div
42     <div class="c2">这是c1中的c2</div>
43     <div class="c2">这是c1中的c2
44         <div>这是c1中的c2中的c2</div>
45     </div>
46 </div>
47
48 <div id="d4">
49     <div class="c3">这是d4中的c3</div>
50     <div class="c4">
51         <div class="c3">这是d4中的c4中的c3</div>
52     </div>
53 </div>
```

```
54
55 <div id="d5">
56     <div>这是d5中的div</div>
57 </div>
58
59
60 <div class="c5">这是c5</div>
61 <span>这是c5后面的span</span>
62 <div class="c5">这是c5</div>
63 <a href="">超链接</a>
64 <span>这是c5后面的超链接后面的span</span>
65 <div class="c5">这是c5</div>
66 <span>这是c5后面的span</span>
67
68 <div id="d6">这是d6</div>
69
70 <span>这是span</span>
71 <span>这是span</span>
72 <span>这是span</span>
73 <span>这是span</span>
74
75
76 </body>
77 </html>
```

4.css伪类

4-1 伪类的类型

- `:link`
 - 表示未访问时的样式
- `:visited`
 - 表示已访问过后的样式
- `:hover`
 - 表示鼠标悬停在元素上的样式
- `:active`
 - 表示鼠标在元素上按下去尚未松开时的样式

4-2 伪类的限制

- `hover`必须在`link`与`visited`之后
- `active`必须在`hover`之后
- 大小写不敏感

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>css伪类</title>
6      <style>
7          a:hover{
8              color: gray;
9          }
10         a:active{
11             color:red;
12         }
13         a:link{
14             color: blue;
15         }
16         a:visited{
17             color:purple;
18         }
19
20         div{
21             width: 200px;
22             height: 100px;
23             background-color: blue;
24         }
25         div:link{
26             background-color: blue;
27         }
28         div:visited{
29             background-color: purple;
30         }
31         div:hover{
32             background-color: gray;
33         }
34         div:active{
35             background-color: red;
36         }
37     </style>
38 </head>
39 <body>
40 <a href="#">超链接</a>
41 <div>这是一个div</div>
42 </body>
43 </html>
```

5.css样式继承

当子元素没有指定样式且其父元素存在了样式

如果子元素本身存在样式,则使用自身的样式,不再使用父元素的


```
1 <div style="color: red;">
2   这是一个div
3   <span style="color: green;">这是div中的span</span><br/>
4   <a>这是div中的超链接</a>
5 </div>
```

6.选择器的优先级

此处只考虑基本选择器

id选择>class选择器>标签选择器

也可以通过 `!important` 破坏优先级

四.常见css样式

1.颜色

color

1-1 通过英文单词来控制颜色

- red
- green
- blue
- pink
- gray
- ...

1-2 通过RGB色系来控制颜色

Red 、 Green 、 Blue

通过红绿蓝三原色进行调色

每一种颜色的取值范围:0-255(十进制)

十六进制:00-ff

十六进制下,最终的RGB色系的值为: `#RRGGBB`

其中RR、GG、BB分别表示红色、绿色、蓝色的程度

取值的大小表示的就是对应颜色的深浅度

其中00表示颜色最浅,偏向黑色

ff表示颜色最深,偏向白色

- 白色: `#ffffff`
- 黑色: `#000000`
- 红色: `#ff0000`
- 绿色: `#00ff00`
- 蓝色: `#0000ff`

1-3 通过rgb函数来控制颜色以及透明度

- rgb(r,g,b)
 - 其中三个参数表示的是红绿蓝三个颜色的深浅度
 - 使用的是十进制
 - 取值范围:0-255
- rgba(r,g,b,a)
 - 前面三个参数表示的是红绿蓝三个颜色的深浅度
 - a:表示的是透明度
 - 取值范围:0-1
 - 0表示完全透明
 - 1表示完全不透明

```
1  div{
2      /*color:#0000ff;*/
3      /*color:rgb(255,0,0);*/
4      color:rgba(255,0,0,0.5);
5  }
6  #d2{
7      color:rgba(255,0,0,1);
8  }
```

2.字体

- font-size
 - 设置字体的大小
- font-style
 - 设置字体的样式
 - italic
- font-weight
 - 设置粗体
 - bold
- font-family
 - 设置字体的类型
 - 宋体、楷体、

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>字体</title>
6      <style>
7          #d1{
8              font-size: 14px;
9              font-style: italic;
10             font-weight: bold;
11             font-family: 宋体;
12         }
13         #d2{
14             font-size: 20px;
15             font-weight: bolder;
16             font-family: 楷体;
17         }
18         #d3{
19             font-weight: lighter;
20             font-family: 黑体;
21         }
22     </style>
23 </head>
24 <body>
25 <div id="d1">这是一个div</div>
26 <div id="d2">这是一个div</div>
27 <div id="d3">这是一个div</div>
28 </body>
29 </html>
```

3.背景

属性	描述
background-color	设置背景色
background-image	设置背景图
background-repeat	设置背景图片的重复方式
background-position	设置背景图片所在位置
background-attachment	设置背景图片是否跟随滚动
background	设置跟背景相关的一切属性

3-1 背景色

语法: background-color:颜色值;

3-2 背景图

语法: background-image:url('图片的路径')

当背景色与背景图片同时存在时,背景色失效

如果背景中存在背景图片与背景色,但是背景图没有全部覆盖时

在背景图没有覆盖到的区域会使用背景色进行填充

- **background-repeat:**设置背景图片的重复方式
 - **repeat:**完全重复,默认值
 - **no-repeat:**不重复
 - **repeat-x:**x轴重复
 - **repeat-y:**y轴重复
- **background-position:**设置背景图所在的方位
 - 只有背景图没有全覆盖时使用,值有5个
 - **top:**上方
 - **bottom:**下方
 - **center:**正中间
 - **left:**左边
 - **right:**右边
- **background-attachment:**设置背景图是否跟随滚动
 - **scroll:**默认值,跟随滚动
 - **fixed:**固定不动

3-3 background

在css中,遇到类似于xxx-yyy的用法的时候

绝大部分情况下:都表示对于xxx的细节进行设置

我们可以直接使用xxx样式进行设置,多个样式值之间以空格分隔

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>背景</title>
6     <style>
7         /*#d1{*/
8             /*width: 200px;*/
9             /*height: 200px;*/
10            /*background-image: url('../images/heihei.gif');*/
11            /*background-color: #ffccd8;*/
12            /*background-repeat: no-repeat;*/
13            /*background-position: bottom;*/
14        /*}*/
15        #d1{
16            width: 200px;
17            height: 200px;
18            background:url("../images/heihei.gif") #ffccd8 no-repeat bottom;
19        }
20        /*#d2{*/
21            /*width:100%;*/
22            /*height: 1400px;*/
23            /*background-image: url('../images/4.jpg');*/
24            /*background-attachment: fixed;*/
25        /*}*/
26        #d2{
27            width:100%;
28            height: 1400px;
29            background:url("../images/4.jpg") no-repeat fixed;
30        }
31    </style>
32 </head>
33 <body>
34 <div id="d1"></div>
35 <div id="d2"></div>
36 </body>
37 </html>
```

4.文本

属性	描述
color	设置文本的颜色
line-height	设置文本的行高
text-align	设置文本的对齐方式
text-indent	设置文本首行缩进,值是一个像素单位
text-decoration	设置文本的装饰
word-spacing	设置单词的间距,只对英文生效,每一个单词之间的距离
letter-spacing	设置字符的间距,每一个字母之间的距离,对中文生效
text-transform	设置大小写

4-1 line-height

设置行高

其值有以下几种方式

- 纯数字
 - 表示最终的行高值为:当前字体大小*该数字
- 固定值
 - 20px
 - 表示最终的行高值为:当前设置的固定值
- 百分比
 - 表示最终的行高值为:当前字体大小*百分比

4-2 text-align

设置文本对齐方式

值有三种:left,right,center

4-3 text-decoration

设置文本的装饰

本质上就是在文本的某个位置加上一条线

- underline
 - 在文本的下方设置一条线
- overline
 - 在文本的上方设置一条线
- line-through
 - 设置穿过文本的一条线

4-4 text-transform

设置大小写

- capitalize
 - 设置首字母大写
- uppercase
 - 设置都大写
- lowercase
 - 设置都小写

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>文本</title>
6      <style>
7          div{
8              color:red;
9              line-height: 40px;
10             /*text-align:center;*/
11             text-indent: 20px;
12         }
13         #d1{
14             text-decoration: underline;
15         }
16         #d2{
17             text-decoration: overline;
18         }
19         #d3{
20             text-decoration: line-through;
21         }
22         p{
23             word-spacing: 10px;
24             /*letter-spacing: 10px;*/
25             text-transform: capitalize;
26         }
27     </style>
28 </head>
29 <body>
30 <div id="d1">这是第1个div</div>
31 <div id="d2">这是第2个div</div>
32 <div id="d3">这是第3个div</div>
33
34 <p>this is a transform</p>
35 <p>这是中文</p>
36
37 </body>
38 </html>
```

5.元素的隐藏与显示

5-1 display

- none
 - 隐藏元素,隐藏后的元素在文档中不占用任何空间
- block
 - 显示为块级元素
 - 块级元素:自带换行的元素
- inline
 - 显示为行级元素
 - 行级元素:不自带换行的元素,元素可以共占一行


```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>display</title>
6      <style>
7          #ul{
8              display: block;
9          }
10         div{
11             display: inline;
12         }
13         span{
14             display: block;
15         }
16     </style>
17 </head>
18 <body>
19 <h2>家用电器</h2>
20 <ul id="u1">
21     <li>电视机</li>
22     <li>冰箱</li>
23     <li>空调</li>
24     <li>洗衣机</li>
25 </ul>
26 <h2>电子产品</h2>
27 <ul>
28     <li>手机</li>
29     <li>ipad</li>
30     <li>电子表</li>
31     <li>收音机</li>
32 </ul>
33 <div>第1个div</div>
34 <div>第2个div</div>
35 <div>第3个div</div>
36 <hr/>
37 <span>第1个span</span><span>第2个span</span><span>第3个span</span>
38 </body>
39 </html>

```

5-2 visibility

- hidden
 - 隐藏元素,隐藏后的元素仍然占用文档的空间
- visible
 - 显示元素

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>visibility</title>
6      <style>
7          div{
8              width: 200px;
9              height: 100px;
10         }
11         #d1{
12             visibility: hidden;
13             /*display: none;*/
14         }
15     </style>
16 </head>
17 <body>
18 <div id="d1">这是一个div</div>
19 <div>这是一个div</div>
20 <div>这是一个div</div>
21 </body>
22 </html>

```

5-2 display与visibility的区别

当元素隐藏的时候

display设置的元素不占空间

visibility设置的元素仍然占用空间

6.浮动

float

可以定义元素向左或者向右浮动

浮动的元素会脱离文档流(可以理解为进入了另一个次元空间)

当浮动的元素触碰到另一个元素的边框或者屏幕的边界的时候会停止浮动

A、B

浮动的值有两种

- left: 向左浮动
- right: 向右浮动

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>float</title>
6      <style>
7          div{
8              width: 200px;
9              height: 100px;
10         }
11         #d1{
12             background: red;
13             float: left;
14         }
15         #d2{
16             background: blue;
17             float:left;
18         }
19         #d3{
20             background: green;
21             float:left;
22         }
23     </style>
24 </head>
25 <body>
26 <div id="d1"></div>
27 <div id="d2"></div>
28 <div id="d3"></div>
29 </body>
30 </html>
```

五.定位

1.什么是定位

position

将一个元素放置到一个静态的、相对的、绝对的、固定的位置中

用于定位元素的位置

2.定位的类型

- 静态定位
 - static
 - 默认值
 - 按照顺序从左往右、从上往下进行一次排列
 - 静态定位无法进行偏移
- 相对定位
 - relative

- 先按照静态定位进行排列
- 然后通过指定偏移量对指定的元素进行偏移
- 通过 `left`、`top`、`right`、`bottom` 四个属性进行偏移量的设置
- 绝对定位
 - `absolute`
 - 参照父元素进行定位
 - 使用绝对定位的元素会从文档流中删除(可以理解为进入了另一次次元空间)
 - 简单来讲,当某个元素设置了绝对定位,则其他元素在进行定位的时候会忽略该元素
 - 此时容易存在多个元素重叠在一起
 - 可以通过`z-index`属性来控制堆叠顺序
 - 该属性的值是一个数字,值大的在外面
- 固定定位
 - `fixed`
 - 参考屏幕进行定位
 - 其他与绝对定位完全一致
 - 使用绝对定位的元素会从文档流中删除(可以理解为进入了另一次次元空间)
 - 简单来讲,当某个元素设置了绝对定位,则其他元素在进行定位的时候会忽略该元素
 - 此时容易存在多个元素重叠在一起
 - 可以通过`z-index`属性来控制堆叠顺序
 - 该属性的值是一个数字,值大的在外面

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <style>
7          #a{
8              width: 300px;
9              height: 300px;
10             background: red;
11             position: absolute;
12         }
13         #b{
14             width: 500px;
15             height: 500px;
16             position: fixed;
17             left:100px;
18             background: green;
19         }
20         #b1{
21             width:200px;
22             height: 200px;
23             background: gray;
24             position: fixed;
25             left:50px;
26             /*此时发现b1是距离屏幕50px*/
27         }
28         #b2{
29             width: 200px;
30             height: 200px;
31             background: pink;
32             position: absolute;
33             left: 50px;
34             /*此时发现b2是距离父元素50px*/
35         }
36     </style>
37 </head>
38 <body>
39 <div id="a">a</div>
40 <div id="b">
41     <div id="b1">b1</div>
42     <div id="b2">b2</div>
43 </div>
44 </body>
45 </html>
```

六.盒子模型

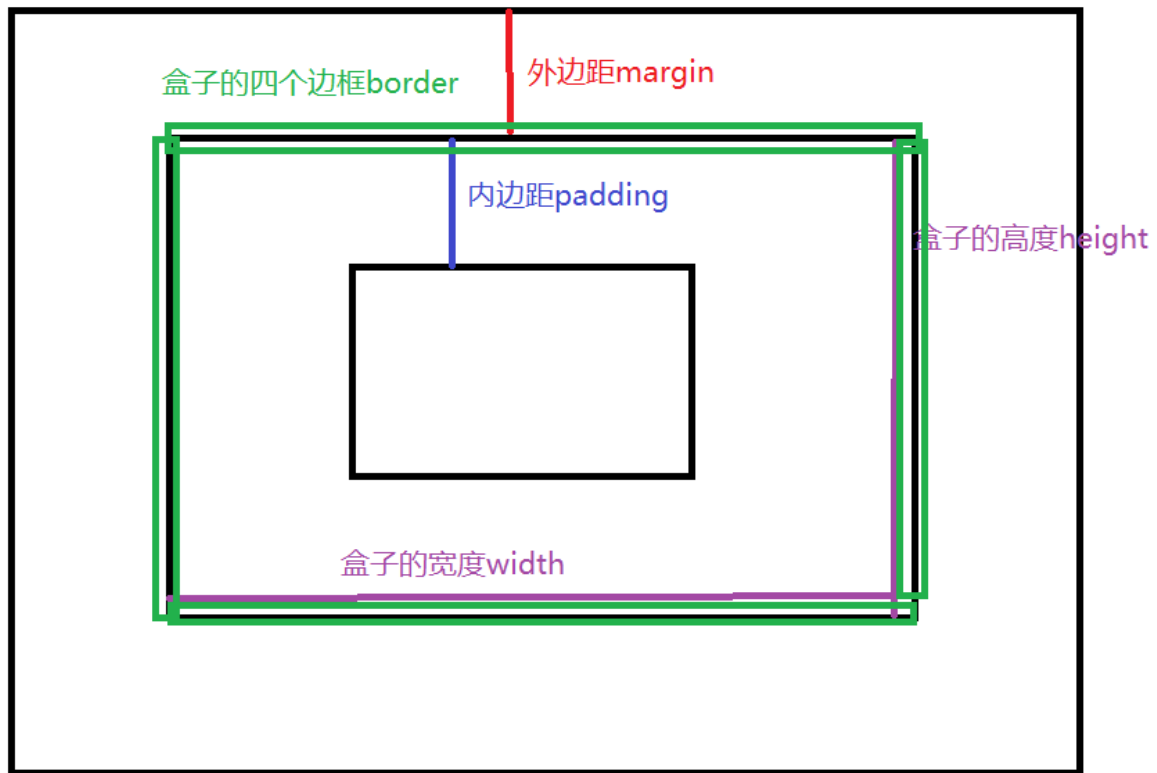
1.介绍盒子

盒子模型一共包含三个盒子

最外面的盒子:表示的是父元素

中间的盒子:表示的是当前元素

最里面的盒子:表示的是当前元素的标签体内容



2.border

border是对于盒子的边框进行样式的设计

可以单独对边框的某个方向进行设置

设置盒子的边框厚度会改变原有盒子的大小

例如: `border-top`, `border-left`, `border-right`, `border-bottom`

边框的子属性

- **style**: 设置盒子边框的样式
 - **solid**: 实线边框
 - **dotted**: 点状边框
 - **dashed**: 虚线边框
 - **double**: 双线边框
 - 其他请参考W3C
- **color**: 设置盒子的边框颜色
- **width**: 设置盒子的边框厚度
- **radius**: 设置盒子的边框弧度

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>盒子模型</title>
6      <style>
7          #d1{
8              width: 200px;
9              height: 100px;
10             border-top-style: solid;
11             border-bottom-style: dotted;
12             border-left-style: dashed;
13             border-right-style: double;
14         }
15         #d2{
16             width: 200px;
17             height: 100px;
18             border-style: solid;
19             /*border-top-color: red;*/
20             /*border-bottom-color: green;*/
21             /*border-left-color: blue;*/
22             /*border-right-color: yellow;*/
23             /*border-color:red yellow green blue;*/
24             /*border-color:red yellow;*/
25             border-color:red yellow green;
26         }
27         #d3{
28             width: 200px;
29             height: 100px;
30             border-style: solid;
31             border-width: 10px;
32         }
33         #d4{
34             width: 200px;
35             height: 100px;
36             border-style: solid;
37             border-radius:100px;
38         }
39     </style>
40 </head>
41 <body>
42 <div id="d1">这是一个div</div>
43 <hr/>
44 <div id="d2">这是一个div</div>
45 <hr/>
46 <div id="d3">这是一个div</div>
47 <hr/>
48 <div id="d4">这是一个div</div>
49 </body>
50 </html>
```

3.盒子的属性参数

适用于 `border`, `margin`, `padding`

- `top`: 上
- `bottom`: 下
- `left`: 左
- `right`: 右

可以单独对某个方向进行设置

也可以通过给予属性传递不同数量的参数来指定不同方向的样式

每一个子属性的参数的值有四种

- 一个参数
 - 表示上下左右四个方向的值都是该参数的值
- 两个参数
 - 上下、左右
- 三个参数
 - 上、左右、下
- 四个参数
 - 上、右、下、左

4.margin

外边距

表示当前盒子与父元素的边框的距离

当上下冲突的时候,优先选择是上

当左右冲突的时候,优先选择的左

当设置盒子的外边距时,不会改变盒子的大小

5.padding

内边距

表示当前盒子与盒子中元素的距离

当上下冲突的时候,优先选择是上

当左右冲突的时候,优先选择的左

当设置了盒子的内边距时,会改变盒子的大小

6.最终盒子大小

最终盒子的高度=原来的高度+上下盒子的边框厚度+上下盒子的内边距

最终盒子的宽度=原来的宽度+左右盒子的边框厚度+左右盒子的内边距


```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>盒子模型</title>
6      <style>
7          *{
8              margin: 0px;
9              padding: 0px;
10         }
11         #d1{
12             width: 200px;
13             height: 100px;
14             background: red;
15             /*margin-left:100px;*/
16             /*margin-top:100px;*/
17             border:solid pink 1px;
18             /*margin-left:50px;*/
19             /*margin-top:50px;*/
20             padding:20px;
21             position:absolute;
22         }
23         #d2{
24             width: 200px;
25             height: 100px;
26             background: #ffccd8;
27             position: absolute;
28             left:242px;
29             top:142px;
30         }
31     </style>
32 </head>
33 <body>
34 <div id="d1">这是一个div</div>
35 <div id="d2">这是第二个div</div>
36 </body>
37 </html>
```