

CSS

cascading style sheet 层叠样式表

引入方式

1. 行间样式

```
<div style="width:100px;height:100px;background-color:red;">  
</div>
```

2. 页面级css

```
<!DOCTYPE html>  
<html>  
<head>  
  <title> </title>  
  <style type="text/css">  
    div{  
      width:100px;  
      height:100px;  
      background-color:red;  
    }  
  </style>  
</head>  
<body>  
  
</body>  
</html>
```

3. 外部css文件

```
<!DOCTYPE html>  
<html>  
<head>  
  <title> </title>  
  <link rel="stylesheet" type="text/css" href="文件路径">  
</head>  
<body>  
  
</body>  
</html>
```

计算机中同步是不一起做的

异步是同时做的和实际生活相反

选择器

1.id选择器

```
<div id="only">123</div>
```

```
#only{    //要一一对应
```

```
}
```

2.class选择器

```
<div class="demo demo1">123</div>
```

```
<div class="demo">234</div>
```

```
.demo{    //可以多对多
```

```
}
```

```
.demo1{
```

3.标签选择器

```
<div></div>
```

```
div{
```

```
    所有的div都
```

```
}
```

4.通配符选择器

```
<div>123</div>
```

```
<span>234</span>
```

```
<strong>456</strong>
```

```
*{
```

```
    所有的标签都
```

```
}
```

5.属性选择器

```
<div id="demo" class="">123</div>
```

```
[id="demo"]{
```

```
    所有有id属性的标签都
```

```
}
```

6.父子选择器

```
    <div class="demo">
```

```
        <span class="test">
```

```
            123
```

```
        </span>
```

```
    </div>
```

```
div span{
```

```
}
```

```
或者
```

```
.demo .text{
```

```
}
```

只要呈父子关系 表示的是该标签下的所有的子标签

7.直接子元素选择器

```
<div>  
  <span>
```

```
  </span>  
</div>
```

```
div > span{  
  表示的是div仅下一层的span  
}
```

8.并列选择器

```
<div>1</div>  
<div class="demo">2</div>  
<p class="demo">3</p>  
div .demo{  
  
}
```

9.分组选择器

```
<em>1</em>  
<strong>2</strong>  
<span>3</span>  
em, strong, span{  
  用来写同样样式的不同标签  
}
```

浏览器遍历标签：自右向左

优先级

! important(写在特定的属性上就会拥有最高权限) **infinity** > 行间样式

1000 > **id** **100** > **class** = 属性选择器 = 伪类 **10** > **div** (标签) = 伪元

素 **1** > ***** (通配符) **0**

权重进制256

权重

属性

font-size

字体大小默认16px设置的是字体的高

font-weight

字体粗细 bolder bold normal lighter 或者数字表示 100-900(整百数)

font-style

值italic 斜体 normal

font-family

设置字体 常用字体arial

color

设置字体颜色rgb(0~255,0~255,0~255) #000000 - #ffffff

文字在容器中垂直居中要让文字大小等于height

border: 设置盒子边框

border-width

border-style dotted虚线 dashed条状虚线

border-color transparent(透明色)

border-left-width(同理)

border 1px solid(边框颜色) black;

text-align 文本对齐方式

left(默认) center right

text-indent 首行缩进

2em 缩进两个文本的距离

text-decoration:

line-through; 相当于 标签 中划线

underline 下划线

overline 上划线

cursor 鼠标样式

pointer 鼠标移上去变小手

help 变小问号

先选中a标签

a:hover{ 鼠标移动上变化

}

opacity

透明度0-1

行级元素 inline内联元素

特点：内容界定元素所占位置

不可通过css改变宽高

span strong em a del

块级元素 block

特点：独占一行

可以通过css改变宽高

div p ul li ol form address

行级块元素 inline-block

特点：内容决定大小

可以改宽高

img

带有**inline**属性的元素都文本类特点，叫做**文本类元素**，

例如：空格作为文字分隔符，对文本类元素也起到相同作用

文本之间底对齐，所以img标签和文本在一起也会底对齐

如果行级块元素里包含文字，外部文字就不和该元素对齐，而和里面文字底对齐

vertical-align: 1px; middle; 调一行元素的对齐线

通过display属性设置元素行级块级

凡是带有inline属性的都有文字特性

先定义功能后选定属性

class可以接多个名字

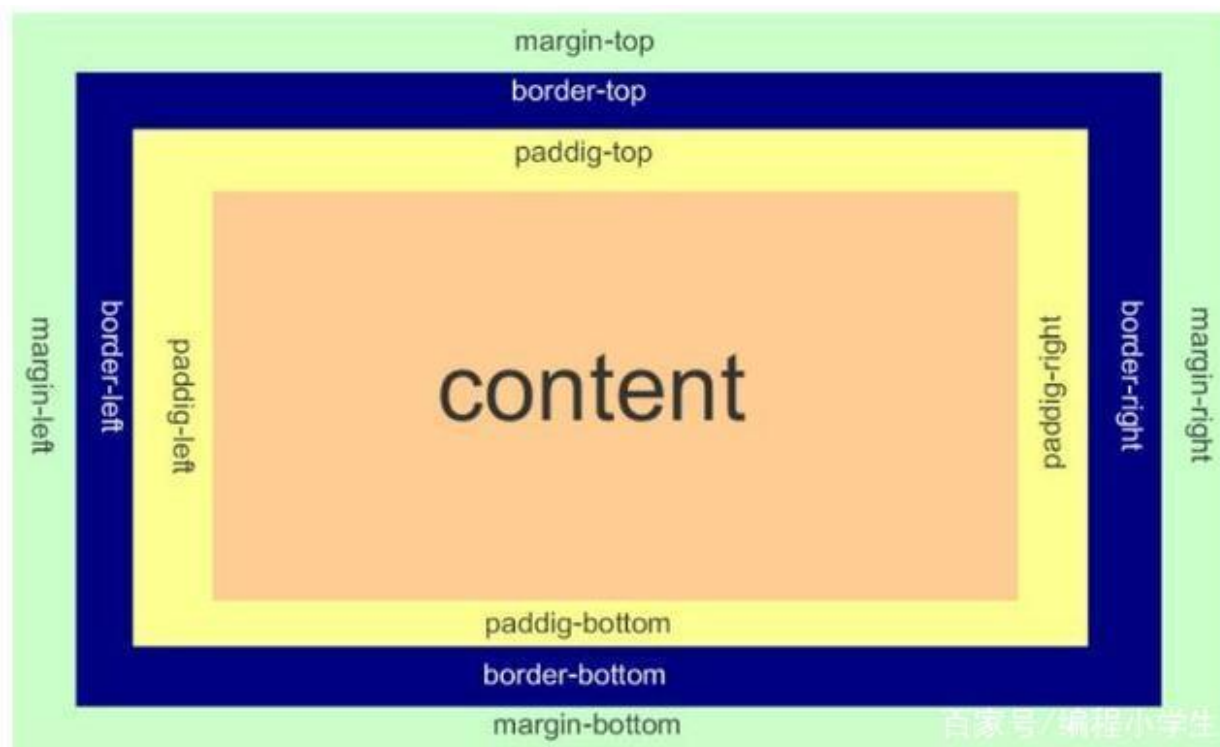
图片之间有间隙，要去除空格

可以自定义标签

使用通配符选择器来初始化margin padding

盒子模型

content = width + height



padding

padding-top padding-left

margin 上 右 下 左 (顺时针)

上 左右 下 (三个值)

上下 左右 (两个值)

margin 0 auto; 上下为零 左右自适应

body有个天生的margin 8px

定位

position

absolute

left

top

right

bottom

特点：脱离原来位置进行定位

相对于最近的有定位的父级定位 (相对边) , 如果没有, 相对于文档定位 (相对边) 一般情况下

父级指的是上一层的div

relative

特点：保留原来位置定位

相对于原来的位置的相对边

fixed

特点：相对窗口定位, (广告)

static(默认值)

margin是相对定位点的边距相对父级的边框

垂直方向的margin 父子元素是一起的，两者取最大的值（margin塌陷，相当于没有棚，子级会以父级为边框）

bfc

block format context(解决margin塌陷问题)

方法：

1. position: absolute;
2. display: inline-block;
3. float: left/right;
4. overflow: hidden;

4.overflow: hidden;

溢出盒子部分隐藏

解决塌陷问题时最好不要大于盒子大小，

兄弟结构的垂直方向的margin是合并的

解决方法：

将其中一个或者所有标签多加一层，该层只变成bfc环境(其实最好不要)

浮动流

所有产生了浮动流的元素，块级元素看不到他们，（直接分层）

产生了bfc的元素和文本属性（带inline属性）的元素以及文本都能看到浮动元素

解决浮动流的方法

```
<div class="wrapper">
  <div class="content">1 </div>
  <div class="content">2 </div>
  <div class="content">3 </div>
  <p> </p>
</div>
.wrapper{
  border: 1px solid black;
}
.content{
  float: left;
```

```

color: rgb(0, 0, 0);
background-color: rgb(0, 0, 0);
width: 100px;
height: 100px;
}
p{
border-top: 0 solid green;
clear: both;//用此属性清除浮动，必须要是块级元素

}

```

不好，因为最好不要改变html的结构

伪元素

形式：

`成哥`

```

*{
margin: 0px;
padding: 0px;
}
span::before{
content:"henshuai"; //修改伪元素内容
display: inline-block;
}
span::after{

}

```

特点：

可以当元素操作，但是没有元素结构

伪元素是行级元素

解决浮动流的最终方法：

利用伪元素

```

<div class="wrapper">
  <div class="content">1</div>
  <div class="content">2</div>
  <div class="content">3</div>
</div>
.wrapper{
border: 1px solid black;
}
.content{

```



```
float: left;
color: rgb(0, 0, 0);
background-color: rgb(0, 0, 0);
width: 100px;
height: 100px;
}
.wrapper::after{
  content:"";
  clear: both;
  display:block;
}
```

总结:

触发了bfc元素，浮动元素文本元素都能解决浮动流问题

凡是使用属性position:absolute; float:right/left;

系统会在内部把元素转换为inline-block

文本垂直居中:

使用line-height= 盒子高度

文字溢出容器处理

1. 单行文本

<p> </p>

```
p{
  width:300px;
  height:20px;
  line-height:20px;
  border: 1px solid black;

  white-space: nowrap; //文本不换行
  overflow:hidden;    //溢出部分隐藏
  text-overflow:ellipsis; //溢出文本用.....展示
}
```

2. 多行文本

并没有方法实现，只能自己计算像素，手动打点
多行只做截断，多余部分overflow:hidden;

背景图片

```
<div></div>
```

```
div{
  width:200px;
  height:200px;
  border: 1px solid black;
  background-image:url(xxx.jpg); //设置背景图片
  background-size:200px 200px; //设置背景图片大小 宽 高
  如果不够铺满整个容器，就会多张图片重复平铺
  background-repeat:no-repeat; //设置为不重复出现
  repeat-x X轴重复平铺 repeat-yY轴重复平铺
  background-position: 100px 100px 在背景的 宽 高 处出现
  也可以填方位      center center 50% 50%
}
```

默认加载策略：

一旦网速不好，先加载HTML，忽略css js 正常情况下同时加载，逐句加载

图片代替文字

方法1

```
<a href="http:www.taobao.com"target="_blank">淘宝网</a>
```

```
a{
  text-decoration: none;
  display: inline-block;
  width: 190px;
  height: 90px;
  /* border: 1px solid black; */
  text-indent: 190px;
  white-space: nowrap;
  overflow: hidden;
  background-image:
url(https://img.alicdn.com/tfs/TB1_uT8a5ERMeJjSspiXXbZLFXa-143-59.png);
  background-size: 190px 90px;
}
```

方法2

```
<a href="http:www.taobao.com"target="_blank">淘宝网</a>
```

```
a{
  text-decoration: none;
  display: inline-block;
  width: 190px;
  height: 0px;
  padding-top: 90px;
  /* border: 1px solid black; */
}
```

```
overflow: hidden;
background-image:
url(https://img.alicdn.com/tfs/TB1_uT8a5ERMeJSSpiXXbZLFXa-143-59.png);
background-size: 190px 90px;
}
```

行级元素只能嵌套行级元素

块级元素可以嵌套所有元素

```
<p>
  <div> </div>
</p>
```

```
<p>
```

```
</p>
  <div> </div>
<p>
```

```
</p>
```

//行级元素被块级元素劈成了两半

a标签不能嵌套a标签

overflow-x:hidden //横向滚动条消失