# 02-CSS02-盒模型

## 1 文字行高

### 1.1 浏览器文字

浏览器默认文字大小：16px

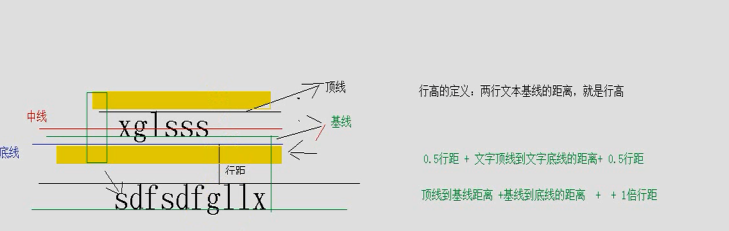
浏览器默认行高大小：18px

行高 = 文字大小 + 上间距 + 下间距

### 1.2 行高的作用

当行高值为父容器的高度时，可以让父容器中的文字垂直显示。

|  |
| --- |
| div{  height: 100px;  width: 300px;  font-size:20px;  line-height: 100px;  } |



### 1.3 行高单位

单独给一个元素设置行高：

|  |  |  |  |
| --- | --- | --- | --- |
| 行高单位 | 赋值 | 文字大小 | 行高值 |
| px | 20px | 20px | 20px |
| em | 2em | 20px | 40px |
| % | 120% | 20px | 24px |
| 不带单位 | 2 | 20px | 40px |

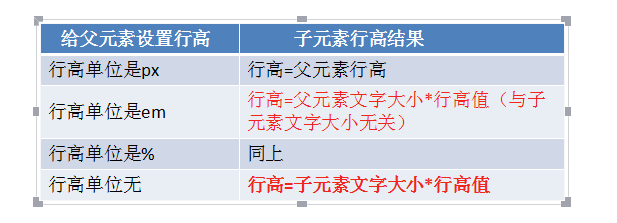
总结：当给一个独立的元素设置行高值的时候，除了以px为单位的行高值与文

字大小无关，其他都与文字大小有关（与文字大小的积）。

盒子嵌套，给父元素设置行高值，子元素的行高问题：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 行高单位 | 设置行高 | 父文字 | 子文字 | 行高 |
| Px | 20px | 20px | 30px | 20px |
| em | 2em | 20px | 30px | 40px |
| % | 120% | 20px | 30px | 24px |
| 不带单位 | 2 | 20px | 30px | 60px |

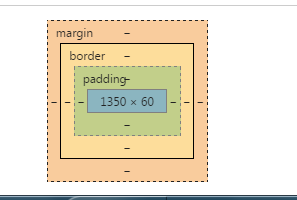
总结：行高可以继承。当父元素设置了行高值（不带单位除外），子元素的行高值都是父元素行高值乘以父元素文字大小。



## 2 盒模型

### 2.1 简介

盒模型主要用来网页布局，盒子包括：border(边框)、内边距（padding）、外边距（margin）。



### 2.2 边框border

☞border-width: 边框宽度

☞border-style: 边框样式

◆solid 实线边框

◆dotted 点线边框

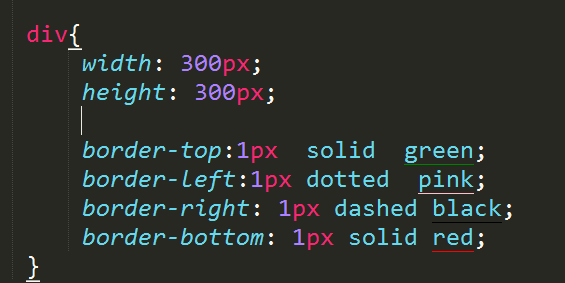
◆dashed 虚线边框

☞border-color: 边框颜色

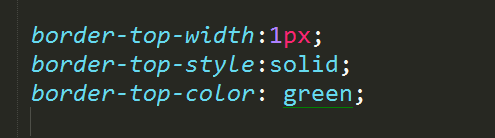
border写法一：



border写法二：



border写法三：



border属性联写：没有先后顺序限制，边框颜色、宽度可以不写

border: solid 1px red;

### 2.3 内边距padding

内边距：设置内容距离盒子边框之间的距离

padding-left: 左边距

padding-right: 右边距

padding-top: 上边距

padding-bottom: 下边距

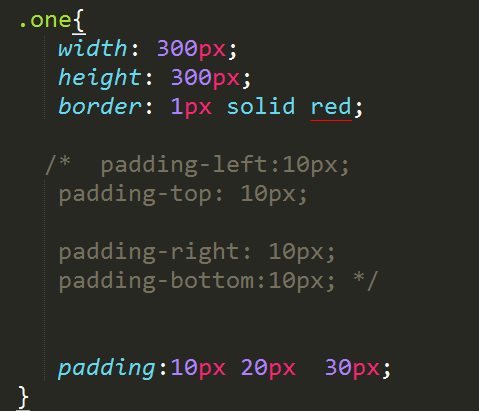
属性联写：

padding: 10px; 上，右，下，左的距离为10px

padding: 10px 20px; 上下10px 左右20px

padding: 10px 20px 30px 上10px 左右20px 下30px

padding: 10px 20px 30px 40px; 上， 右 ， 下， 左



### 2.4 盒子大小计算

边框和内边距可以影响盒子的大小；

宽度=内容宽度+左右边框+左右内边距；

注意：进行页面布局时，如果给盒子设置了内边距，对应的要将内容宽度或者高度减去相应的值。

继承的盒子在父盒子宽度范围内，padding值不会影响该盒子大小，即继承的宽度 padding不会挤开 。

### 2.5外边距margin

外边距设置盒子与盒子之间的距离。属性同padding。

注意：当两个盒子垂直显示的时候，外边距以设置的最大值为准。

行内元素只有左右margin，没有上下margin，同理padding也是

### 2.6 外边距塌陷

嵌套的盒子，直接给子盒子设置外边距

|  |
| --- |
| <**style**>  .**father**{  **width**: 500**px**;  **height**:500**px**;  **background-color**: **red**;  }  .**son**{  **width**: 200**px**;  **height**: 300**px**;  **background-color**: **green**;  **margin-top**: 50**px**; *<!--* **子盒子理应向下移动50像素，父子却一起塌陷** *-->* }  </**style**> </**head**> <**body**>  <**div class="father"**>  <**div class="son"**></**div**>  </**div**> |

解决方案：

◆父盒子设置边框

◆父盒子设置overflow:hidden（会触发bfc-格式化上下文）

### 2.7 margin与padding场景

在不考虑宽高时：

1. 需要使用背景图的时候必须用padding。
2. 会出现外边距合并或者margin塌陷的时候用padding。
3. 行内元素上下只能设置padding，不能设置margin。（行内高16px）
4. 看border，如果是a连接，看能不能让字体变色，或者显示小手。
5. 看需求。

## 3 浮动

### 3.1 文档流

元素默认的显示方式（如块级元素独占一行）就是标准流（文档流）。

### 3.2 浮动

浮动的目的是为了让块级元素能够在一行显示。浮动常用来解决文字环绕图片问题，也可用来制作导航栏、网页布局。

用法：float:left| right

特点：

☞设置了浮动的元素不占原来的位置（脱标）

☞可以让块级元素在一行上显示

☞浮动可以行内元素转化为行内块元素

注意：模式转换的过程中，能用display就用display

### 3.2 清除浮动

清除浮动不是删除浮动，而是清除浮动对布局的影响。

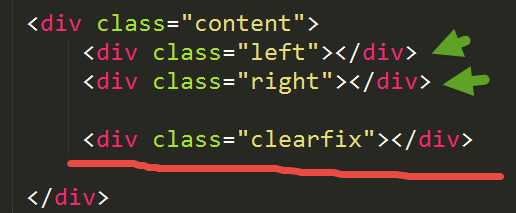
当子元素设置了浮动，父元素没有高度的时候，会造成页面布局混乱，这种情况下进行清除浮动。

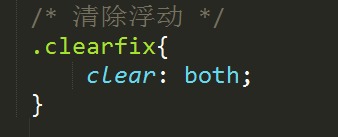
即解决了 父 盒子高度为0 的问题：

子盒子的内容需要撑开父盒子的大小，如果子盒子浮动，就无法撑开高度，这时候清除浮动既可以满足撑开父盒子大小的要求，又可以让子盒子保持浮动后的效果。

谁出问题给谁加一个clearfix类名，清除浮动的四种方式:

方式一：使用clear:left| right | both

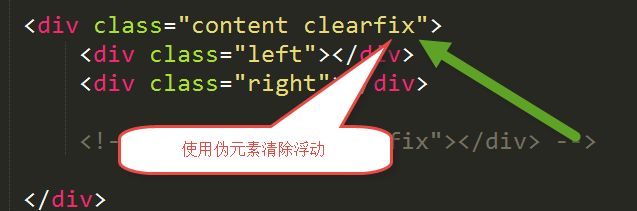




方式二：给父盒子设置overflow:hidden 使用此属性用来触发bfc清。

如果父盒子中有定位的元素，一般不推荐使用该种方式清除浮动，因为子盒子中的元素如果超出了父盒子的高度，超出部分会被切掉。

方式三：使用伪元素清除浮动

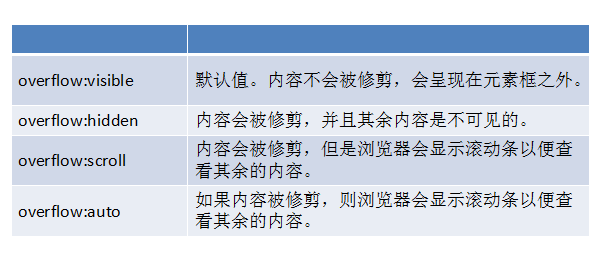




方式四：使用双伪元素清除浮动（淘宝、小米采用）

|  |
| --- |
| .clearfix:before,.clearfix:after{  display: table;  content: ""; } .clearfix:after {  clear: both; } .clearfix {  zoom: 1： } |

### 3.4 overflow的使用



## 4 定位

### 4.1 定位简介

方位名称：left， right ， top, bottom

本身浮动的不浮动: float:none; 此外还有left both right 等

本身定位的不定位: position: static; 如淘宝首页搜索框的效果

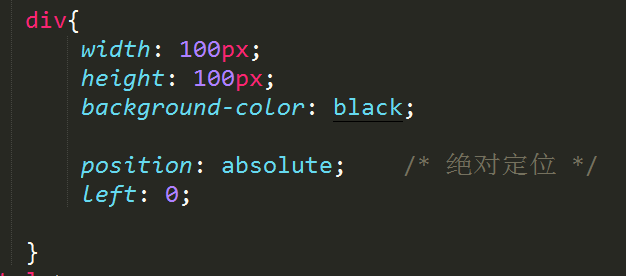
### 4.2 静态定位

用法：position:static 就是元素标准流的显示方式，静态定位约等于标准流。

注意：静态定位和相对定位占据位置。

### 4.3 绝对定位 absolute

用法： position:absolute;



特点：

☞给单独的元素设置绝对定位，以浏览器左上角（body）为基准

☞给行内元素设置绝对定位后，该元素转化为了行内块元素

☞给盒子设置了绝对定位，该盒子不占位置（脱标）

☞当盒子发生嵌套关系的时候：

父盒子没有定位，子盒子定位，以浏览器左上角为基准;

父盒子设置定位，子盒子设置定位，以父盒子左上角为基准。

注意：绝对定位和固定定位不占位置，设置绝对定位后，通过具体方位名称可以随便设置元素位置。

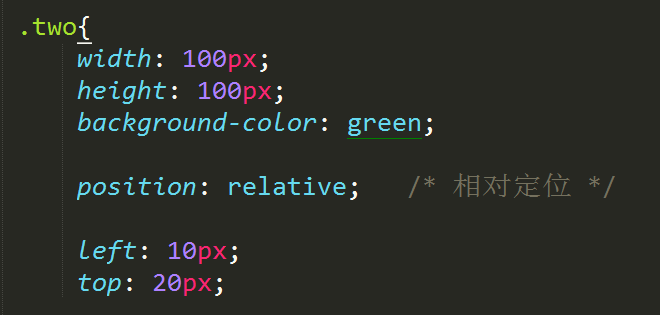
### 4.4 相对定位 relative

☞元素设置了相对定位后占原来的位置

☞设置相对定位以自己的位置为参照设置位置

☞相对定位不能进行元素的模式转换

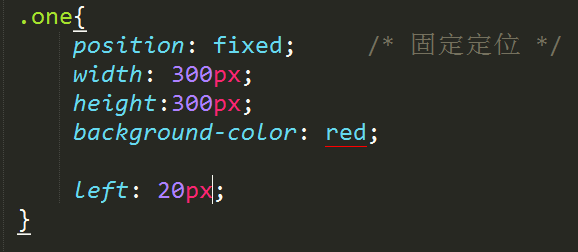
☞子元素设置绝对定位，父元素设置相对定位（子绝父相，也要看情况）



### 4.5 固定定位 fixed

☞固定定位不占位置（脱标）

☞将行内元素转化为行内块元素



## 5 知识点

### 5.1 盒子居中

标准文档流盒子居中：margin:0 auto;

脱离文档流盒子居中：left:50% margin-left:-100px

原因：脱标后margin:0 auto不起作用，直接书写left数值会影响适配。

解析：

首先子盒子走父盒子宽度的一半

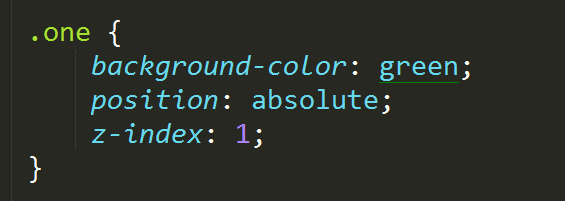
然后子盒子往回走自己宽度的一半（-100px代表的是方向）

### 5.2 定位盒子的层级关系

后定位的盒子的层级要高于前面定位的盒子的层级，使用z-index设置定位盒子之间的层级关系，z-index可以取负数。

只有定位的盒子（除了static）才有z-index；

如果都是绝对定位，他们默认的z-index都是0；



### 5.3 标签嵌套规范

☞段落标签中不能再包含段落标签

☞段落标签不能包含标题标签

☞段落标签中不能包含div,ul ,ol......

☞行内元素最好不要包含块级元素。行内元素可以包含行内元素。

### 5.4规避脱标流

☞在网页布局中，布局优先标准流，然后考虑浮动和定位。

☞如果要让一个元素实现模式转化： 优先使用display

☞如果想让一个元素移动到另一侧： margin-left: auto;

### 5.5 vertical-align

vertical-align用来设置元素的垂直对其方式，默认值是baseline，middle可以实现文字图片垂直居中显示，使用vertical-align起作用的元素只有，行内块元素和table。

|  |
| --- |
| <style type="text/css">  img.top {vertical-align:text-top}  img.bottom {vertical-align:text-bottom}  </style>  </head>  <body>  <p>  这是一幅<img class="top" border="0" src="/i/eg\_cute.gif" />位于段落中的图像。  </p>  <p>  这是一幅<img class="bottom" border="0" src="/i/eg\_cute.gif" />位于段落中的图像。  </p> |

### 5.6 CSS可见性

☞overflow:hidden 将容器中超出部分隐藏

☞visibility:hidden; 将页面中的元素隐藏

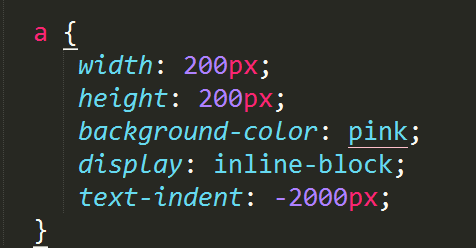
☞display: none; 将页面中的元素进行隐藏，不占位置（block为显示）

visibility:hidden 隐藏盒子，占据位置

opacity:0 隐藏盒子，占据位置

position/top/left/...-999px 隐藏盒子，而且占位置。

☞使用text-indent可以实现内容移除效果



### 5.7 favicon 图标

<link rel="shortcut icon" href="[favicon.ico](view-source:file:///C:/Users/andy/Desktop/%E4%BA%AC%E4%B8%9C%E7%BD%91%E7%AB%99/favicon.ico)" />

### 5.8 鼠标样式

cursor: pointer 鼠标变成小手

cursor: default; 小白

cursor : move; 移动

cursor : text ; 文本输入

### 5.9 圆角矩形border-radius

border-radius: 7px 7px 7px 0;

Border-radius: 左上 右上 右下 左下； 顺时针

### 5.10背景半透明

CSS3中的背景半透明

background: rgba(0,0,0,0.5);

opacity: 0.5; 让盒子半透明 里面的内容也半透明

### 5.11 文字居中与大小

文字居中的办法：

1. 给文字设置标签，然后用定位。（比较精准，但是麻烦）
2. Text-align: center; (文本居中。方便，但是不能精准控制)
3. 给定两边Padding或者margin。（比较方便，精准控制）

文字字体的设置：

font-weight: normal; 加粗变正常

font-style: normal; 倾斜变正常

text-decoration: none; 下划线删除线变正常

outline-style: none; 去除蓝色外边框

resize: none; 禁止文本框拖拽

### 5.12 JS关闭a链接跳转

<a href="javascript:;">**1111111111**</a>

<a href="javascript:void(0)">**1111**</a>

### 5.13 权重问题

left比right，权重高。有left又有right的时候，执行left的值。

top比bottom，权重高。有top又有bottom的时候，执行top的值。

### 5.14 定位问题

定位中，static和relative两个定位不能给行内元素设置宽高，另外两个可以。

## 6 精灵图

精灵图是一种处理网页背景图像的方式（背景图片），可以给一个元素将精灵图设置为背景图片，使用坐标找到指定的背景图片，减少请求压力

☞浏览器坐标系：圆点往右为正，圆点往上移动为负

## 7 css 初始化

@charset "UTF-8";

/\*css 初始化 \*/

html, body, ul, li, ol, dl, dd, dt, p, h1, h2, h3, h4, h5, h6, form, fieldset, legend, img { margin:0; padding:0; }

fieldset, img,input,button { border:none; padding:0;margin:0;outline-style:none; }

ul, ol { list-style:none; }

input { padding-top:0; padding-bottom:0; font-family: "SimSun","宋体";}

select, input { vertical-align:middle;}

select, input, textarea { font-size:12px; margin:0; }

textarea { resize:none; } /\*防止拖动\*/

img {border:0;  vertical-align:middle; } /\* 去掉图片低测默认的3像素空白缝隙\*/

table { border-collapse:collapse; }

body {

font:12px/150% Arial,Verdana,"\5b8b\4f53";

color:#666;

background:#fff

}

.clearfix:before,.clearfix:after {

content:"";

display:table;

}

.clearfix:after{clear:both;}

.clearfix{

\*zoom:1;/\*IE/7/6\*/

}

a {color:#666; text-decoration:none; }

a:hover{color:#C81623;}

h1,h2,h3,h4,h5,h6 {text-decoration:none;font-weight:normal;font-size:100%;}

s,i,em{font-style:normal;text-decoration:none;}

.col-red{color: #C81623!important;}

/\*公共类\*/

.w { /\*版心 提取 \*/

width: 1210px;margin:0 auto;

}

.fl {

float:left

}

.fr {

float:right

}

.al {

text-align:left

}

.ac {

text-align:center

}

.ar {

text-align:right

}

.hide {

display:none

}