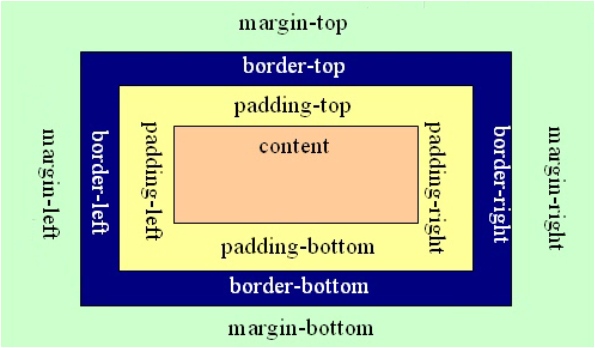
|  |
| --- |
|  |
|  |

# CSS盒子模型

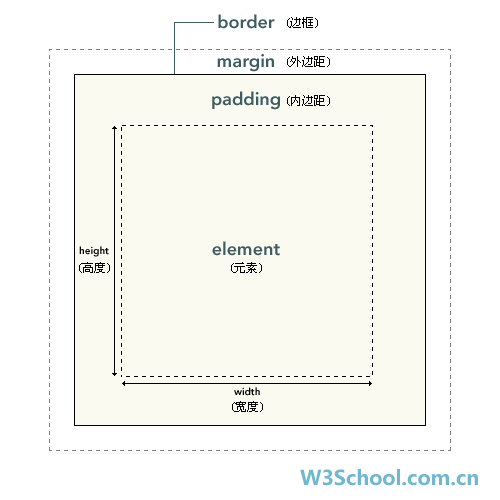
## 介绍

　　CSS中， Box Model叫盒子模型（或框模型），Box Model规定了处理元素内容（element content）、内边距（padding）、边框（border） 和 外边距（margin） 的方式。在HTML文档中，每个元素（element）都有盒子模型，所以说在Web世界里（特别是页面布局），Box Model无处不在。下面是Box Model的图示：



　　　　　　　　　　　　　　　　　　Box-Model 1

　　说明：上图中，由内而外依次是元素内容（content）、内边矩（padding-top、padding-right、padding- bottom、padding-left）、边框（border-top、border-right、border-bottom、border- left）和外边距（marging-top、margin-right、margin-bottom、margin-left）。下图是W3School的Box Model 图解：



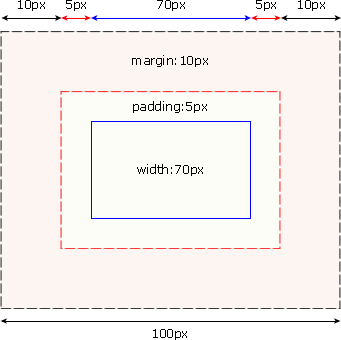
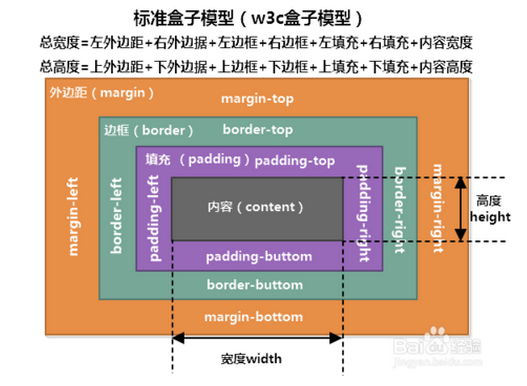
　　说明：

　　1.在上图中，元素框的最内部分是实际的内容（element）；直接包围内容的是内边距（padding），内边距呈现了元素的背景（background）；内边距的边缘是边框（border）；边框以外是外边距（margin），外边距默认是透明的，因此不会遮挡其后的任何元素（其实元素的margin就是其所在父元素的padding）。元素的背景应用于由内容和内边距、边框组成的区域。

　　2.内边距、边框和外边距都是可选的，默认值是零。但是，许多元素将由用户代理样式表（比如：浏览器）设置外边距和内边距。可以通过将元素的 margin 和 padding 设置为零来覆盖这些浏览器样式。这可以分别进行，也可以使用通用选择器（\*）对所有元素进行设置：

　　3. 在 CSS 中，width 和 height 指的是内容区域（element）的宽度和高度。增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸。假设框的每个边上有 10 个像素的外边距和 5 个像素的内边距。如果希望这个元素框达到 100 个像素，就需要将内容的宽度设置为 70 像素，以下是CSS代码：

下图是对上面CSS代码的解释：

# **CSS边框与边距**

## 边框

元素的边框 (border) 是围绕元素内容和内边距的一条或多条线。CSS border 属性允许你规定元素边框的样式、宽度和颜色。

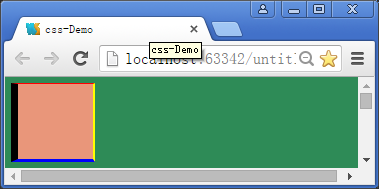
### 常见的边框属性及说明

|  |
| --- |
| **属性名称 属性值 说明**  border-color 十六进制 可依次设置上右下左边框颜色  英文名称 border-color:red(四边均为红色)  三原色 border-color:red green  (上下为红色，左右为绿色)  border-color:red green blue  (上为红色、左右为绿色、下为蓝色)  border-color:red green blue yellow  (上右下左分别为红绿蓝黄)  border-style none 不显示边线  dotted 点线  dashed 虚线  solid 实线  double 双线  border-width 边框宽度 border-width:0.2cm 0.3cm 0.4cm 0.5cm;  border-width:1 2 3 4;（上右下左） |

**eg：**

|  |
| --- |
| **css代码：**  div {  width: 100px;  height: 100px;  background: darksalmon;  **border-color**: red yellow blue black;（上右下左）  **border-style**: solid;  **border-width**: 1px 3px 5px 10px; （上右下左）  }  **html代码：**  <div></div> |

**效果：**

****

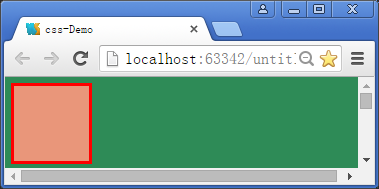
### 边框的简化设置

|  |
| --- |
| **格式:**  边框宽度 -> 边框样式 -> 边框颜色  border: 5px solid red |

**eg：**

|  |
| --- |
| **css代码：**  div **{**  width: 100px;  height: 100px;  background: darksalmon;  **border: 5px solid red;**  **}**  **html代码：**  <div></div> |

**效果：**

****

## CSS边距

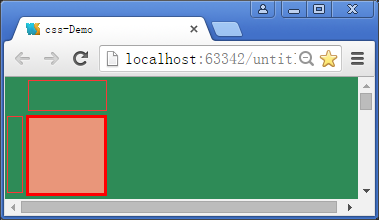
### 外边距margin属性

|  |
| --- |
| **属性名称 属性值 说明**  margin-bottom auto 自动调整边距  长度/百分比 设置下边距  margin-left auto 自动调整边距  长度/百分比 设置左边距  margin-right auto 自动调整边距  长度/百分比 设置右边距  margin-top auto 自动调整边距  长度/百分比 设置上端边距 |

**eg：**

|  |
| --- |
| **css代码：**  div {  width: 100px;  height: 100px;  background: darksalmon;  border: 5px solid red;  **margin-top: 50px;**  **margin-left: 20px;**  }  **html代码：**  <div></div> |

**效果：**

****

### margin属性简写方案

|  |
| --- |
| **格式:**  **margin:** 20px 40px **（上下，左右 ）**  **margin:** 20px 60px 100px**（上，左右，下）**  **margin:** 10px 20px 30px 40px**（上，右，下，左）** |

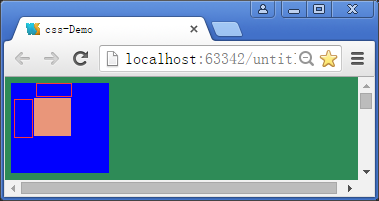
### 内边距padding属性（内填充）

|  |
| --- |
| **属性名称 属性值 说明**  padding-bottom 长度/百分比 元素下端内边距  padding-left 长度/百分比 元素左端内边距  padding-right 长度/百分比 元素右端内边距  padding-top 长度/百分比 元素上端内边距 |

**eg：**

|  |
| --- |
| **css代码：**  .div1 {  width: 100px;  height: 100px;  background: blue;  **padding-left: 30px;**  **padding-top: 20px;**  }  .div2 {  width: 50px;  height: 50px;  background: darksalmon;  }  **html代码：**  <div class="div1">  <div class="div2"></div>  </div> |

**效果：**

****

注意：使用padding（内填充）属性，外面的box1会被“撑起来”。

### padding属性简写方案

|  |
| --- |
| **格式:**  **padding**: 10px (**上下左右 各10px**)  **padding**: 2px 4px（**上下，左右**）  **padding**: 2px 6px 10px（**上，左右，下**）  **padding**: 1px 2px 3px 4px（**上，右，下，左**） |

# **CSS区块、浮动**

## HTML元素的布局级别

1、块级元素（block-level elements）

块级元素通常被显示为独立的一块，会单独换一行。

2、内联元素（inline elements）

内联元素则前后不会产生换行，一系列inline元素都在一行内显示，直到该行排满。

3、常见的块级元素有： DIV, FORM, TABLE, P, PRE, H1~H6, DL, OL, UL 等。

常见的内联元素有： SPAN, A, STRONG, EM, LABEL, INPUT, SELECT, TEXTAREA, IMG, BR 等。

4、block元素可以包含block元素和inline元素；但inline元素只能包含inline元素。要注意的是这个是个大概的说法，每个特定的元素能包含的元素也是特定的，所以具体到个别元素上，这条规律是不适用的。比如 P 元素，只能包含inline元素，而不能包含block元素。

5、<div>元素

用途：网页布局；排版网页内容。

用法：<div>网页的内容</div>

6、<span>元素

用途：组合文档中的行内元素

用法：<span>网页内容</span>

## CSS区块

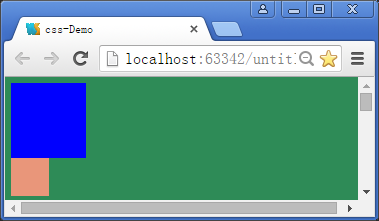
### 区块元素常见的属性

|  |
| --- |
| **属性名称 属性值 说明**  width 像素/百分比 区块的宽度  height 像素/百分比 区块的高度  min-height 像素/百分比 区块最小高度  max-height 像素/百分比 区块最大高度  min-width 像素/百分比 区块最小宽度  max-width 像素/百分比 区块最大宽度 |

**eg：**

|  |
| --- |
| **css代码：**  .div1 {  width: 100px;  height: 100px;  background: blue;  }  .div2 {  width: 50px;  height: 50px;  background: darksalmon;  }  **html代码：**  <div class="div1"></div>  <div class="div2"></div> |

**效果：两个div各占一行**

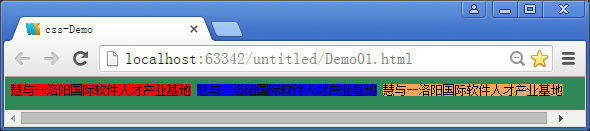
****

**内联元素：**

**eg：**

|  |
| --- |
| **css代码：**  .sp1{  background: red;  }  .sp2{  background: blue;  }  .sp3{  background: sandybrown;  }  **html代码：**  <span class="sp1">慧与--洛阳国际软件人才产业基地</span>  <span class="sp2">慧与--洛阳国际软件人才产业基地</span>  <span class="sp3">慧与--洛阳国际软件人才产业基地</span> |

**效果：三个span默认会在一行显示**



### display属性

可以通过设置**display属性的值来**改变元素的布局级别。display的值有三个：**block，inline和inline-block，none。**

**display:block**

block元素的默认值，会独占一行，多个block元素会各自新起一行。默认情况下，block元素宽度自动填满其父元素宽度。

block元素可以设置width,height属性。块级元素即使设置了宽度,仍然是独占一行。

block元素可以设置margin和padding属性。

**display:inline**

inline元素的默认值，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。

inline元素设置width,height属性无效。

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；但竖直方向的padding-top, padding-bottom, margin-top, margin-bottom不会产生边距效果。

**display:inline-block**

简单来说就是将对象呈现为inline对象，但是对象的内容作为block对象呈现。之后的内联对象会被排列在同一行内。比如我们可以给一个link（a元素）inline-block属性值，使其既具有block的宽度高度特性又具有inline的同行特性。

## CSS中浮动的使用

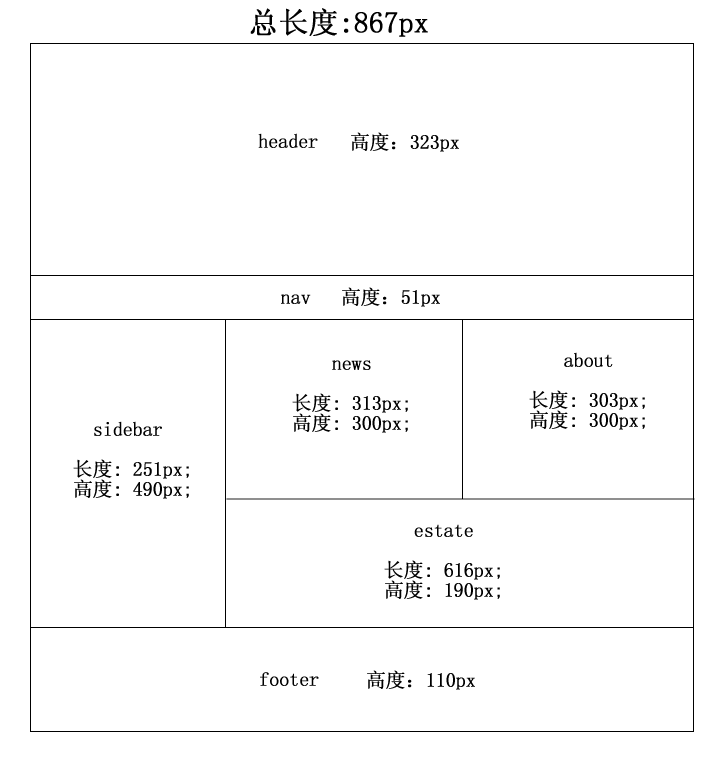
### float属性的值和说明

CSS 中 float（浮动）属性，会使元素向左或向右移动，其周围的元素也会重新排列。float（浮动），往往是用于图像，但它在布局时一样非常有用。

浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

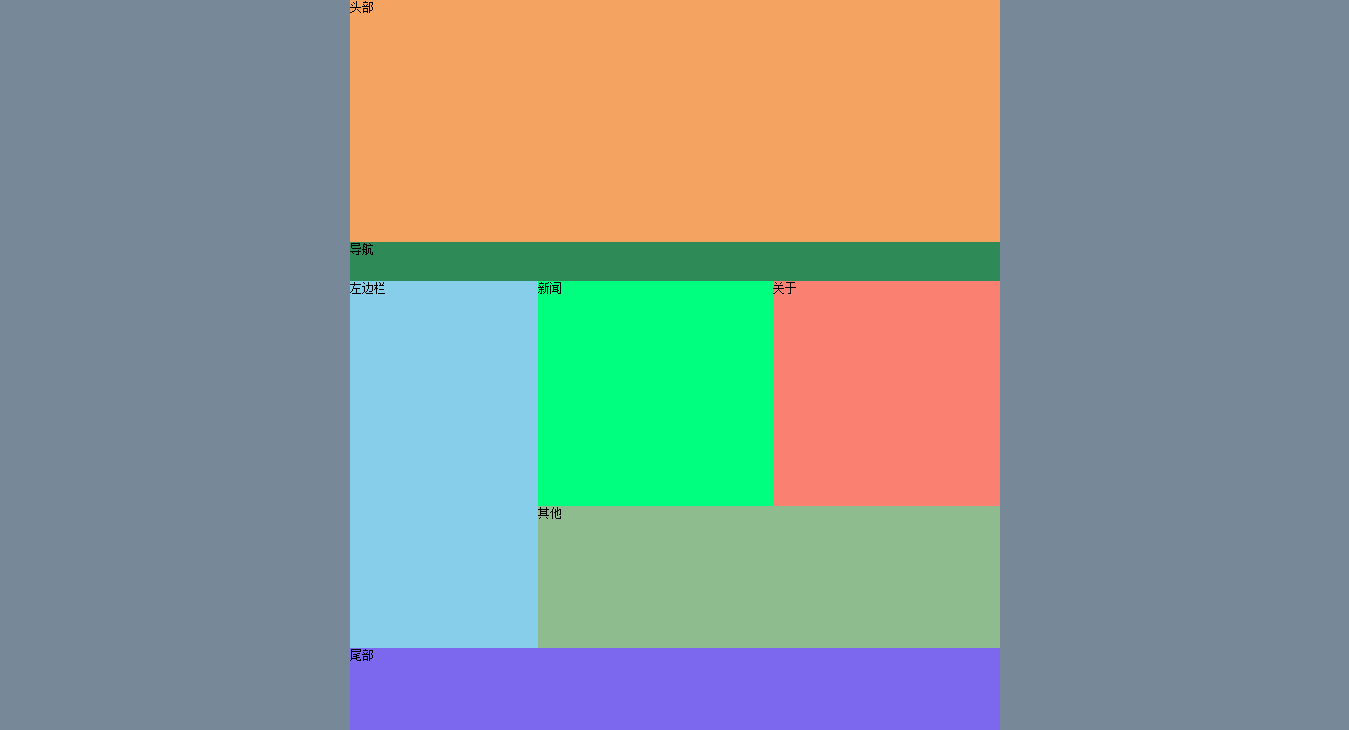
|  |
| --- |
| **属性名称 属性值 说明**  float none 正常显示  left 左浮动  right 右浮动  clear none 允许两边浮动  left 清除左边浮动  right 清除右边浮动  both 清除两边浮动 |

**eg：采用浮动（float）布局实现如下页面布局：**

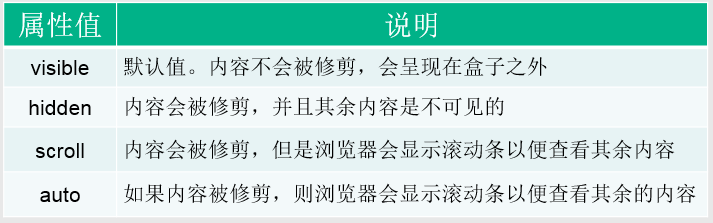


|  |
| --- |
| **css代码：**  \*{  margin: 0px;  padding: 0px;  }  body{  width: 867px;  margin: 0 auto;  }  #header{  width: 867px;  height: 323px;  background: sandybrown;  }  #nav{  width: 867px;  height: 51px;  background: seagreen;  }  #sidebar{  width: 251px;  height: 490px;  background: skyblue;  **float: left;**  }  #news{  width: 313px;  height: 300px;  background: springgreen;  **float: left;**  }  #about{  width: 303px;  height: 300px;  background: salmon;  **float: left;**  }  #estate{  width: 616px;  height: 190px;  background: darkseagreen;  **float: left;**  }  #footer{  width: 867px;  height: 110px;  background: mediumslateblue;  **clear: both;**  }  **html代码：**  **<div id="header">头部</div>**  **<div id="nav">导航</div>**  **<div id="sidebar">左边栏</div>**  **<div id="news">新闻</div>**  **<div id="about">关于</div>**  **<div id="estate">其他</div>**  **<div id="footer">尾部</div>** |

**效果：**

****

### 溢出overflow属性



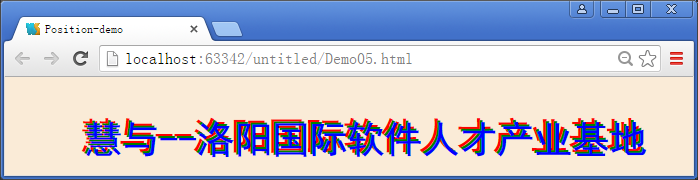
### CSS中定位的使用（了解）

CSS定位属性允许你为一个元素定位。它也可以将一个元素放在另一个元素后面。

元素可以使用的顶部，底部，左侧和右侧属性定位。

|  |
| --- |
| **属性名称 属性值 说明**  position relative 设置区块基准点为左上角  absolute 设置网页的左上角为基准点  static 无设置（默认）  fixed 固定定位  left auto 以基准点定位在左边  像素/百分比 定位在左边  top auto 以基准点定位在上边  像素/百分比 定位在上边  right auto 以基准点定位在右边  像素/百分比 定位在右边  bottom auto 以基准点定位在下边  像素/百分比 定位在下边  z-index auto 自动调整高度  数字 数字越大越往上层 |

**eg:在浏览器中实现阴影效果：**



|  |
| --- |
| **css代码：**  \*{  margin: 0px;  padding: 0px;  font-size: 50px;  font-family: 黑体;  }  #p1{  **position: absolute;**  **top: 50px;**  **left: 100px;**  color: red;  **z-index: 1;**  }  #p2{  **position: absolute;**  **top: 53px;**  **left: 103px;**  color: green;  **z-index: 5;**  }  #p3{  **position: absolute;**  **top: 56px;**  **left: 106px;**  color: blue;  **z-index: 10;**  }  **html代码：**  **<p id="p1">慧与--洛阳国际软件人才产业基地</p>**  **<p id="p2">慧与--洛阳国际软件人才产业基地</p>**  **<p id="p3">慧与--洛阳国际软件人才产业基地</p>** |