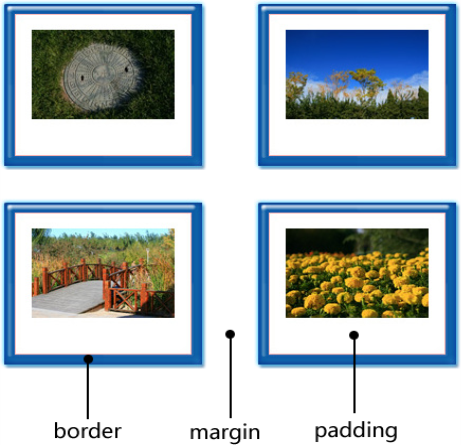
### 1. “盒子”与“模型”的概念

网页设计中常听的属性名：

内容(content)、填充(padding)、边框(border)、边界(margin)， 这些都是CSS盒子具备属性。

这些属性我们可以把它转移到我们日常生活中的盒子（箱子）上来理解，日常生活中所见的盒子也就是能装东西的一种箱子，也具有这些属性，所以叫它盒子模型。CSS盒子模型就是在网页设计中经常用到的CSS技术所使用的一种思维模型。

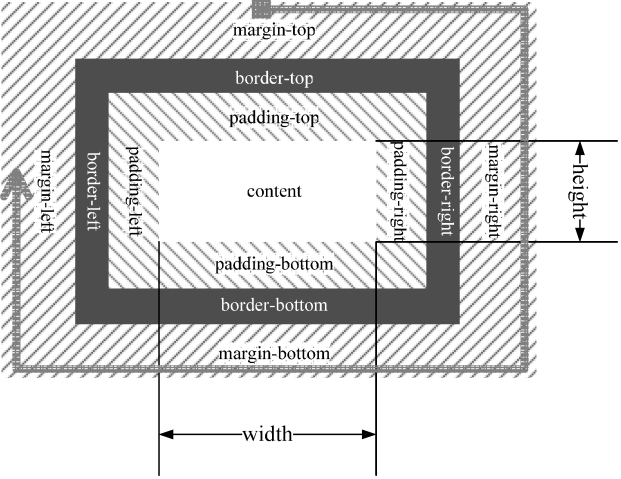


特点：每个盒子都有：**边界、边框、填充、内容**四个属性；并且每个盒子都有尺寸。

**每个属性都包括四个部分：上、右、下、左；**这四部分可同时设置，也可分别设置。

列如：我们把把盒子模型理解网上购物的包装盒，填充可以理解为盒子里的抗震辅料厚度，而边框有大小和颜色之分，我们又可以理解为生活中所见盒子的厚度以及这个盒子是用什么颜色材料做成的，边界就是该盒子与其它东西要保留多大距离。

### 2. 盒子模型详解



盒子有尺寸，用CSS写法为**宽度（width）**和**高度（height）**

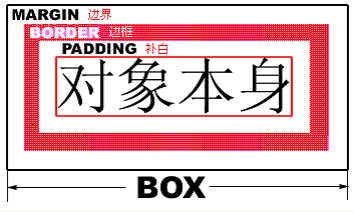
盒子有边框，用CSS写法为上下左右**边框（border）**

盒子有内边距，CSS写法为上下左右**内边距（padding）**

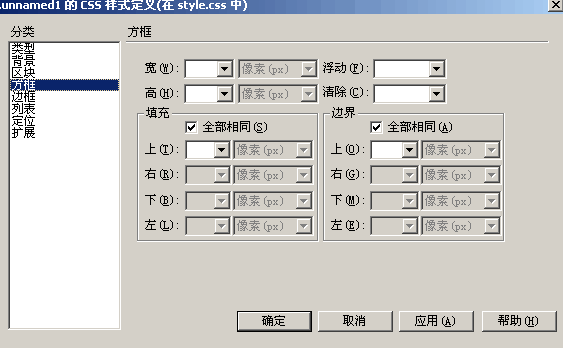
盒子有外边距，CSS写法为**外边距（margin）**

**在HTML中，所有的容器标签都可以认为是一个盒子，但是行内标签与块标签有着明显的差别。 行内标签无法设置尺寸（宽度和高度）等等，块标签是一个完整的盒子。**

简化版盒子模型：



## 二、尺寸 width、height（掌握）



### 1. 宽度width

作用： 检索或设置对象的宽度。

语法： **width :auto**| *length*

示例：

|  |
| --- |
| p { width: 200px; } |

只有块元素才可以设置宽度，行内元素无法设置宽度。

### 2. 高度 height

作用： 检索或设置对象的高度。

语法： **height:auto**| *length*

示例：

|  |
| --- |
| p { height: 200px; } |

只有块元素才可以设置高度，行内元素无法设置高度。

### 3. 最小高度min-height

作用： 检索或设置对象的最小高度。

语法： **min-height :none**| *length*

示例：

|  |
| --- |
| p { min-height: 200px; } |

只有块元素才可以设置最小高度，行内元素无法设置最小高度。

实际高度大于最小高度时，以实际高度为准

### 4. 最大高度max-height

作用： 检索或设置对象的最大高度。

语法： **max-height :none**| *length*

示例：

|  |
| --- |
| p { max-height: 200px; } |

只有块元素才可以设置最大高度，行内元素无法设置最大高度。

**实际高度大于最大高度时，超出内容的处理方式可以通过overflow决定是否显示。**

## 三、内边距 padding

内边距也称为内补丁，是边框与内容之间的距离。

### 1. 上边内边距 padding-top

作用： 检索或设置对象的上边内边距。

语法： **padding-top :***length*

示例：

|  |
| --- |
| p { padding-top: 36px; } |

### 2. 右边内边距 padding-right

作用： 检索或设置对象的右边内边距。

语法： **padding-right:***length*

示例：

|  |
| --- |
| p { padding-right: 36px; } |

### 3. 下边内边距 padding-bottom

作用： 检索或设置对象的下边内边距。

语法： **padding-bottom :***length*

示例：

|  |
| --- |
| p { padding-bottom: 36px; } |

### 4. 左边内边距 padding-left

作用： 检索或设置对象的左边内边距。

语法： **padding-left:***length*

示例：

|  |
| --- |
| p { padding-left: 36px; } |

### 5. 内边距 padding（重点中的重点）

作用： 检索或设置对象四边的内边距。

语法： **padding :***length*

示例：

|  |
| --- |
| p { padding: 20px; } |

如果只提供一个，将用于全部的四条边。  
如果提供两个，第一个用于上－下，第二个用于左－右。  
如果提供三个，第一个用于上，第二个用于左－右，第三个用于下。  
如果提供全部四个参数值，将按上－右－下－左的顺序作用于四边。

## 四、外边距 margin（掌握）

外边距也称为外补丁，是边框与其它标签之间的距离。

### 1. 上边外边距 margin-top

作用： 检索或设置对象的上边外边距。

语法： **margin-top :***length*

示例：

|  |
| --- |
| p { margin-top: 36px; } |

### 2. 右边外边距 margin-right

作用： 检索或设置对象的右边外边距。

语法： **margin-right:***length*

示例：

|  |
| --- |
| p { margin-right: 36px; } |

### 3. 下边外边距 margin-bottom

作用： 检索或设置对象的下边外边距。

语法： **margin-bottom :***length*

示例：

|  |
| --- |
| p { margin-bottom: 36px; } |

### 4. 左边外边距 margin-left

作用： 检索或设置对象的左边外边距。

语法： **margin-left:***length*

示例：

|  |
| --- |
| p { margin-left: 36px; } |

### 5. 外边距 margin

作用： 检索或设置对象四边的外边距。

语法： **margin:***length*

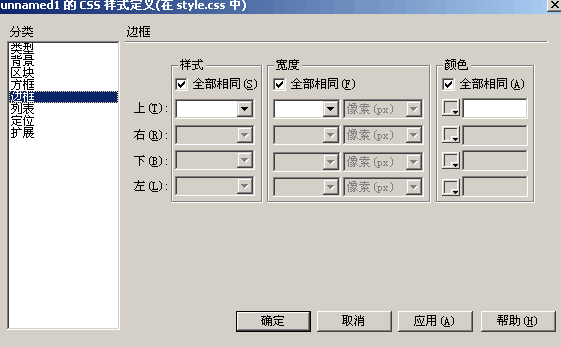
示例：

|  |
| --- |
| p { margin: 20px; } |

如果只提供一个，将用于全部的四条边。  
如果提供两个，第一个用于上－下，第二个用于左－右。  
如果提供三个，第一个用于上，第二个用于左－右，第三个用于下。  
如果提供全部四个参数值，将按上－右－下－左的顺序作用于四边。

**外边距在不同浏览器的效果会有很大不同，建议尽量少用。**

## 五、边框 border





### 1. 边框宽度 border-width

作用： 检索或设置对象的边框宽度。

语法： **border-width :medium**| **thin**| **thick**| *length*

示例：

|  |
| --- |
| p { border-style: solid; border-width: 1px; border-color: #666666;} |

如果只提供一个，将用于全部的四条边。  
如果提供两个，第一个用于上－下，第二个用于左－右。  
如果提供三个，第一个用于上，第二个用于左－右，第三个用于下。

如果提供全部四个参数值，将按**上－右－下－左**的顺序作用于四个边框。

要有边框线型的情况下，才能设置边框的宽度。

通常边框宽度都会设置成一个实际的象素值。

### 2. 边框线型 border-style

作用： 检索或设置对象的边框线型。

语法： **border-style :none**| **dotted**| **dashed**| **solid**| **double**

参数：

none（无边框）

dotted（点线）

dashed（虚线）

solid（实线）

double（双线）

groove（凹槽）

ridge（凸槽）

inset（凹边）

outset（凸边）

示例：

|  |
| --- |
| p { border-style: double; } |

注：如果写多个值，边框线型的数量和顺序与边框的宽度一致。

### 3. 边框颜色 border-color

作用： 检索或设置对象的边框颜色。

语法： **border-color :** *color*

示例：

|  |
| --- |
| p { border-color: red; } |

color的表现方式有三种，顺序同上。

### 4. 边框 border

作用： 检索或设置对象的边框。这是一个复合属性。

语法： **border : [border-width](c_borderwidth.html)**|**[border-style](c_borderstyle.html)**|**[border-color](c_bordercolor.html)**

示例：

|  |
| --- |
| /\* 设置 p标签的边框为： 宽度2px ， 线型实线 ， 颜色为红色 \*/  p { border:2px solid red; }  /\* 另一种写法，定义宽度和高度 \*/  .box{border: solid 2px #f00;width:200;height:100px;} |

### 5. 边框的细节控制

#### 5.1 上边框 border-top

作用： 检索或设置对象的上边框。这是一个复合属性。

语法： **border-top: [border-width](c_borderwidth.html)**|**[border-style](c_borderstyle.html)**|**[border-color](c_bordercolor.html)**

示例：

|  |
| --- |
| /\* 设置 p标签的上边框为： 宽度2px ， 线型实线 ， 颜色为红色 \*/  p { border-top:2px solid red; } |

#### 5.2 下边框 border-bottom

作用： 检索或设置对象的下边框。这是一个复合属性。

语法： **border-bottom : [border-width](c_borderwidth.html)**|**[border-style](c_borderstyle.html)**|**[border-color](c_bordercolor.html)**

示例：

|  |
| --- |
| /\* 设置 p标签的下边框为： 宽度2px ， 线型实线 ， 颜色为红色 \*/  p { border-bottom:2px solid red; } |

#### 5.3 左边框 border-left

作用： 检索或设置对象的左边框。这是一个复合属性。

语法： **border -left: [border-width](c_borderwidth.html)**|**[border-style](c_borderstyle.html)**|**[border-color](c_bordercolor.html)**

示例：

|  |
| --- |
| /\* 设置 p标签的左边框为： 宽度2px ， 线型实线 ， 颜色为红色 \*/  p { border-left:2px solid red; } |

#### 5.4 右边框 border-right

作用： 检索或设置对象的右边框。这是一个复合属性。

语法： **border -right: [border-width](c_borderwidth.html)**|**[border-style](c_borderstyle.html)**|**[border-color](c_bordercolor.html)**

示例：

|  |
| --- |
| /\* 设置 p标签的右边框为： 宽度2px ， 线型实线 ， 颜色为红色 \*/  p { border-right:2px solid red; } |

## 六、标准模式与怪异模式

### 1. 什么是标准模式与怪异模式？

盒模型一共有两种模式，一种是标准模式，另一种就是怪异模式。

当创建网页时通常会在第一行写上标准声明，如：

|  |
| --- |
| <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <!DOCTYPE HTML> |

以上几种DOCTYPE都是标准的文档类型，无论使用哪种模式完整定义DOCTYPE，都会触发标准模式，而如果DOCTYPE缺失则在**ie6，ie7，ie8**下将会触发怪异模式（quirks 模式）。

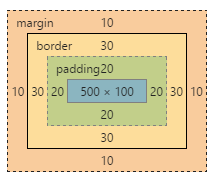
### 2. 标准盒子模型

**标准模式**的盒子模型：padding和border不被包含在定义的width和height之内。对象的实际宽度等于设置的width值和border、padding之和，即 ( Element width = width + border + padding ) 。

示例：

|  |
| --- |
| div {  background:#ddd;  width:500px;  padding:20px;  box-sizing:content-box;  border:30px solid #f00;  height:100px;  margin:10px;  } |

效果：



### 3. 怪异盒子模型

**怪异模式**的盒子模型：padding和border被包含在定义的width和height之内。对象的实际宽度就等于设置的width值，即使定义有border和padding也不会改变对象的实际宽度，即 ( Element width = width ) 。

示例：

|  |
| --- |
| div {  background:#ddd;  width:500px;  padding:20px;  box-sizing:border-box;  border:30px solid #f00;  height:100px;  margin:10px;  } |

效果：

### 4. Box-sizing 盒模型组成模式

作用：设置或检索对象的盒模型组成模式。

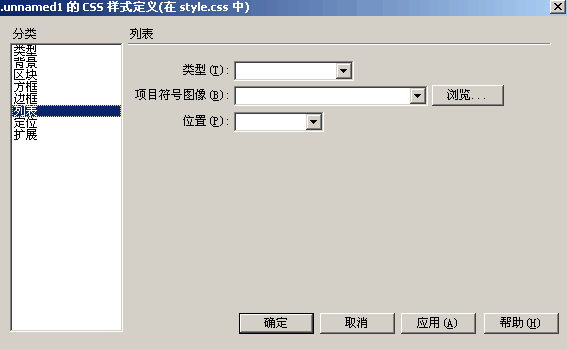
语法：box-sizing：content-box | border-box

默认值：content-box

content-box：将对象定义为标准模式下的盒模型，{element width=border+padding+content}。

border-box：将对象定义为怪异模式下的盒模型，{element width=content}。

## 七、列表 list（掌握）



### 1. 列表标记位置

作用： 设置或检索作为对象的列表项标记如何根据文本排列。

语法： **list-style-position :outside**| **inside**

示例：

|  |
| --- |
| Ul li { list-style-position: inside; } |

### 2. 列表标记符号

作用： 设置或检索对象的列表项所使用的预设标记。

语法： **list-style-type :disc**| **circle**| **square**| **decimal**| **lower-roman**| **upper-roman**| **lower-alpha**| **upper-alpha**| **none**| **armenian**| **cjk-ideographic**| **georgian**| **lower-greek**| **hebrew**| **hiragana**| **hiragana-iroha**| **katakana**| **katakana-iroha**| **lower-latin**| **upper-latin**

**disc（实心圆）、circle（空心圆）、square（方块）、decimal（阿拉伯数字）、lower-roman（小写罗马数字）、upper-roman（大写罗马数字）、lower-alpha（小写英文字母）、upper-alpha（大写英文字母）、none（无项目符号）**

示例：

|  |
| --- |
| Ul li { list-style-type: square; } |

注： 实际应用中通常都不需要标记符号，而是使用图片实现效果

### 列表图片

**作用：list-style-image 属性使用一幅图像来替换列表项的标记。**

**参数：**

url() 图像的路径。

none 默认。无图形被显示。

示例：

|  |
| --- |
| Ul li { list-style-image: url(img/HBuilder.png);} |

### 4. 列表样式

作用： 设置列表项目相关内容的样式。

语法： **list-style : [list-style-image](c_liststyleimage.html)**|**[list-style-position](c_liststyleposition.html)**|**[list-style-type](c_liststyletype.html)**

示例：

|  |
| --- |
| Ul li { list-style:outside square; } |

注： 实际应用中通常都不需要标记符号，而是使用图片实现效果。

## 八、其它

### 1. 表格边框合并 table

作用： border-collapse 属性设置表格的边框是否被合并为一个单一的边框，还是象在标准的 HTML 中那样分开显示。

语法： **border-collapse :separate**| **collapse**

参数：

separate默认。边框会被分开。

collapse如果可能，边框会被合并为一个单一的边框。

<style type="text/css">

table

{border-collapse:collapse;}

td

{border:solid 1px #333333;}

</style>

示例效果，细边线表格：

|  |
| --- |
| <style type="text/css">  table  {border-collapse:collapse;}  td  {border:solid 1px #333333;}  </style>  <table width="360" border="1" cellspacing="0" cellpadding="2">  <tr>  <td>&nbsp;</td>  <td>&nbsp;</td>  </tr>  <tr>  <td>&nbsp;</td>  <td>&nbsp;</td>  </tr>  <tr>  <td>&nbsp;</td>  <td>&nbsp;</td>  </tr>  <tr>  <td>&nbsp;</td>  <td>&nbsp;</td>  </tr>  <tr>  <td>&nbsp;</td>  <td>&nbsp;</td>  </tr>  </table> |

注： 常用于让表格生成细边框线条，必须与 td的边框搭配。

### 2. 鼠标形状 cursor

作用： 设置或检索当鼠标滑过样式控制的对象时改变鼠标形状。

语法： **cursor:** auto| crosshair| default| hand| move| help| wait| text| w-resize|s-resize| n-resize|e-resize| ne-resize|sw-resize| se-resize| nw-resize|pointer| url (*url*)

**hand（手型，IE8及以下，新版浏览器要使用Pointer表示点击的手型）、crosshair（“十”型）、text （“I”型）、wait（等待）、default（默认）、help（帮助）、e-resize（东箭头）、ne-resize（东北箭头）、n-resize（北箭头）、nw-resize（西北箭头）、w-resize（西箭头）、sw-resize（西南箭头）、s-resize（南箭头）、se-resize（东南箭头）和auto（自动）**

示例：

|  |
| --- |
| div { cursor:pointer; } |

注： 常在非超链接的对象上设置成手型，配合JavaScript效果，让用户认为是可以点击的模块。