# **CSS盒子模型**

## **一、CSS盒子模型**

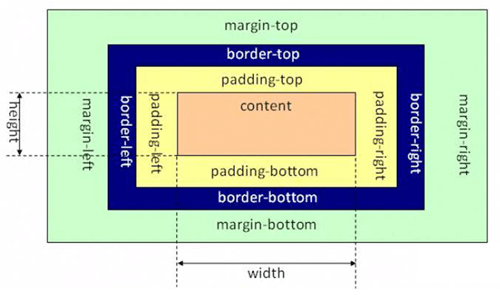
在之前的HTML学习中，我们学习了一个很重要的理论“**块元素和行内元素**”。在这我们将接触CSS中一个极其重要的理论，那就是“CSS盒子模型”理论。

在“CSS盒子模型”理论中，所有页面中的元素都可以看成一个盒子，并且占据着一定的页面空间。

一个页面由很多这样的盒子组成，这些盒子之间会互相影响，因此掌握盒子模型需要从两个方面来理解：一是理解单独一个盒子的内部结构，二是理解多个盒子之间的相互关系。（**现在不懂没关系，听我娓娓道来**）

每个元素都看成一个盒子，盒子模型是由content（内容）、padding（内边距）、margin（外边距）和border（边框）这四个属性组成的。此外，在盒子模型中，还有宽度width和高度height两大辅助性属性。记住，所有的元素都可以看做一个盒子！

下图为一个CSS盒子模型的内部结构：



从上图中我们可以得出盒子模型的属性如下：

| **属性** | **说明** |
| --- | --- |
| border | （边框）元素边框 |
| margin | （外边距）用于定义页面中元素与元素之间的距离 |
| padding | （内边距）用于定义内容与边框之间的距离 |
| content | （内容）可以是文字或图片 |

其中，padding称为“内边距”，也常常称为“补白”。

### **1、内容区**

内容区是CSS盒子模型的中心，它呈现了盒子的主要信息内容，这些内容可以是文本、图片等多种类型。内容区是盒子模型必备的组成部分，其他的3部分都是可选的。

内容区有3个属性：width、height和overflow。使用width和height属性可以指定盒子内容区的高度和宽度。在这里注意一点，width和height这两个属性是针对内容区而言，并不包括padding部分。

当内容信息太多时，超出内容区所占范围时，可以使用overflow溢出属性来指定处理方法。对于overflow这个属性，我们在后面会详细讲解到。

### **2、内边距**

内边距，指的是内容区和边框之间的空间，可以被看做是内容区的背景区域。

关于内边距的属性有5种，即padding-top、padding-bottom、padding-left、padding-right以及综合了以上4个方向的简洁内边距属性padding。使用这5种属性可以指定内容区域各方向边框之间的距离。

### **3、边框**

在CSS盒子模型中，边框跟我们之前学过的边框是一样的。

边框属性有border-width、border-style、border-color以及综合了3类属性的快捷边框属性border。

### **4、外边距**

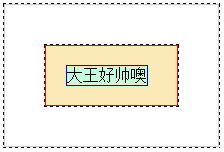
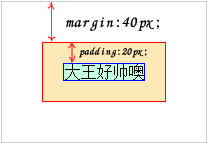
外边距，指的是两个盒子之间的距离，它可能是子元素与父元素之间的距离，也可能是兄弟元素之间的距离。

外边距使得元素之间不必紧凑地连接在一起，是CSS布局的一个重要手段。

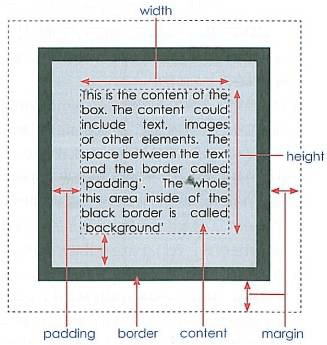
外边距的属性也有5种，即margin-top、margin-bottom、margin-left、margin-right以及综合了以上4个方向的简洁内边距属性margin。

同时，CSS允许给外边距属性指定负数值，当指定负外边距值时，整个盒子将向指定负值的相反方向移动，以此可以产生盒子的重叠效果。这就是传说中的“负margin技术”。那个以后再讲~

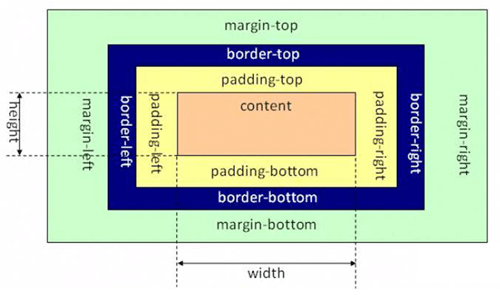
![`D0[C_OSH01TIDG]({KUHBR](data:image/png;base64,)

****

**接下来我们一起试试~**



# **宽度width和高度height**

从CSS盒子模型中，我们可以看出，元素的宽度width和高度height是针对内容区而言的，大家要非常清楚这一点。很多初学者容易把补白也认为是内容区的一部分。

|  |  |
| --- | --- |
| 1  2 | width：像素值;  height：像素值 |

只有块元素能设置width和height，行内元素无法设置width和height。

 <style type="text/css">

        #main div

        {

            width:100px;

            height:30px;

            border:1px solid red;

        }

        #main span

        {

            width:100px;

            height:30px;

            border:1px solid blue;

        }

    </style>

</head>

<body>

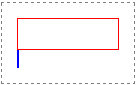
    <div id="main">

        <div></div>

        <span></span>

    </div>

</body>

←这个是效果

“#main div{}”这个选择器表示选中id为#main的元素下面的子元素div，这是一个子元素选择器；而“#main span{}”这个选择器表示选中id为#main的元素下面的子元素span，这也是一个子元素选择器，div元素是块元素，span是行内元素。因此div元素可以设置宽度width和高度height，而span元素无法设置宽度和高度。如果想给行内元素设置宽度和高度，需要用到display:inline-block这个属性，所以我们现在不讲！

## **一、padding 局部样式**

从CSS盒子模型中，我们可以看出，内边距padding分为四个方向的内边距：padding-top、padding-right、padding-bottom、padding-left。

|  |  |
| --- | --- |
| 1  2  3  4 | padding-top:像素值;  padding-right:像素值;  padding-bottom:像素值;  padding-left:像素值; |

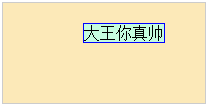
## **二、padding简洁写法**

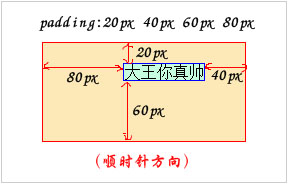
我们可以使用padding属性来设置四个方向的内边距。在实际编程中，我们往往使用的是padding的这种高效简洁写法来编程。padding写法有3种，分别如下：

|  |  |
| --- | --- |
| 1  2  3 | padding:像素值;  padding:像素值1 像素值2;  padding:像素值1 像素值2 像素值3 像素值4; |

1“padding:20px;”表示四个方向的内边距都是20px；

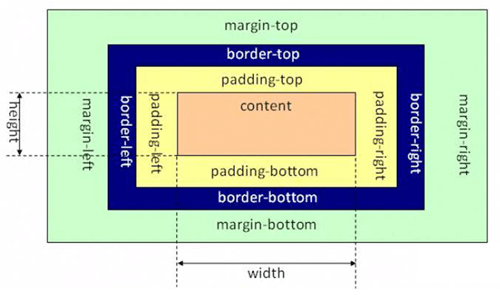
2“padding:20px 40px;”表示padding-top上和下padding-bottom为20px，padding-right右和左padding-left为40px。

3“padding:20px 40px 60px 80px;”表示padding-top为20px上，padding-right为40px右，padding-bottom为60px下，padding-left为80px左。

我们先试试各个属性设置，然后再试一下简写设置

# **外边距margin**

外边距margin，指的是边框到父元素或者同级元素之间的那一部分。



## **一、margin局部样式**

从CSS盒子模型中，我们可以看出，内边距分为四个方向的内边距：margin-top、margin -right、margin -bottom、margin -left。这一点跟内边距padding非常相似。

|  |  |
| --- | --- |
| 1  2  3  4 | margin-top:像素值;  margin-right:像素值;  margin-bottom:像素值;  margin-left:像素值; |

## **二、margin简洁写法**

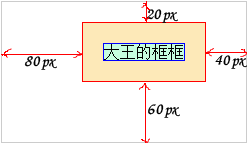
margin跟padding一样，也有简洁写法。我们可以使用margin属性来设置四个方向的外边距。在实际编程中，我们往往使用的是margin的这种高效简洁写法来编程。

margin写法有3种，分别如下：

|  |  |
| --- | --- |
| 1  2  3 | margin:像素值;  margin:像素值1 像素值2;  margin:像素值1 像素值2 像素值3 像素值4; |

1“margin:20px;”表示四个方向的外边距都是20px；

2“margin:20px 40px;”表示margin-top和margin-bottom为20px，margin-right和margin-left为40px。

3“margin:20px 40px 60px 80px;”表示margin-top为20px，margin-right为40px，margin-bottom为60px，margin-left为80px。大家按照顺时针方向记忆就可以了。我们先试试，然后用简易写法试试上下。左右同等数值。