

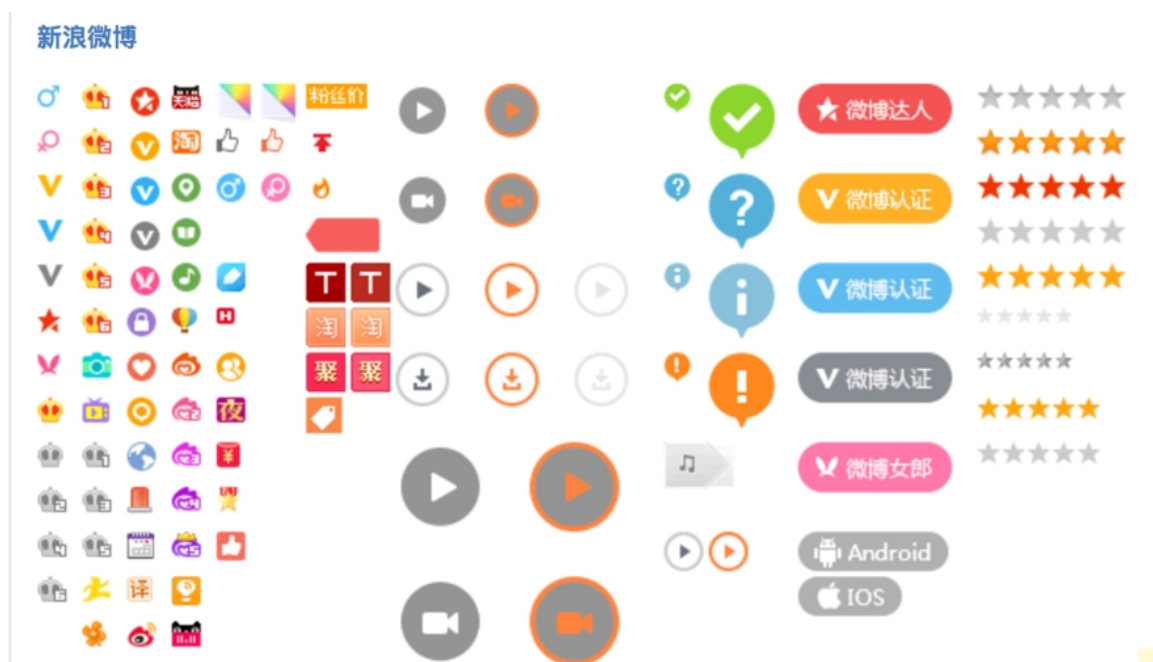
CSS02- 精灵图,盒模型,浮动

一.精灵图

1. 概念

精灵图(CSS Sprites)是一种网页图片应用处理方式, 它将一个页面中的多个小图片整合到一张单独的图片中, 然后通过CSS的背景图属性 background-image和背景图位置 background-position属性来显示图片的不同部分。这样做的好处是可以减少HTTP请求, 从而加快页面加载速度。

精灵图原图:



通过设置标签的背景图片属性来显示精灵图中的某一个图标;

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title></title>
    <style>
      body{
        background-color: hotpink;
      }
      #one{
        width: 40px;
        height: 30px;
        margin: 50px;
        /* 精灵图设置关键代码 */
        background-image: url("img/jinglingtu.webp");
        /* 通过设置背景图片的位置 来显示指定图标 */
        background-position: -50px -50px;
      }
    </style>
```

```
</head>
<body>
  <div id="one">

    </div>
</body>
</html>
```

运行结果:



二. 盒模型

1. 概念

盒模型结构像一个盒子一样，包含多个组成部分，用来页面的精细化布局。**所有HTML元素可以看作盒子**，在CSS中，"box model"这一术语是用来设计和布局时使用。CSS盒模型本质上是一个盒子，封装周围的HTML元素，它包括：边距，边框，填充，和实际内容。盒模型允许我们在其它元素和周围元素边框之间的空间放置元素。

2. 盒模型的组成

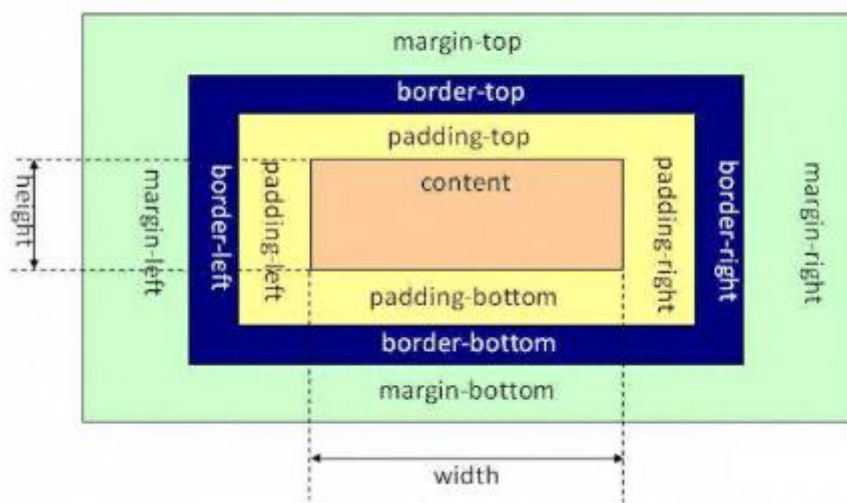
- 内容区域：content，展示内容（文字，图片等），大小只由宽和高决定。width和height。
- 内填充：padding，调整是元素自身的内容和元素边框的距离。注意padding的简写形式，四个值，三个值，两个值，一个值。
- 边框：border，可以分别设置四个方向上的边框。
- 外间距：margin，调整的是该元素和相邻元素的距离。

3. 标准盒模型

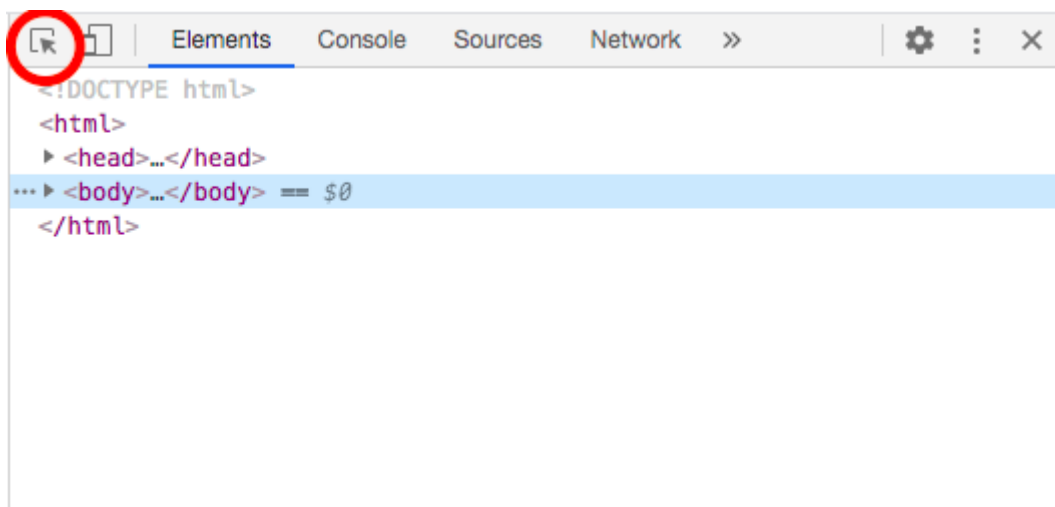
- `box-sizing`属性，默认值是`content-box`，叫做标准盒模型。
- 标准盒模型的宽度是：内容区域width+padding左右内填充+border左右边框宽度+margin左右外间距。
- 标准盒模型的高度同宽度计算方法一致。

```
选择器{  
    box-sizing: content-box;  
}
```

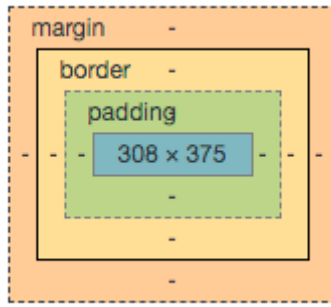
■ 标准盒子模型



- 打开浏览器的检查工具,可以快速查看盒模型状态.
 - 在网页的空白区域右击，弹出菜单。
 - 在菜单中选择"检查"。
 - 选中代码检查区的左上角的选择按钮。



- 移动鼠标，移动到要查看盒模型的矩形区域上，然后鼠标左键单击，在代码检查区的右部可查看该元素的盒模型结构。



4. 构建盒模型

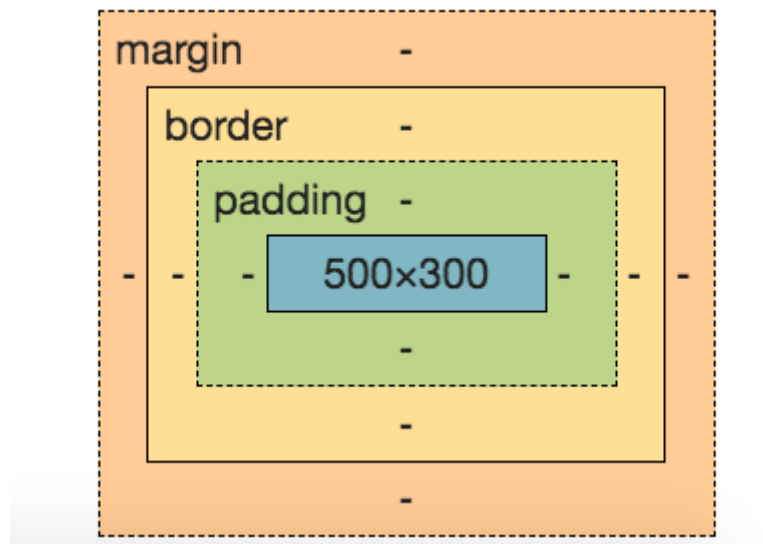
4.1. 在只设置宽高情的情况下

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>01盒模型</title>
    <style type="text/css">
      .one{
        width: 500px;
        height: 300px;
        background-color: palegreen;
      }
    </style>
  </head>
  <body>

    <div class="one">
      蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦
      沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙
      卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙
      卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙
      卡拉卡
    </div>
  </body>
</html>
```

蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦
沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙
卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙
卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙
卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙
卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡

浏览器解析盒模型的尺寸:



4.2. 设置padding内填充

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>01盒模型</title>
    <style type="text/css">
      .one{
        width: 500px;
        height: 300px;
        background-color: palegreen;

        /* padding */

        /* 写法1: 上下左右 */
        /* padding: 20px; */

        /* 写法2: 上下 左右 */
        /* padding: 20px 30px; */

        /* 写法3: 上 左右 下 */
        /* padding: 20px 30px 0; */

        /* 写法4: 上 右 下 左 */
        /* padding: 10px 20px 0 30px; */

        /* 写法5: 按照方向 left top right bottom */
        padding-top: 10px;
        padding-right: 20px;
        padding-bottom: 50px;
        padding-left: 50px;

      }
    </style>
  </head>
  <body>

    <div class="one">
```


- 示例

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>01盒模型</title>
    <style type="text/css">
      .one{
        width: 500px;
        height: 300px;
        background-color: palegreen;

        /* padding */

        /* 写法1: 上下左右 */
        /* padding: 20px; */

        /* 写法2: 上下 左右 */
        /* padding: 20px 30px; */

        /* 写法3: 上 左右 下 */
        /* padding: 20px 30px 0; */

        /* 写法4: 上 右 下 左 */
        /* padding: 10px 20px 0 30px; */

        /* 写法5: 按照方向 left top right bottom */
        padding-top: 10px;
        padding-right: 20px;
        padding-bottom: 50px;
        padding-left: 50px;

        /* border边框 */
        /* 写法1: 单独设置 */
        /* border-width: 5px; */ /*宽度*/
        /* border-color: red; */ /*颜色*/
        /* border-style: solid; */ /*样式: solid 实线; dotted 圆形虚
线; dashed 方形虚线; double 双线*/

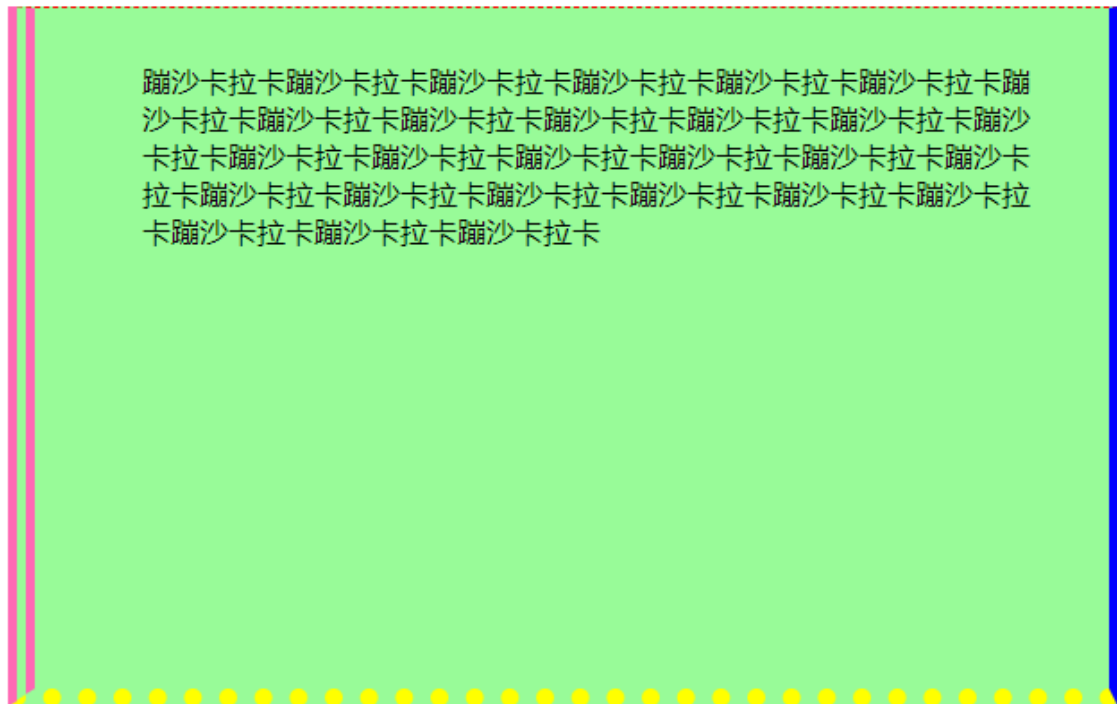
        /* 写法2: 综合写法 */
        /* border: 10px solid yellow; */

        /* 写法3: 按照方位来写 */
        border-top: 1px red dashed;
        border-right: 5px blue solid;
        border-bottom: 10px yellow dotted;
        border-left: 15px hotpink double;
      }
    </style>
  </head>
  <body>
```

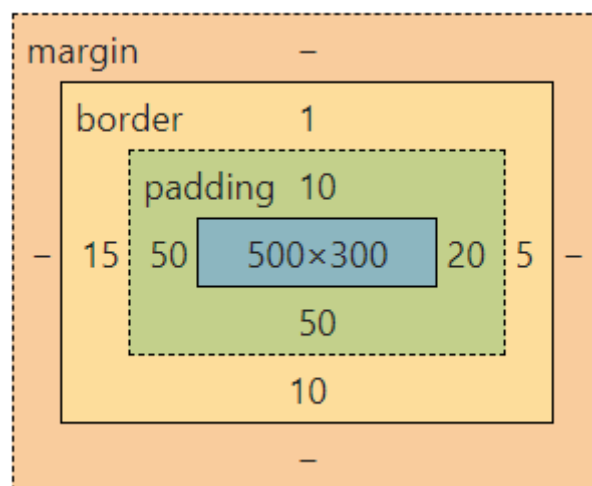
```

<div class="one">
    蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡
    卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡
    蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦沙卡拉卡蹦
    沙卡拉卡蹦沙卡拉卡
    </div>
</body>
</html>

```



浏览器解析盒模型尺寸:

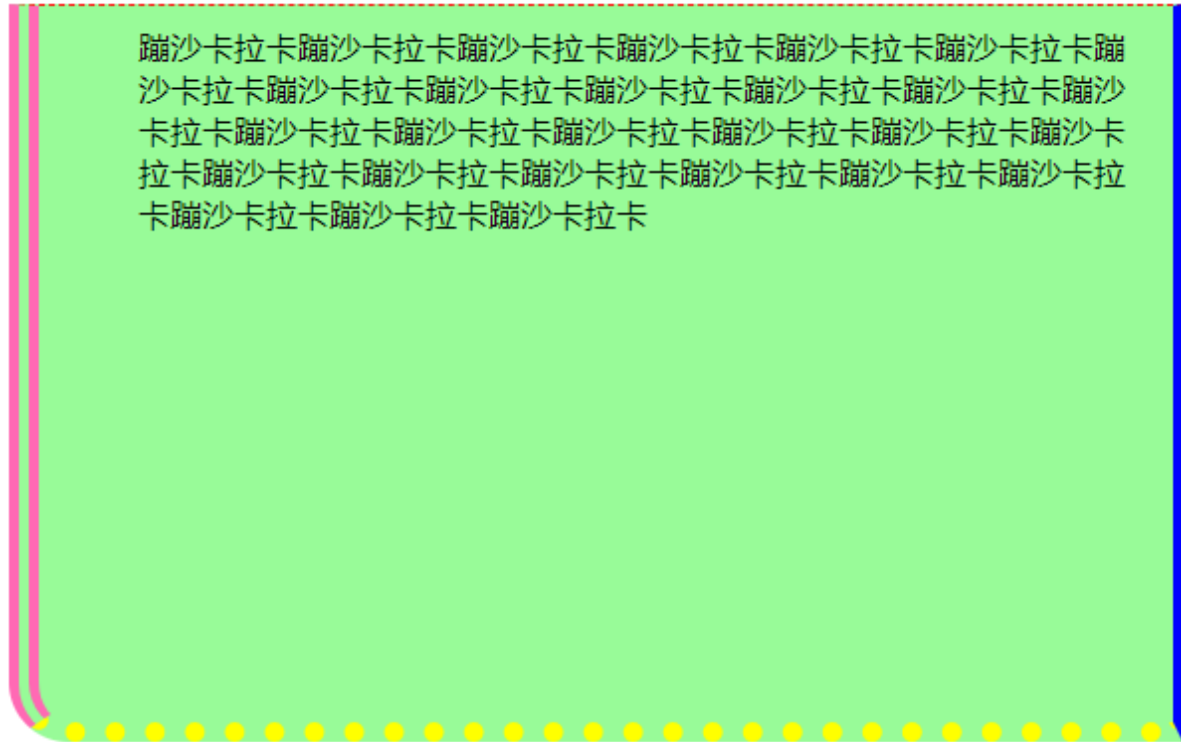


- 边框圆角属性:
 - 此属性只是对边框的样式设置,不影响盒模型尺寸。
 - 如果标签为正方形,border-radius:50%; 可将其切成圆形。


```
/* border圆角设置 */

/* 写法1: 四个圆角 */
/* border-radius: 10px; */

/* 写法2: 按照方位切圆角 */
border-bottom-left-radius: 30px;
```



4.4. 设置margin

```
/* margin */
/* 写法1: 上下左右 */
/* margin: 20px; */


/* 写法2: 上下 左右 */
/* margin: 20px 30px; */

/* 写法3: 上 左右 下 */
/* margin: 20px 30px 0; */

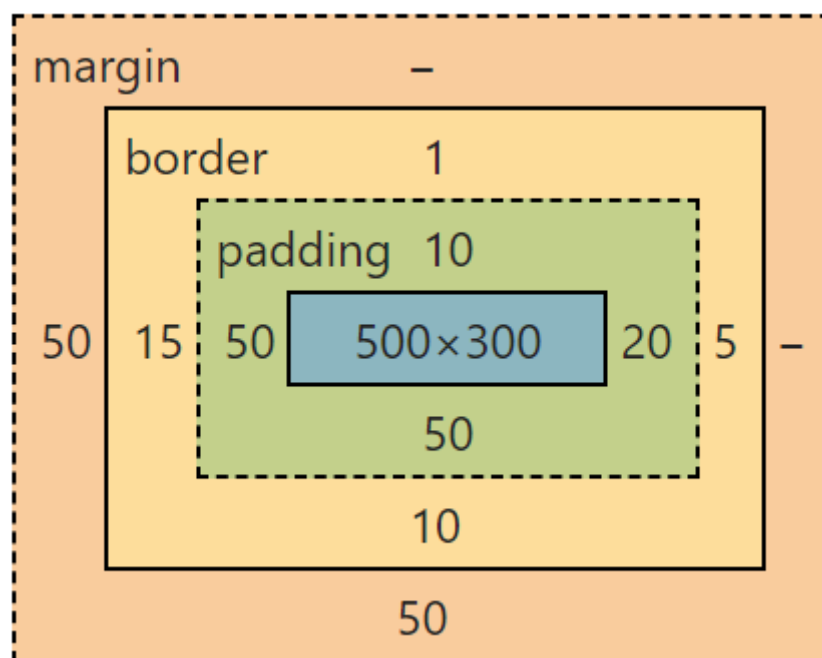
/* 写法4: 上 右 下 左 */
/* margin: 10px 20px 0px 30px; */

/* 写法5: 按照方向 left top right bottom */
margin-bottom: 50px;
margin-left: 50px;
```

蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦
沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙
卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡
拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉
卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡蹦蹦沙卡拉卡



浏览器解析盒模型尺寸:



4.4.1. 块元素位置相对父标签居中

- 就是给其设置 `margin:0 auto;` 。
- 示例

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>01盒模型</title>
    <style type="text/css">
      .one{
        width: 500px;
        height: 300px;
        background-color: palegreen;
```

```
/* padding */

/* 写法1: 上下左右 */
/* padding: 20px; */

/* 写法2: 上下 左右 */
/* padding: 20px 30px; */

/* 写法3: 上 左右 下 */
/* padding: 20px 30px 0; */

/* 写法4: 上 右 下 左 */
/* padding: 10px 20px 0 30px; */

/* 写法5: 按照方向 left top right bottom */
padding-top: 10px;
padding-right: 20px;
padding-bottom: 50px;
padding-left: 50px;
```

```
/* border边框 */
/* 写法1: 单独设置 */
/* border-width: 5px; */ /*宽度*/
/* border-color: red; */ /*颜色*/
/* border-style: solid; */ /*样式: solid 实线; dotted 圆形虚线; dashed 方形虚线; double 双线*/
```

```
/* 写法2: 综合写法 */
/* border: 10px solid yellow; */

/* 写法3: 按照方位来写 */
border-top: 1px red dashed;
border-right: 5px blue solid;
border-bottom: 10px yellow dotted;
border-left: 15px hotpink double;
```

```
/* border圆角设置 */

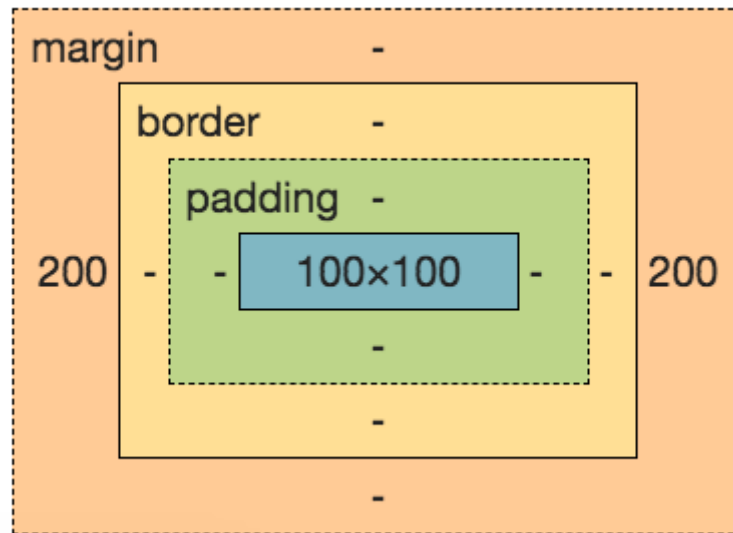
/* 写法1: 四个圆角 */
/* border-radius: 10px; */

/* 写法2: 按照方位切圆角 */
border-bottom-left-radius: 30px;
```

```
/* margin */
/* 写法1: 上下左右 */
/* margin: 20px; */

/* 写法2: 上下 左右 */
/* margin: 20px 30px; */

/* 写法3: 上 左右 下 */
```

4.4.2. margin值在使用时的问题

- 两个元素上下布局的时候, 如果同时设置了上面标签的margin-bottom和下面标签的margin-top,那么两个标签的上下间隔取其中的最大值. 不会累加!。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>01盒模型</title>
    <style type="text/css">
      #p1{
        width: 300px;
        height: 300px;
        background-color: green;
        margin-bottom: 110px;
      }

      #p2{
        width: 500px;
        height: 500px;
        background: hotpink;
        margin-top: 100px;
      }
    </style>
  </head>
  <body>
    <div>
      <p id="p1">1111111111</p>
      <p id="p2">2222222222</p>
    </div>
  </body>
</html>
```



- 两个元素在左右布局的时候,如果同时设置了左边标签的margin-right和右边标签的margin-left,那么两个标签的水平间隔是左边标签的margin-right + 右边标签的margin-left。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style type="text/css">
      span{
        background-color: seagreen;
      }
      #s1{
        margin-right: 100px;
      }
      #s2{
        margin-left: 100px;
      }
    </style>
  </head>
  <body>
    <span id="s1">这是span1</span>
```

```

    <span id="s2">这是span2</span>
  </body>
</html>

```



- 在父子关系中, 第一个子标签也就是one的margin-top值会传递给父标签box;

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>CSS盒模型问题3</title>

    <style>
      *{
        margin: 0;
        padding: 0;
      }
      #box {
        width: 500px;
        height: 300px;
        background-color: palegoldenrod;
        /* border: 1px solid red */ /*阻止盒内容传递外边距给盒子方式1*/
        /* overflow: hidden; */ /*阻止盒内容传递外边距给盒子方式2*/
        /* padding: 50px; */
      }

      #one {
        width: 100px;
        height: 100px;
        background-color: red;
        /* margin: 0 auto; */
        /* margin-top: 100px; */ /*盒内容是没有外边距操作的，外边距操作自动
        传递给盒子进行操作，最终其实就是设置盒子的外边距*/

        /* html的所有元素的位置，默认都是以左上角为准的
        垂直居中的计算方式：（盒子的高度 - 盒内容的高度） / 2      设置给
        margin-top

        */
      }
    </style>

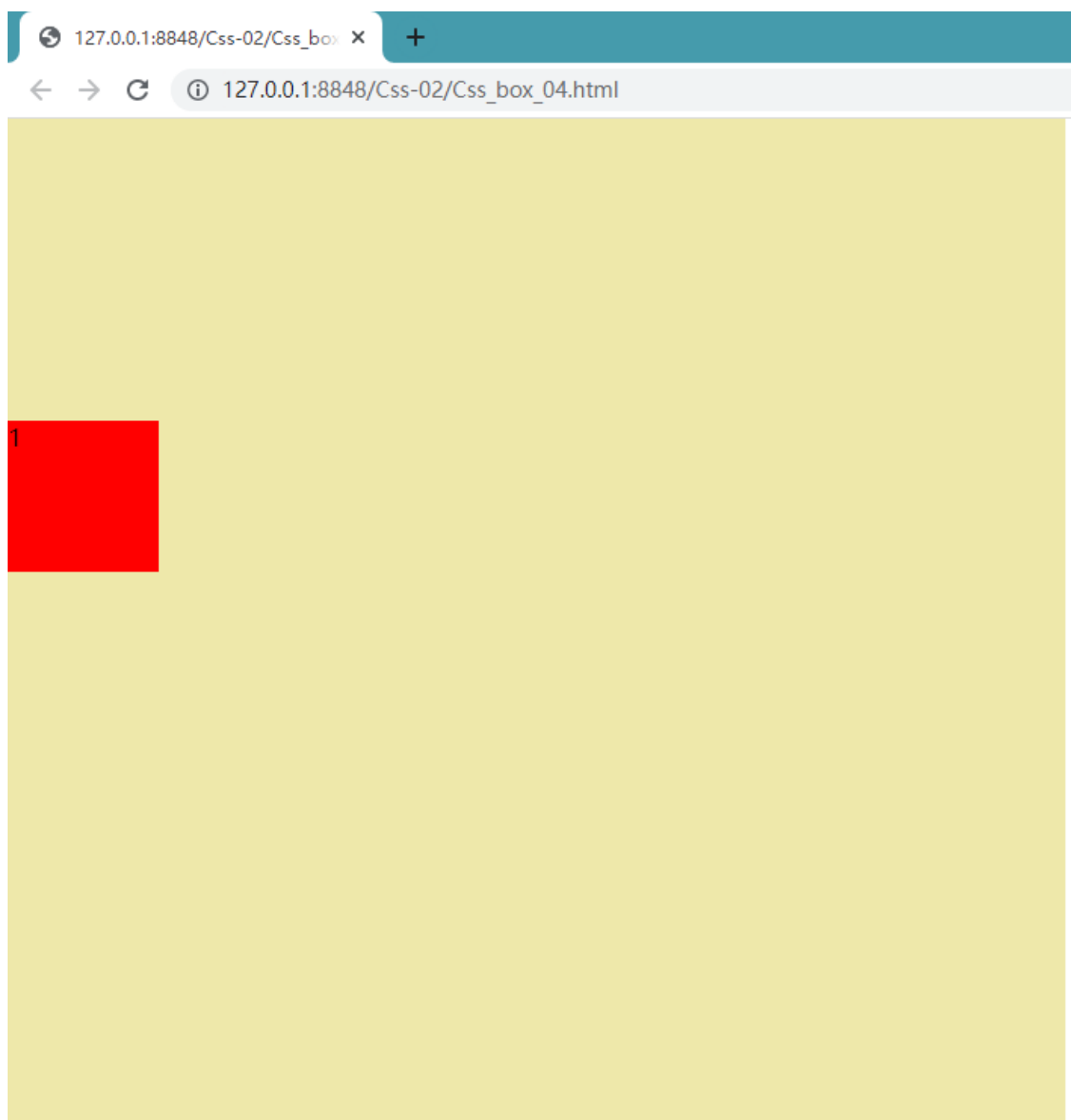
  </head>
  <body>
    <div id="box">
      <!--盒子-->
      <div id="one">1</div>
      <!--盒内容-->
    </div>
  </body>
</html>

```



```
#box{
  width: 700px;
  height: 670px;
  background-color: palegoldenrod;
  /* border: 10px solid black; */ /*父标签可以阻断传递*/
  overflow: hidden; /*父标签可以阻断传递*/
}
#one{
  width: 100px;
  height: 100px;
  background-color: red;
  margin-top: 200px; /*会传递给父标签*/
}

</style>
</head>
<body>
  <div id="box">
    <div id="one">1</div>
  </div>
</body>
</html>
```



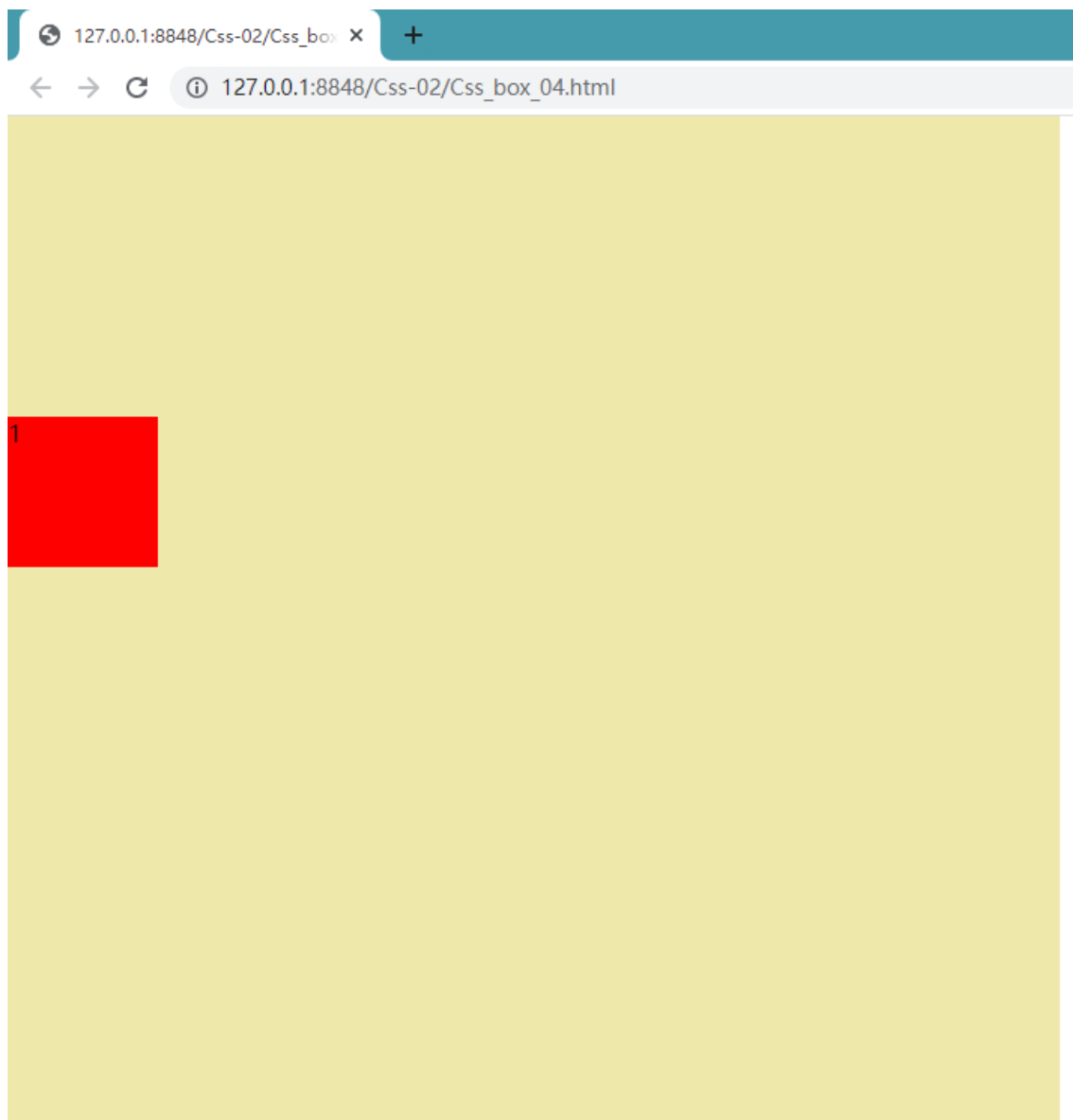
- 如果真的需要父子标签有间隔，推荐给父标签设置 padding内填充!!

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style type="text/css">
      *{
        margin: 0;
        padding: 0;
      }

      #box{
        width: 700px;
        height: 670px;
        background-color: palegoldenrod;
        /* border: 10px solid black; */ /*父标签可以阻断传递*/
        /* overflow: hidden; */ /*父标签可以阻断传递*/
        padding-top: 200px; /*如果真的需要父子标签有间隔，推荐使用padding内
填充*/
      }

      #one{
        width: 100px;
        height: 100px;
        background-color: red;
        /* margin-top: 200px; */ /*会传递给父标签*/
      }

    </style>
  </head>
  <body>
    <div id="box">
      <div id="one">1</div>
    </div>
  </body>
</html>
```



三. 盒模型练习

效果图:



代码如下:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>盒模型练习</title>

    <style>
      *{
        margin: 0;
        padding: 0;
      }
      a{
        text-decoration: none;
```

```

    }
    .nav{
        width: 100%;
        background-color: rgb(0, 0, 0, 0.75);
        text-align: center;
        padding: 5px 0;
    }
    .nav img{
        height: 44px;
    }
    .title{
        padding-top: 6px;
    }
    .title a{
        color: darkgray;
        font-size: 14px;
        margin: 0 30px;
    }
    .title a:hover{
        color: #f2f2f3;
    }
    .title a:first-child{
        color: #f2f2f3;
    }
}
</style>

</head>
<body>
    <div class="nav">
        
        <div class="title">
            <a href="###">HOME</a>
            <a href="###">WORK</a>
            <a href="###">CONTACT</a>
            <a href="###">JOIN</a>
        </div>
    </div>
</body>
</html>

```

四. 标签的显隐性

- opacity: 0; 隐藏 opacity:1; 通过透明度控制显示和隐藏，为标签保留位置；。
- display:none; 隐藏 display:block;显示，不保留标签位置。
- visibility: hidden;隐藏 visibility: visible;显示，为标签保留位置。

1. 准备工作

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>显隐性</title>
    </head>
</html>

```

```
<style type="text/css">

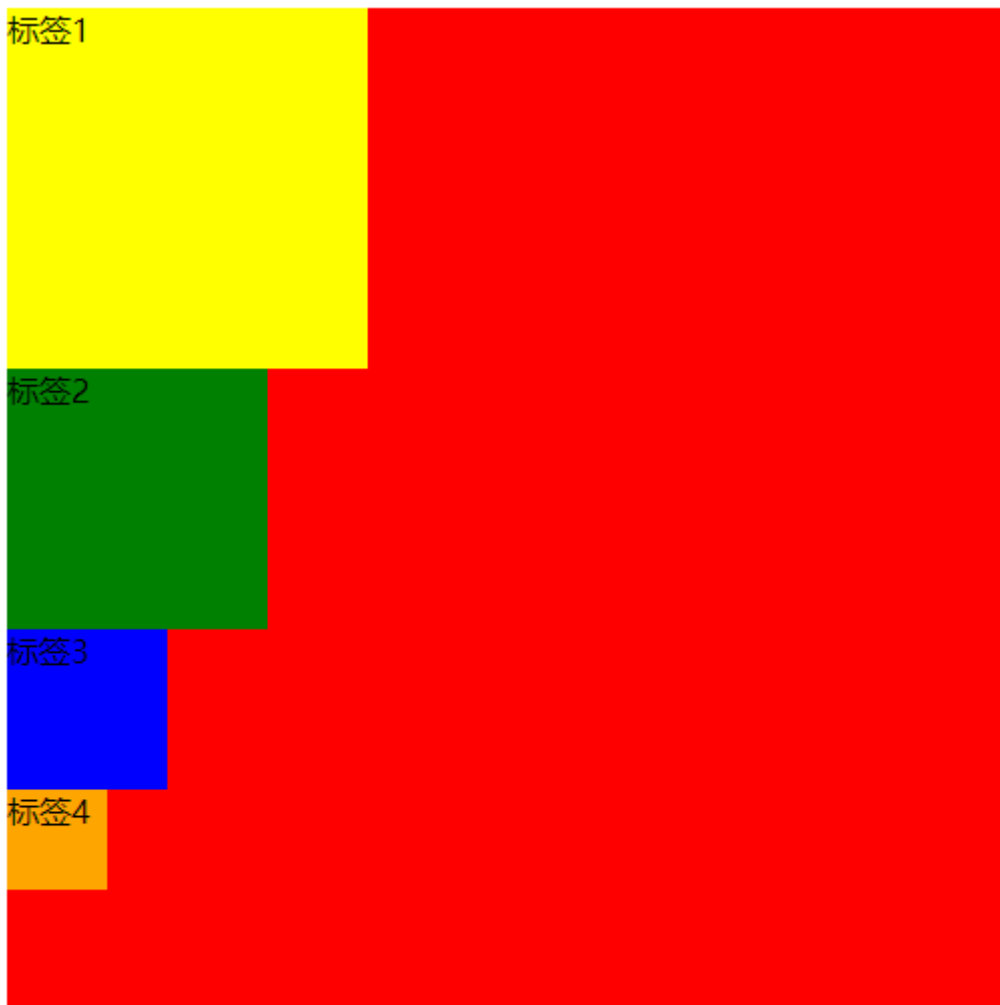
    #box {
        width: 500px;
        height: 500px;
        background-color: red;
    }

    #one {
        width: 180px;
        height: 180px;
        background-color: yellow;
        /* 通过透明度控制隐藏显示 0:隐藏; 1:显示; 0.5:半透明*/
        /* opacity: 0; */
    }

    #two {
        width: 130px;
        height: 130px;
        background-color: green;
        /* 通过display控制隐藏显示 none:隐藏; block:显示*/
        /* display: none; */
    }

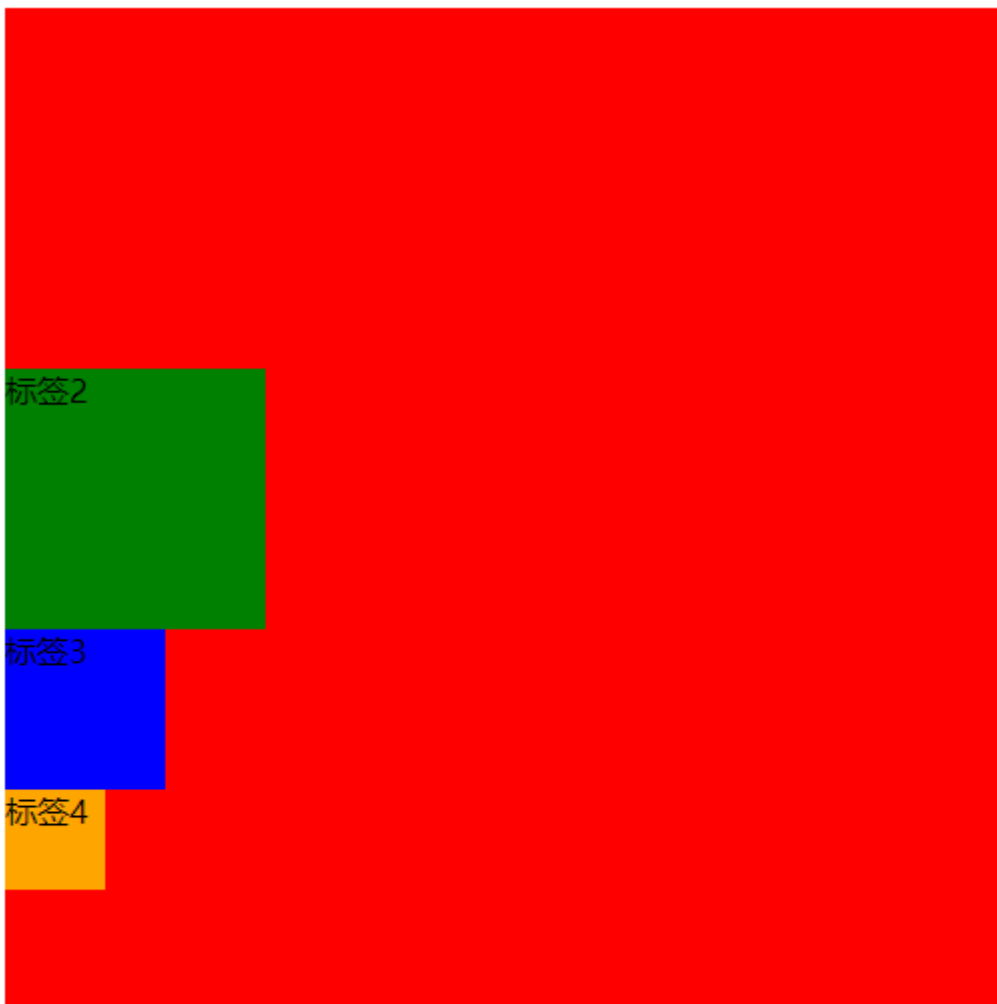
    #three {
        width: 80px;
        height: 80px;
        background-color: blue;
        /* 通过visibility控制隐藏显示 hidden:隐藏; visible:显示*/
        /* visibility: hidden; */
    }

    #four{
        width: 50px;
        height: 50px;
        background-color: orange;
    }
</style>
</head>
<body>
    <div id="box">
        <div id="one">标签1</div>
        <div id="two">标签2</div>
        <div id="three">标签3</div>
        <div id="four">标签4</div>
    </div>
</body>
</html>
```



1.1. 给标签one设置opacity透明度来隐藏

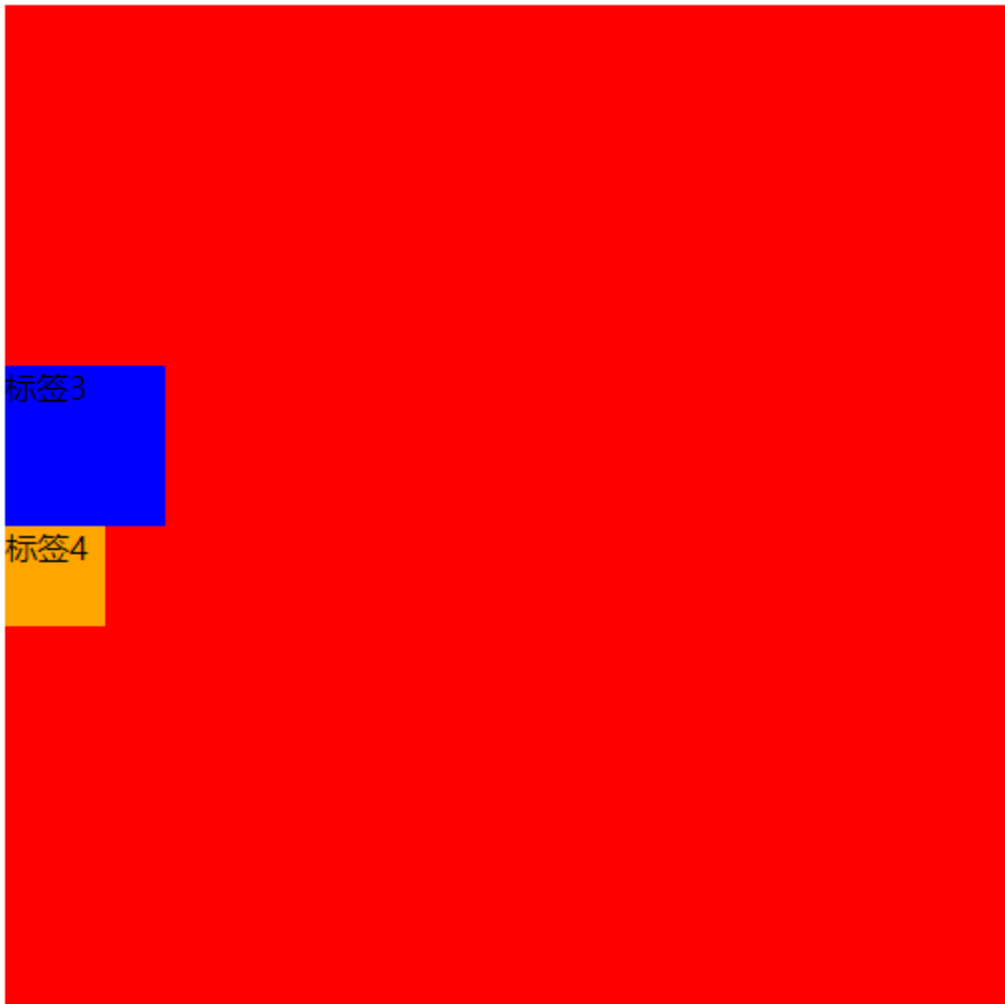
```
#one {  
  width: 180px;  
  height: 180px;  
  background-color: yellow;  
  /* 通过透明度控制隐藏显示 0:隐藏; 1:显示; 0.5:半透明*/  
  opacity: 0;  
}
```



保留标签one的位置

1.2. 给标签two设置display属性来隐藏

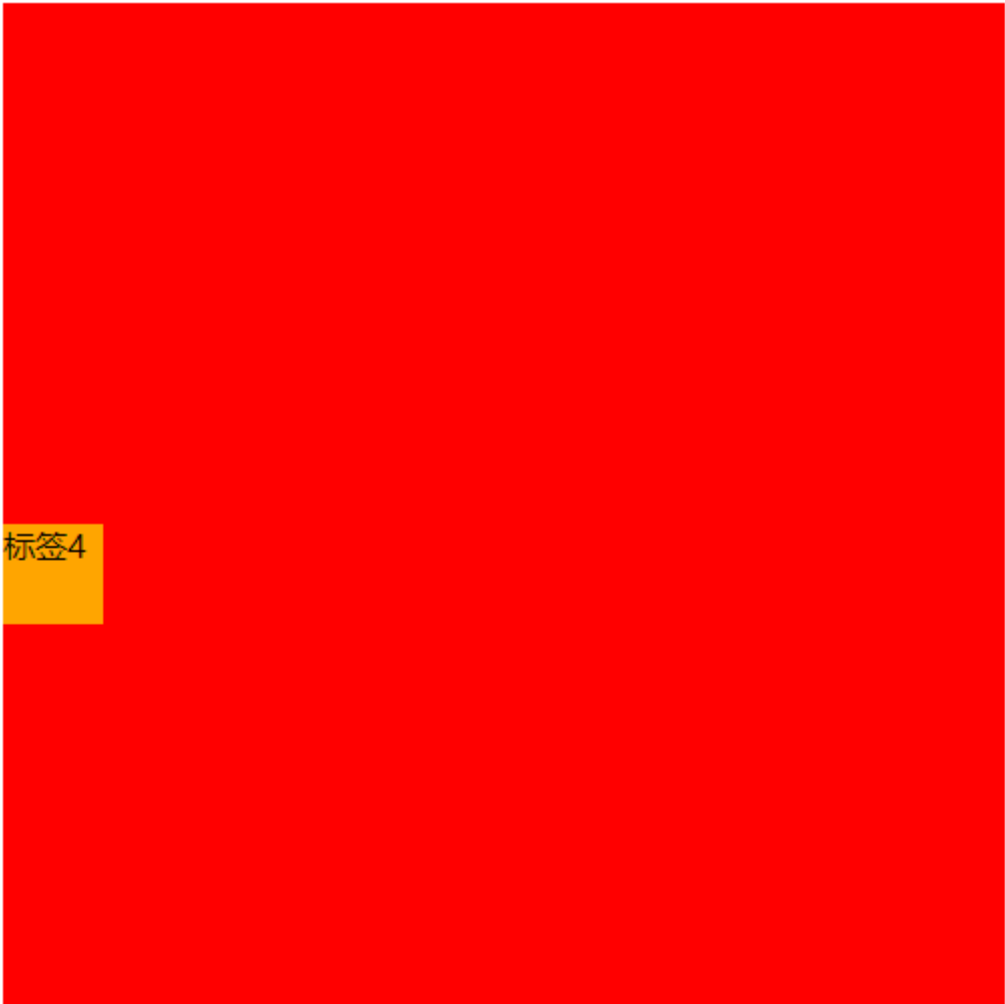
```
#two {  
  width: 130px;  
  height: 130px;  
  background-color: green;  
  /* 通过display控制隐藏显示 none:隐藏; block:显示*/  
  display: none;  
}
```



不保留标签two的位置。

1.3. 给标签three设置visibility属性来隐藏

```
#three {  
  width: 80px;  
  height: 80px;  
  background-color: blue;  
  /* 通过visibility控制隐藏显示 hidden:隐藏; visible:显示*/  
  visibility: hidden;  
}
```

保留标签three的位置。

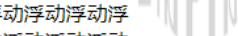
五. 浮动

CSS 的 Float（浮动），会使元素向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止，其周围的元素也会重新排列。

1. 元素向右浮动

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>浮动</title>
    <style>
      img{
        width: 95px;
        height: 84px;
        float: right;
      }
    </style>
  </head>
  <body>
    
    <p>
      浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动
    </p>
  </body>
</html>
```

浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动 浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动
浮动浮动 浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动 浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动
浮动浮动浮动 浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动 浮动浮动浮动浮动浮动浮动浮动浮动浮动浮
动浮动浮动浮动浮动 浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动 浮动浮动浮动浮动浮动浮动浮动
浮动浮动浮动浮动浮动浮动 浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动浮动 浮动浮动浮动浮动浮动浮动
浮动浮动浮动浮动浮动浮动

[illegible]

[illegible]

第一行



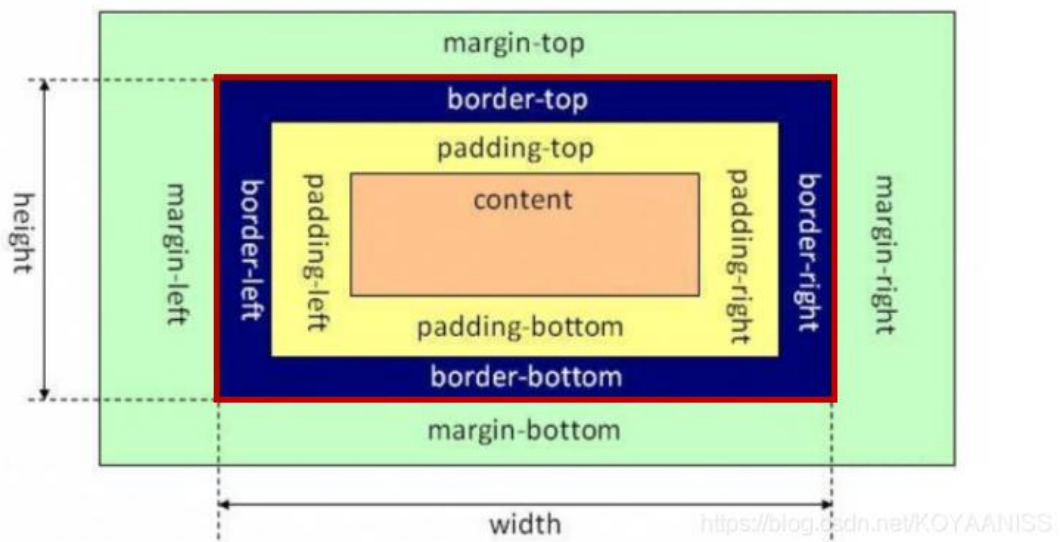
第二行



六. 怪异盒模型

- box-sizing属性,也可以设置值为border-box 叫做怪异盒模型。
- 怪异盒模型的宽度是: 内容区域width+margin左右外间距, 如果怪异盒模型标签有设置padding和border则会占用已设定的宽度值, 不会再向外扩展盒模型的尺寸。
- 怪异盒模型的高度同宽度计算方法一致。

```
选择器{  
    box-sizing: border-box;  
}
```



- 示例

标准盒子及模型宽度图如下:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>怪异盒子模型</title>  
    <style>  
      .one{  
        width: 500px;
```