

第 7 天【float 和 position】

主要内容

- 1、Float 的原理
- 2、Float 的语法
- 3、清除浮动的影响
- 4、position 定位
- 5、display 属性

学习目标

节数	知识点	要求
第一节 (float)	Float 的原理	掌握
	Float 的语法	掌握
	清除浮动的影响	掌握
第二节 (浮动练习)	金立官网布局	掌握
第三节 (position)	概念	了解
	Relative	掌握
	Absolute	掌握
	Fixed	掌握
第四节 (position 练习)	下拉列表效果	掌握
	两张图片错开效果	掌握
	图片上面定位文字	掌握
	元素从下面出来效果	掌握
	照片墙	掌握
第五节 (display)	display 属性	掌握

一、Float

1、浮动的定义



Float 属性定义元素在哪个方向浮动。以往这个属性总应用于图像，使文本围绕在图像周围，不过在 CSS 中，任何元素都可以浮动。**浮动元素会生成一个块级框**，而不论它本身是何种元素。

CSS 的 Float（浮动）使元素**脱离文档流**，按照指定的方向（左或右发生移动），**直到它的外边缘碰到包含框或另一个浮动框的边框为止。**

说到脱离文档流要说一下什么是文档流，文档流是是文档中可显示对象在排列时所占用的位置/空间，而**脱离文档流就是在页面中不占位置了。**

【实例 1-1】

```
<div class="box">
```

```

```

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Magni quo laboriosam iusto repellat cupiditate accusamus reprehenderit suscipit officiis quas dolore non sed, temp commodi, repellendus, exercitationem volupta provident?

```
</div>
```

```
img{
```

```
width: 200px;
```

```
float: left;
```

```
}
```

效果如下：



Lorem ipsum dolor sit amet consectetur, adipisicing elit.

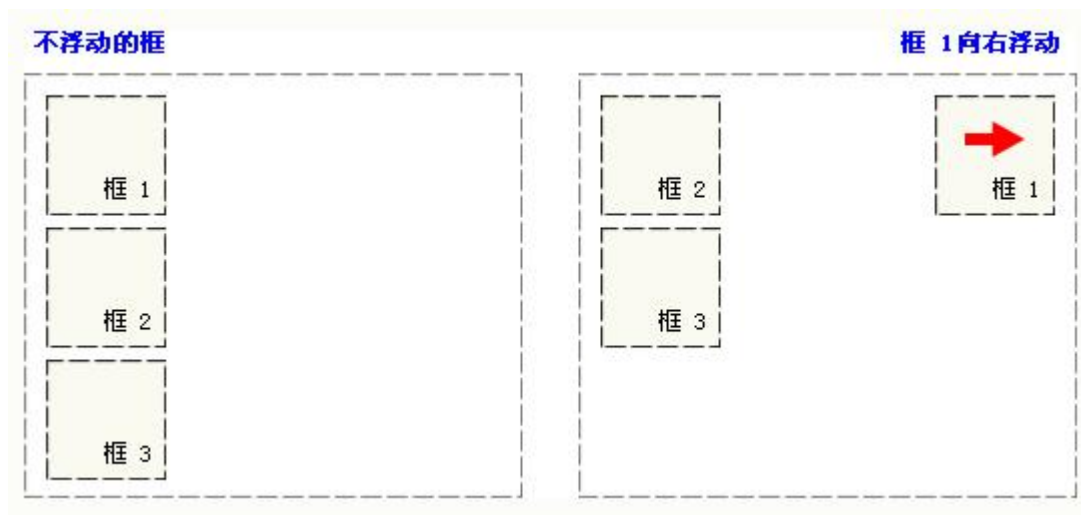
Magni quo laboriosam iusto quae repellat cupiditate accusamus reprehenderit sit suscipit officiis quas dolore non sed, tempore commodi, repellendus, exercitationem voluptatum provident? Lorem ipsum dolor sit amet consectetur, adipisicing elit. Magni quo laboriosam iusto quae repellat cupiditate accusamus reprehenderit sit suscipit officiis quas dolore non sed, tempore commodi, repellendus, exercitationem voluptatum provident? Lorem ipsum dolor sit amet consectetur, adipisicing elit. Magni

quo laboriosam iusto quae repellat cupiditate accusamus reprehenderit sit suscipit officiis quas dolore non sed, tempore commodi, repellendus, exercitationem voluptatum provident? Lorem ipsum dolor sit amet consectetur, adipisicing elit. Magni quo laboriosam iusto quae repellat cupiditate accusamus reprehenderit sit

2、浮动的原理

- (1) 浮动以后使元素脱离了文档流（在页面中不占据位置）
- (2) 浮动是碰到父元素的边框或者浮动元素的边框就会停止
- (3) 浮动不会重叠
- (4) 浮动只有左右浮动，没有上下浮动
- (5) 浮动以后块级元素在同一行显示，行内元素可以设置宽高
- (6) 元素没有设置宽度和高度时，宽度为内容撑开

请看下图，当把框 1 向右浮动时，它脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘：

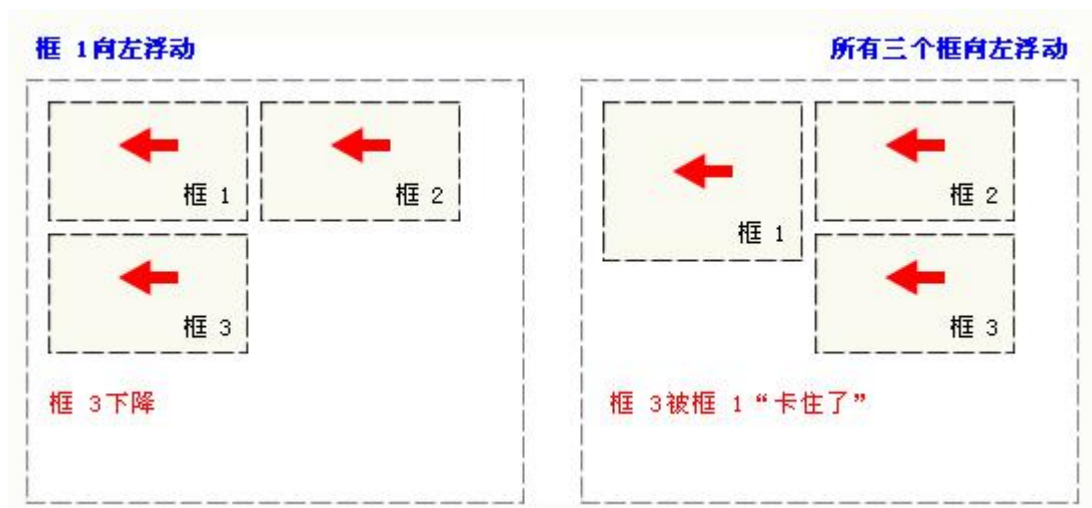


再请看下图，当框 1 向左浮动时，它脱离文档流并且向左移动，直到它的左边缘碰到包含框的左边缘。因为它不再处于文档流中，所以它不占据空间，实际上覆盖住了框 2，使框 2 从视图中消失。

如果把所有三个框都向左移动，那么框 1 向左浮动直到碰到包含框，另外两个框向左浮动直到碰到前一个浮动框。



如下图所示，如果包含框太窄，无法容纳水平排列的三个浮动元素，那么其它浮动块向下移动，直到有足够的空间。如果浮动元素的高度不同，那么当它们向下移动时可能被其它浮动元素“卡住”：



3、浮动的语法

可能的值

值	描述
left	元素向左浮动。
right	元素向右浮动。
none	默认值。元素不浮动，并会显示在其在文本中出现的位置。

4、清除浮动的影响

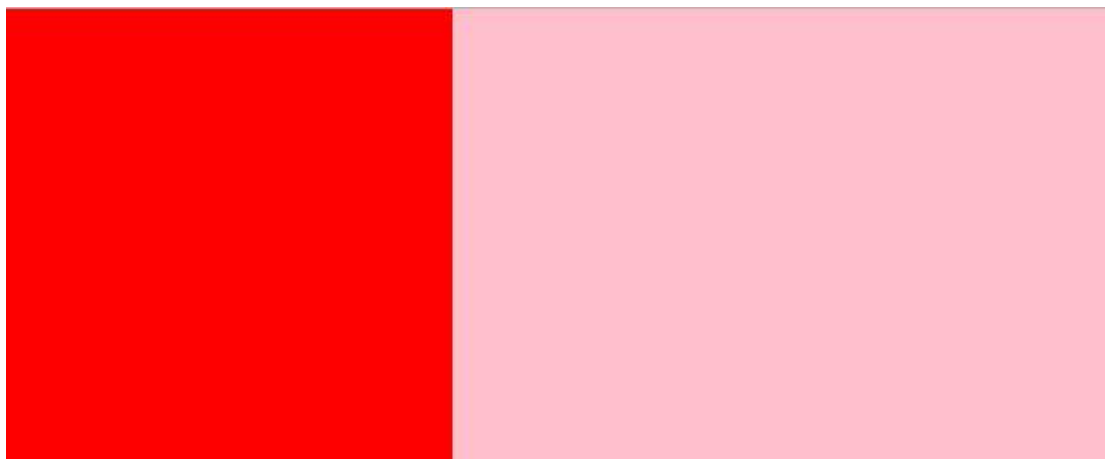
当元素设置 float 浮动后，该元素就会脱离文档流并向左/向右浮动，直到碰到父元素或者另一个浮动元素，浮动元素会造成父元素高度塌陷。

【实例 1-2】

```
<div class="box">
  <div class="one"></div>
</div>
<style>
  .box{
    background-color: pink;
  }
  .one{
    width: 200px;
    height: 200px;
    background-color: red;
  }
</style>
```

```
}
</style>
```

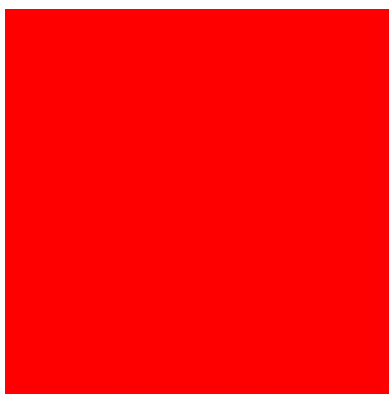
子元素.one 没加浮动时的样式如下：



父元素高度由子元素撑开

```
<style>
    .box{
        background-color: pink;
    }
    .one{
        width: 200px;
        height: 200px;
        background-color: red;
        float: left;
    }
</style>
```

给子元素设置浮动后效果如下：



父元素的背景颜色消失，高度为0

所以当我们设置 float 后，要根据不同情况来清除浮动。下面介绍几种常用的方法：

(1) 设置父布局的高度

设置父标签合适的高度，前提必须确定子布局的高度，来计算父布局的合适高度，包裹住子布局。

【实例 1-3】

```
<div class="box">
  <div class="one"> </div>
  <div class="two"> </div>
  <div class="three"> </div>
</div>
```

```
<style>
  .box{
    background-color: pink;
    height: 200px;
  }
  .one,.two,.three{
    width: 200px;
    height: 200px;
    float: left;
  }
  .one{
    background-color: red;
  }
```



```
.two{
    background-color: gold;
}
.three{
    background-color: green;
}
</style>
```

(2) 受影响的元素加 clear 属性

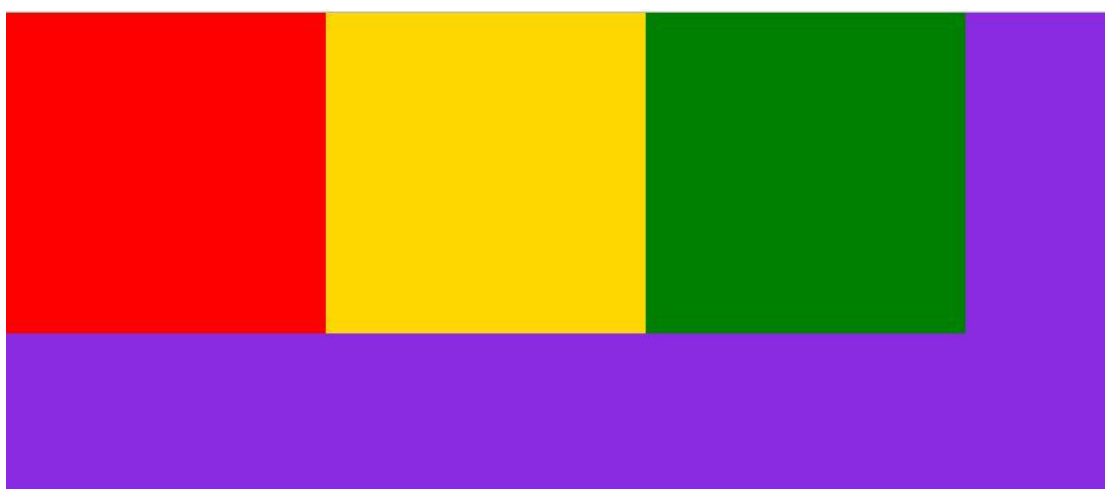
clear:left | right | both;

【实例 1-4】

```
<div class="box">
    <div class="one"> </div>
    <div class="two"> </div>
    <div class="three"> </div>
</div>
<div class="footer"> </div>
<style>
.box{
    background-color: pink;
}
.one,.two,.three{
    width: 200px;
    height: 200px;
    float: left;
}
}
```



```
.one{
  background-color: red;
}
.two{
  background-color: gold;
}
.three{
  background-color: green;
}
.footer{
  height: 300px;
  background-color: blueviolet;
  clear:both;
}
</style>
```

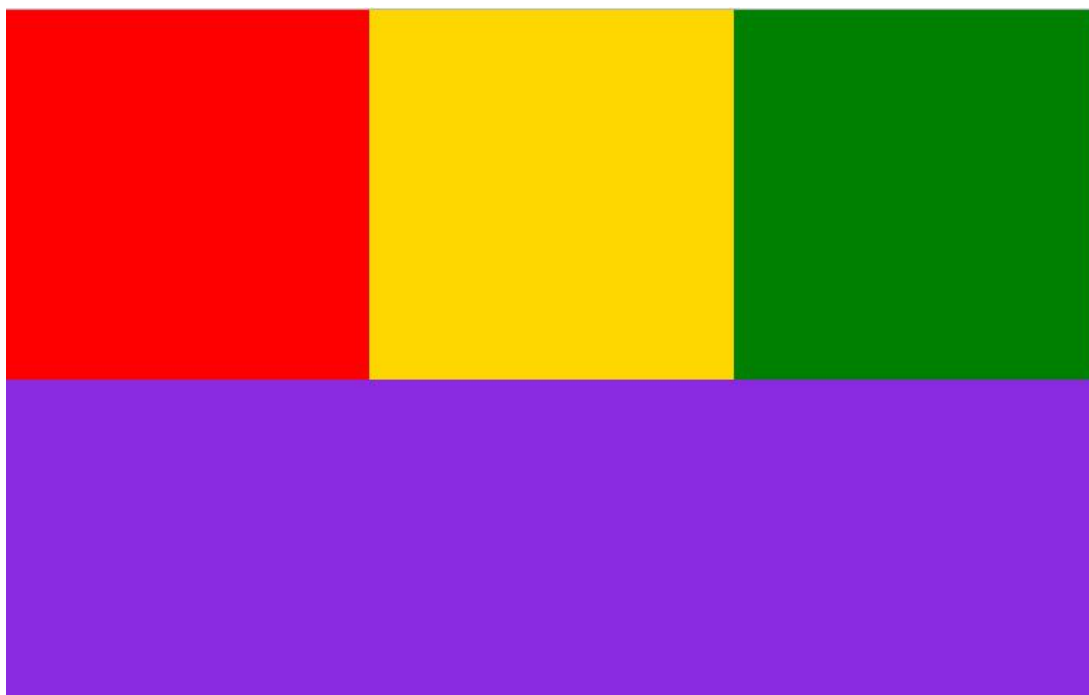


清除浮动之前，由于 box 的三个子元素都加了浮动，排除到文档流之外，box 的高度坍塌，所以下面的 div 被覆盖住了。

给受影响的 div 加 clear，清除浮动的影响

```
.footer{
  height: 300px;
  background-color: blueviolet;
  clear: both;
}
```

清除完效果如下：



(3) overflow 清除浮动

这种情况下，父布局不能设置高度。

父级标签的样式里面加: overflow:hidden;

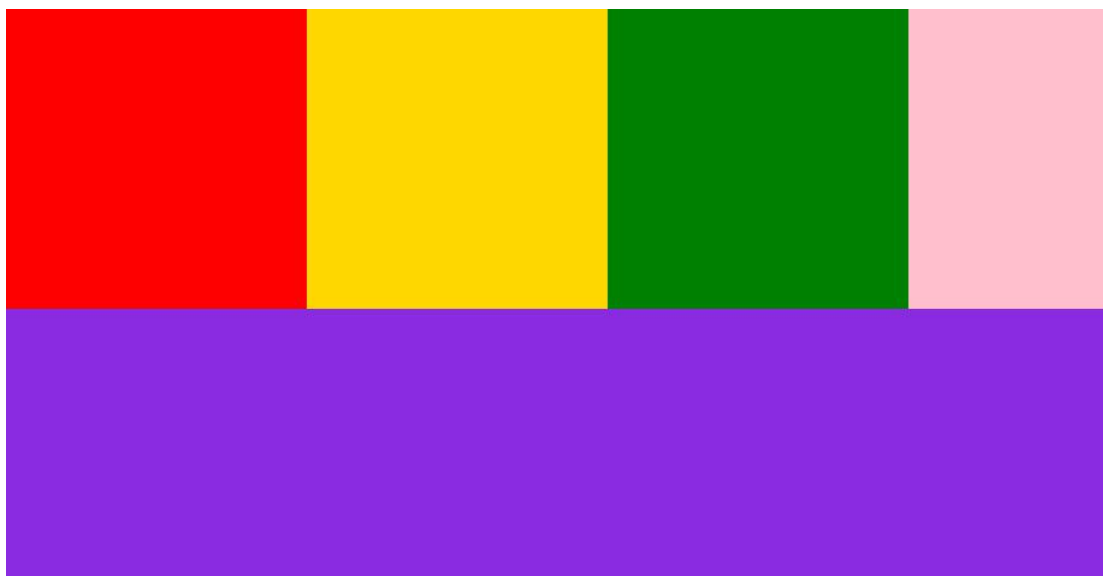
【实例 1-5】

```
<div class="box">
  <div class="one"> </div>
  <div class="two"> </div>
  <div class="three"> </div>
```

```
</div>
<div class="footer"> </div>
<style>
.box{
    background-color: pink;
    overflow: hidden;
}
.one,.two,.three{
    width: 200px;
    height: 200px;
    float: left;
}
.one{
    background-color: red;
}
.two{
    background-color: gold;
}
.three{
    background-color: green;
}
.footer{
    height: 300px;
    background-color: blueviolet;
}
```

```
}  
</style>
```

效果如下：



(4) 空 div 法

在最后一个浮动的盒子的后面，新添加一个标签。然后设置 clear 清除浮动。

这种情况下，父布局不能设置高度。

优点：通俗好理解。

缺点：增加了太多的标签。

【实例 1-6】

```
<div class="box">  
  <div class="one"> </div>  
  <div class="two"> </div>  
  <div class="three"> </div>  
  <div class="clear"> </div>  
</div>
```

```
<div class="footer"> </div>
```

```
.clear{
    clear: both;
}
```

(5) 伪对象发

为父标签添加伪类 After,设置空的内容,并且使用 clear:both;

这种情况下,父布局不能设置高度。

优点: 无需添加多余的标签,并且可以全局调用。

【实例 1-7】

```
<div class="box">
    <div class="one"> </div>
    <div class="two"> </div>
    <div class="three"> </div>
</div>
<div class="footer"> </div>
.box::after{
    content: "";
    display: block;
    clear: both;}
```

二、Position

1、定义

position 属性指定了元素的定位类型。

这个属性定义建立元素布局所用的定位机制。任何元素都可以定位,不过绝对或固定元素会生成一个块级框,而不论该元素本身是什么类型。相对定位元素会相对于它在正常流中的默认位置偏移。

元素可以使用的顶部,底部,左侧和右侧属性定位。然而,这些属性无法工作,除

非是先设定 position 属性。他们也有不同的工作方式，这取决于定位方法。

2、取值

值	描述
absolute	生成绝对定位的元素，相对于 static 定位以外的第一个父元素进行定位。 元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。
fixed	生成绝对定位的元素，相对于浏览器窗口进行定位。 元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。
relative	生成相对定位的元素，相对于其正常位置进行定位。 因此，"left:20" 会向元素的 LEFT 位置添加 20 像素。
static	默认值。没有定位，元素出现在正常的流中（忽略 top, bottom, left, right 或者 z-index 声明）。

position:sticky

position:sticky 是一个新的 css3 属性，它的表现类似 position:relative 和 position:fixed 的合体，在目标区域在屏幕中可见时，它的行为就像 position:relative; 而当页面滚动超出目标区域时，它的表现就像 position:fixed，它会固定在目标位置。

另外需要注意的是，如果同时定义了 left 和 right 值，那么 left 生效，right 会无效，同样，同时定义了 top 和 bottom，top 赢~~

【实例】

```
<body>
  <div class="header"></div>
  <div class="nav"></div>
</body>
```

```
<style>
  *{
    margin: 0;
    padding: 0;
  }
  body{
    height: 3000px;
  }
  .header{
    width: 100%;
    height: 45px;
    background-color: #333;
  }
  .nav{
    width: 100%;
    height: 75px;
    background-color: indianred;
    position: sticky;
    left: 0;
    top: 0;
  }
</style>
```

3、Z-index

z-index 属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面

auto 默认。堆叠顺序与父元素相等。

注释：元素可拥有负的 z-index 属性值。

注释：Z-index 仅能在定位元素上奏效（例如 position:absolute;）

该属性设置一个定位元素沿 z 轴的位置，z 轴定义为垂直延伸到显示区的轴。如果为正数，则离用户更近，为负数则表示离用户更远。

【实例 2-1】

```
<div class="box">
<div class="one"></div>
```



```
<div class="two"></div>
```

```
</div>
```

```
<style>
```

```
.box{
```

```
    position: relative;
```

```
}
```

```
.one{
```

```
    width: 200px;
```

```
    height: 200px;
```

```
    background-color: red;
```

```
    position: absolute;
```

```
    top: 50px;
```

```
    left: 50px;
```

```
    z-index: 666;
```

```
}
```

```
.two{
```

```
    width: 200px;
```

```
    height: 200px;
```

```
    background-color: pink;
```

```
    position: absolute;
```

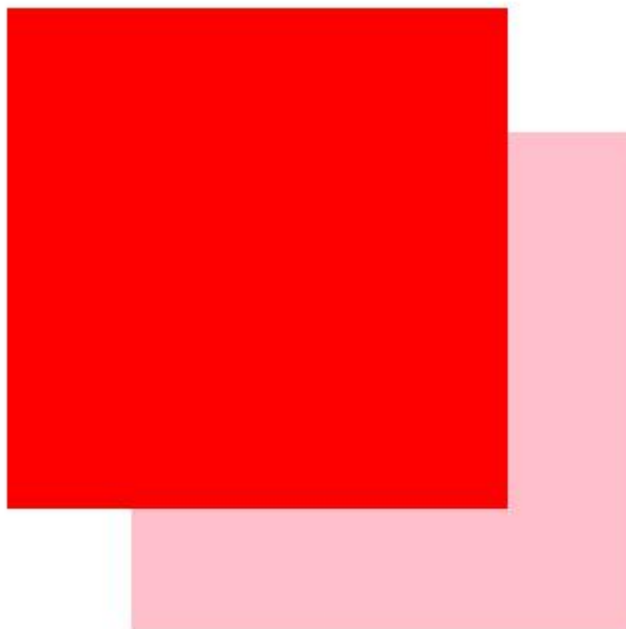
```
    Left: 100px;
```

```
    top: 100px;
```

```
}
```

```
</style>
```

效果如下：



4、position 练习

4.1 下拉列表效果

```
<div class="nav">  
<ul>  
  <li><a href="#">首页</a></li>  
  <li><a href="#">手机</a>  
    <div class="nav-list"></div>  
  </li>  
  <li><a href="#">配件</a></li>  
  <li><a href="#">服务</a></li>  
  <li><a href="#">下载</a></li>
```

```
<li><a href="#">amigoOS</a></li>

</ul>

</div>
```

```
<style>

.nav{

    width: 100%;

    height: 61px;

    position: relative;

}

.nav li{

    float: left;

    padding: 20px 26px;

}

.nav a{

    color: #000;

}

.nav-list{

    width: 100%;

    height: 300px;

    background-color: red;

    position: absolute;

    left: 0;

    top: 100%;

}
```

```
display: none;
}
.nav li:hover>.nav-list{
display: block;
}
</style>
```

鼠标悬停在手机上，下面的 div 元素显示出来

首页

手机

配件

服务

下载

amigoOS



4.2 两张图片错开效果

```
<div class="box">
<div class="img-box">


</div>
<p class="name">金立 F6</p>
```

<p class="desc">5.7 英寸高清全面屏，四曲面机身</p>

<p class="price">¥1399</p>

</div>

```
.box{  
  width: 290px;  
  height: 390px;  
  background-color: #f2f4f5;
```

```
}
```

```
.img-box{  
  width: 290px;  
  height: 240px;  
  position: relative;
```

```
}
```

```
.img-box>img{  
  width: 120px;  
  position: absolute;  
  top:0;  
  left: 85px;  
  transition: all .5s
```

```
}
```

```
.box:hover .left{
```

```
left: 55px;
}
.box:hover .right{
left: 115px;
}
.name{
color: #0f0e0e;
height: 24px;
line-height: 24px;
margin-top: 10px;
}
.desc{
color: #9e9e9e;
font-size: 14px;
height: 24px;
line-height: 24px;
margin-top: 3px;
}
.price{
color: #fe6a00;
font-size: 24px;
margin-top: 12px;
}
.price>span{
```

```
font-size: 18px;
margin-right: 2px;
}
```

效果如下：



4.3 图片上面定位文字

```
<div class="box">

<div class="text">
<p class="name">线控自拍杆</p>
<p class="desc">迷你携带，线控美拍</p>
```



```
<p class="price"><span>¥</span>49</p>
```

```
</div>
```

```
</div>
```

```
.box{  
  width: 386px;  
  height: 343px;  
  position: relative;  
}  
.text{  
  width: 100%;  
  height: 86px;  
  position: absolute;  
  left: 0;  
  bottom: 21px;  
}  
.name{  
  color: #0f0e0e;  
  height: 24px;  
  line-height: 24px;  
}  
.desc{
```

```
color: #9E9E9E;

font-size: 14px;

height: 24px;

line-height: 24px;

margin-top: 3px;
}

.price{

color: #f06261;

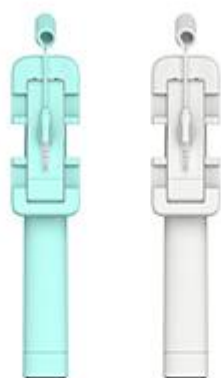
font-size: 24px;

margin-top: 2px;
}

.price>span{

font-size: 18px;

margin-right: 2px;
}
```



热卖

线控自拍杆

迷你携带，线控美拍

¥49

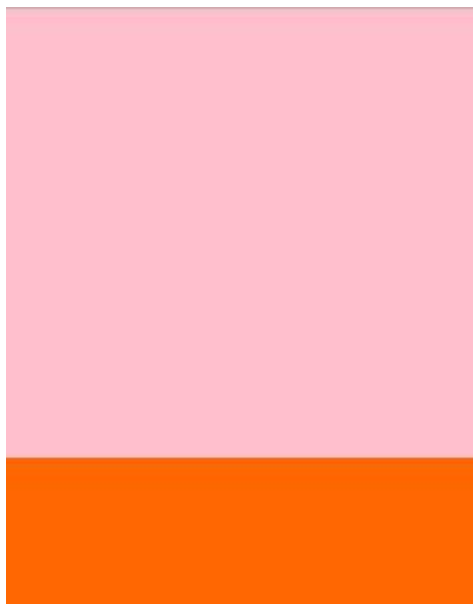
4.4 元素从下面出来效果

```
<div class="box">  
<div class="hidden"> </div>  
</div>
```

```
.box{  
  width: 234px;  
  height: 300px;  
  background-color: #fff;  
  position: relative;  
  overflow: hidden;  
}  
.hidden{
```

```
width: 100%;
height: 76px;
background-color: #ff6700;
position: absolute;
top: 100%;
left: 0;
transition: all .5s;
}
.box:hover>.hidden{
top: 224px;
}
```

鼠标悬停在 box 上，.hidden 从下面出来
效果如下：



4.5 照片墙

```
<div class="box">
```

```


</div>
```

```
.box{
    width: 960px;
    height: 450px;
    margin: 0 auto;
    position: relative;
}

.box>img{
    width: 210px;
    border: 1px solid #ddd;
    padding: 10px;
    background-color: #fff;
    position: absolute;
    transition: all .5s;
}

.img1{
    top: 0;
    left: 375px;
    transform: rotate(5deg);
}

.box>img:hover{
    box-shadow: 10px 10px 10px rgba(0,0,0,.1);
```

```
transform: scale(1.1);  
  
z-index: 666;  
  
}
```



三、display

1、定义

display 属性规定元素应该生成的框的类型。

这个属性用于定义建立布局时元素生成的显示框类型。对于 HTML 等文档类型，如果使用 display 不谨慎会很危险，因为可能违反 HTML 中已经定义的显示层次结构。

2、取值

值	描述
none	此元素不会被显示。
block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
inline-block	行内块元素。

table-cell	此元素会作为一个表格单元格显示（类似 <td> 和 <th>）
Flex	弹性盒模型

3、display:none;和 visibility:hidden;和 opacity:0;和 overflow:hidden;

- 1) display:none; 隐藏自己，隐藏后原位置不保留
- 2) visibility:hidden; 隐藏自己，隐藏后原位置保留
- 3) opacity:0; 隐藏自己，隐藏后原位置保留
- 4) overflow:hidden; 溢出部分隐藏

4、原位置不保留的属性

- 1) float;
- 2) position:absolute;
- 3) position:fixed;
- 4) display:none;

四、BFC


BFC (Block formatting context) 直译为“块级格式化上下文”。他是一个独立渲染的区域它规定了内部如何布局，容器里面的子元素不会在布局上影响到外面的元素。

如何产生 BFC ?

- 1、float 取值 left 或者 right
- 2、Position 取值为 absolute 或者 fixed
- 3、Overflow 的值不是 visible
- 4、Display 的值是 inline-block、table-cell、flex 等

五、作业

<https://www.gionee.com/>


[首页](#)
[手机](#)
[配件](#)
[服务](#)
[下载](#)
[GioneeOS](#)
[GioneeOS](#)

金立手机 · 拍出最美的你

— 摄影作品展 —

明星产品

金立S11

超薄全面屏 前置摄像头 4G+64GB

立即购买

金立S10

超薄全面屏 前置摄像头 4G+64GB

立即购买

金立M7

6.81英寸超薄全面屏 前置摄像头 4G+64GB

立即购买

热卖推荐

金立S11

四摄全面屏 更美更清晰

¥1899

金立P6

5.7英寸超薄全面屏 前置摄像头

¥1399

S10C

5.5英寸超薄全面屏 前置摄像头

¥1399

大金M6

6.81英寸超薄全面屏 前置摄像头

¥1799

金立S9

5.5英寸超薄全面屏 前置摄像头

¥1199

金立M6S Plus

6.0英寸超薄全面屏 前置摄像头

¥3299 起

金立S9

5.5英寸超薄全面屏 前置摄像头

¥1699

精美配件

快充数据线

快充数据线 快充数据线

¥49

Micro USB数据线

Micro USB数据线 快充数据线

¥20

旅行充电器

旅行充电器 快充数据线

¥40

快充充电器

快充充电器 快充数据线

¥69

蓝牙耳机

蓝牙耳机 蓝牙耳机

¥69

贴膜

贴膜 贴膜

¥10

品牌动态

科技微生活

金立影像艺术

金立S11S产品视频

7天 退换货

30天 质保承诺

99% 好评率

57家 售后服务点

关于金立

品牌历史

品牌理念

品牌文化

品牌荣誉

品牌合作

品牌发展

品牌未来

联系我们

联系我们

联系我们

联系我们

联系我们

联系我们

联系我们

金立手机

金立手机

金立手机

金立手机

金立手机

金立手机

金立手机