

# Web开发一

## 第十一章 网页布局

# 本讲目标



- ▶ 布局概念、布局常见版式
- ▶ 理解布局思想
- ▶ 掌握布局的方法
- ▶ 掌握相对定位、绝对定位
- ▶ 熟练使用布局

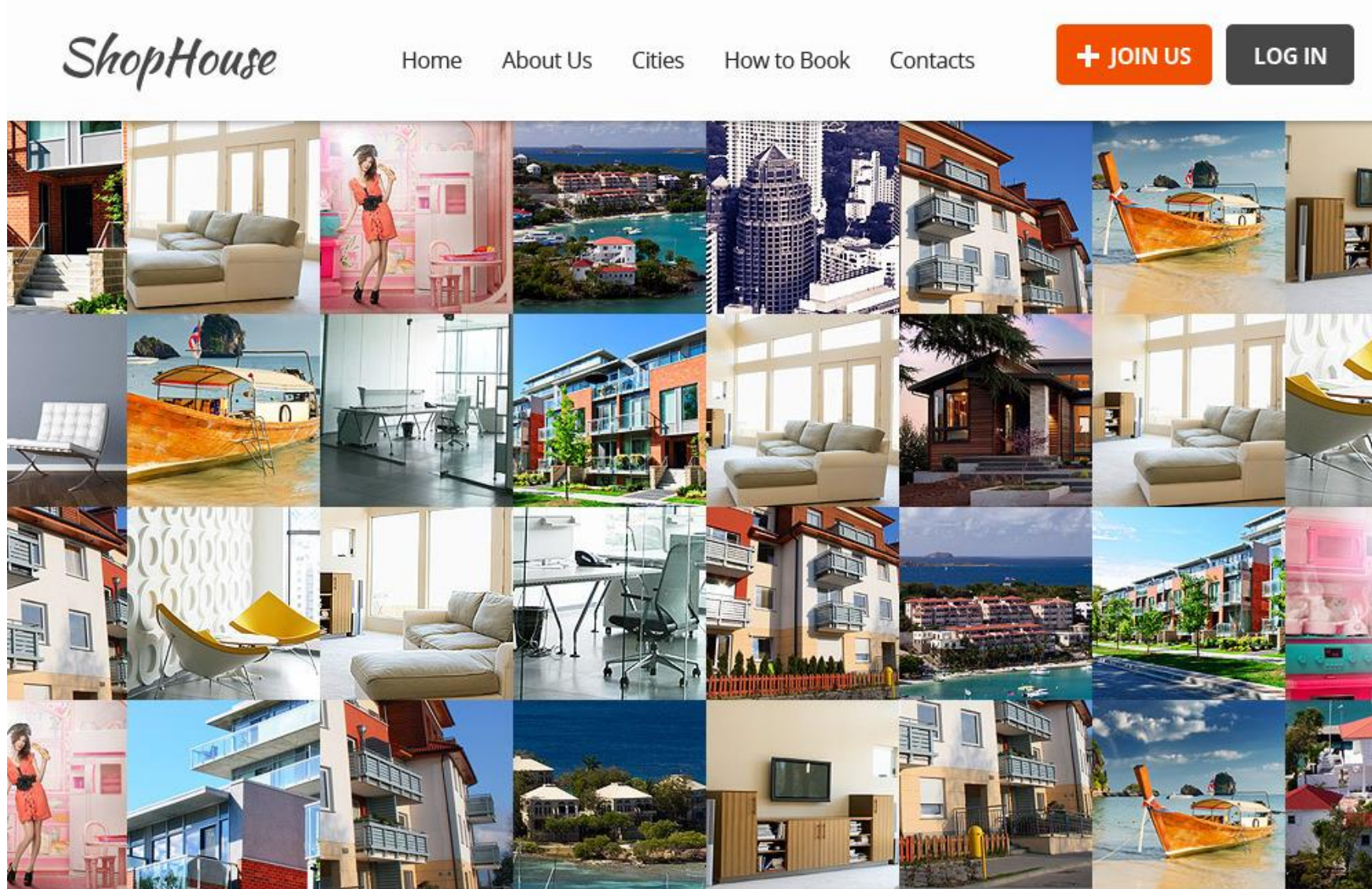
# 本节内容



- ▶ 布局的基本概念、版式
- ▶ 布局思想
- ▶ 如何布局
- ▶ 相对定位、绝对定位

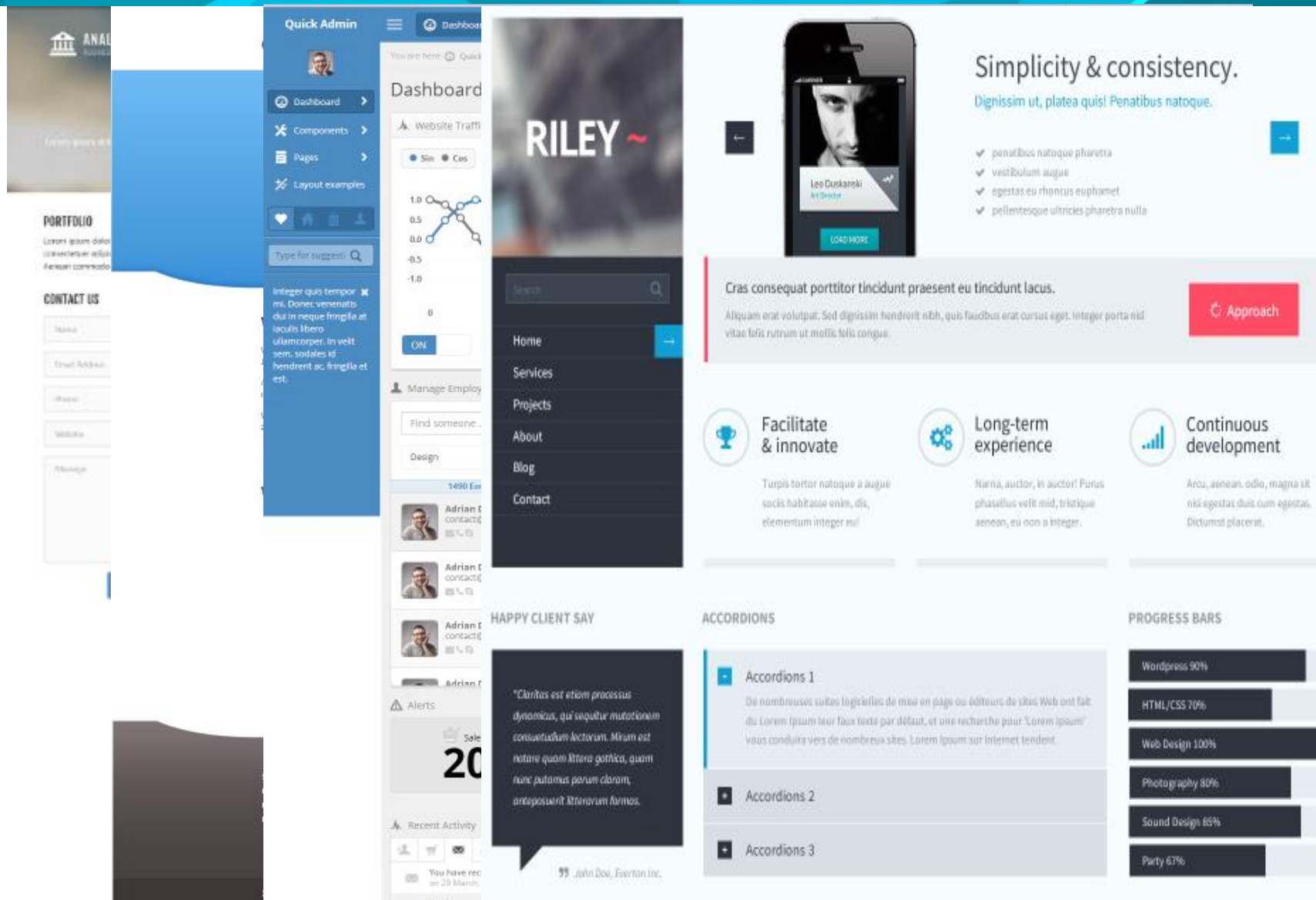


# 认识布局





# ► 布局的概念-常见版式



## ▶ 布局的概念-页面尺寸

### ✓ 页面尺寸：

分辨率:800x600时，页面尺寸:780x428个像素

分辨率:640x480时，页面尺寸:620x311个像素

分辨率:1024x768时，页面尺寸:1005x600个像素

进行网页布局设计时，需要从用户的体验角度出发。

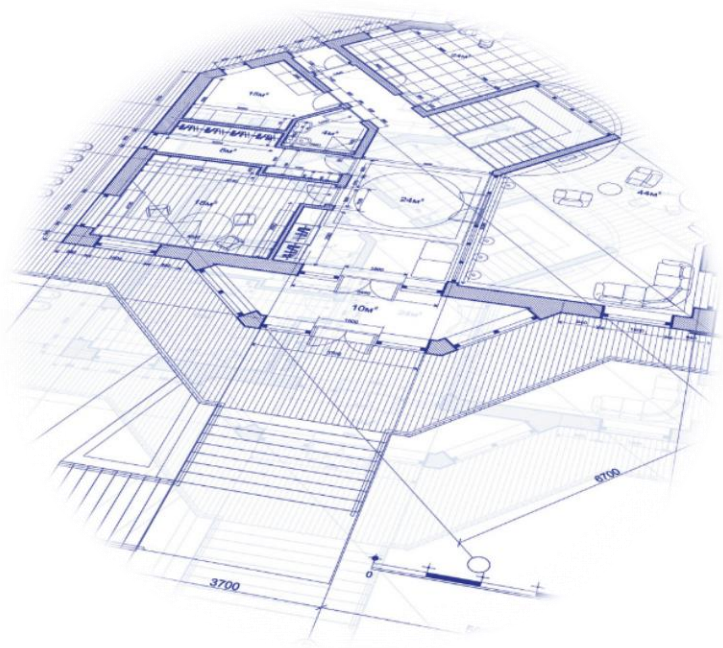
指导性的原则：**页面宽度不超过一屏，高度不超过三屏。**

# 本节内容



- ▶ 布局的基本概念、版式
- ▶ 布局思想
- ▶ 如何布局
- ▶ 相对定位、绝对定位

# 网页布局



网页布局是指网页内容在页面上  
所处位置的设计。

网页布局宗旨：

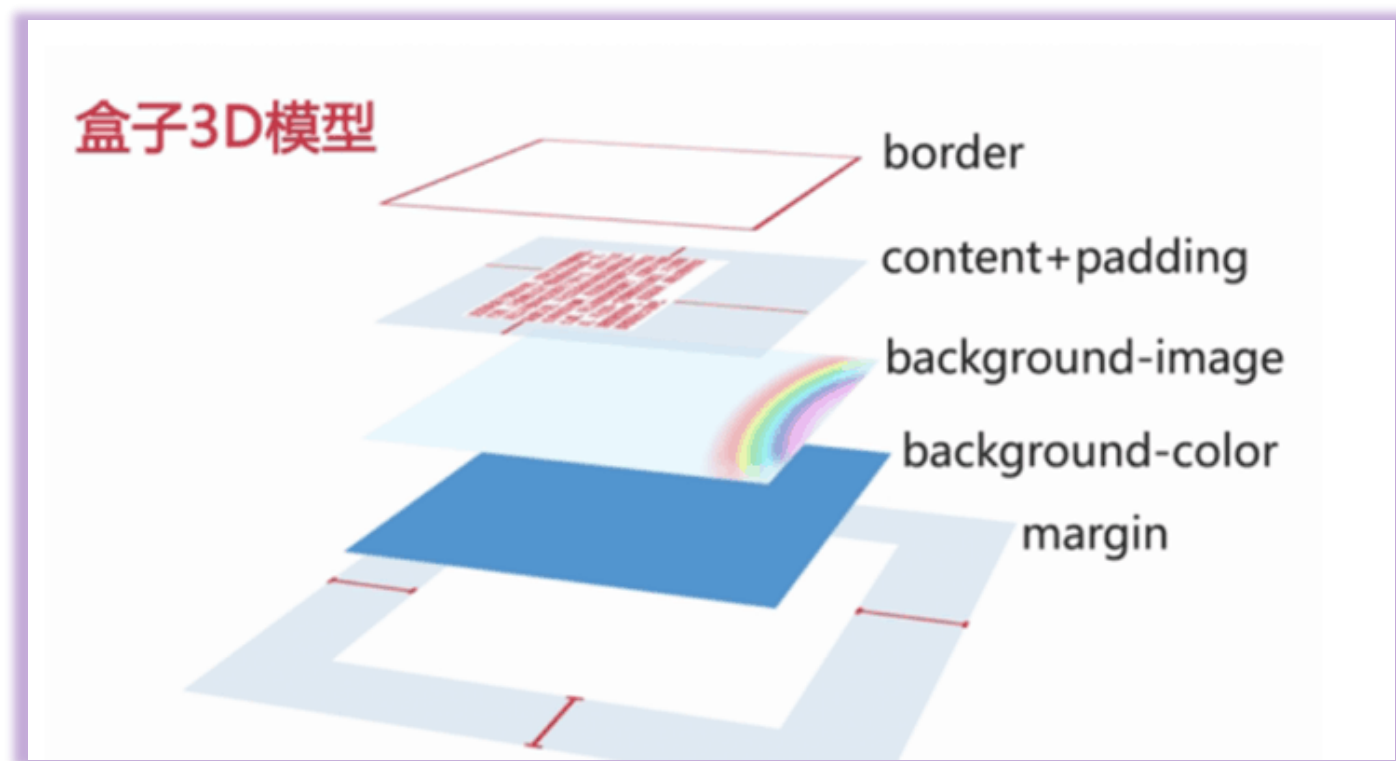
**由整体到局部**



## 布局标签：**div**

- div相当于一个容器（盒子）
- div标签可以把文档分割为独立的、不同的部分
- 通过id或class属性区分不同的容器

**div**具有盒子模型的所有属性，  
布局时用来控制元素之间的距离和相对位置。





### 教务动态

[更多>>](#)**06**

2015.12

**关于2014-2015 学年第开始一学雅通识课程开课的新闻**

专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读, 专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读。

**06**

2015.12

**关于2014-2015 学年第开始一学雅通识课程开课的新闻**

专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读, 专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读。

**06**

2015.12

**关于2014-2015 学年第开始一学雅通识课程开课的新闻**

专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读, 专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读。

**06**

2015.12

**关于2014-2015 学年第开始一学雅通识课程开课的新闻**

专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读, 专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读。

### 通知公告

[更多>>](#)**2015年寒假公共教室使用安排公告**

专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读。

**2015年寒假公共教室使用安排公告**

专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读。

**2015年寒假公共教室使用安排公告**

专家解读、权威评论、优秀事迹、大学责任及我校工作动态专家解读。

**学生服务**

学生网上综合系统  
网上教学平台  
空闲教室查询  
网上教评

**教师服务**

网上教学平台  
教师网上综合系统  
学院网上综合系统  
教评结果查询

**公共服务**

最新下载 管理登录  
成绩查询 办事指南  
大练兵  
办公自动化登录

**质量工程**

精品课程 创新高地  
教学团队 教学名师  
实验教学示范中心

# 本节内容



- ▶ 布局的基本概念、版式
- ▶ 布局思想
- ▶ 如何布局
- ▶ 相对定位、绝对定位

# 布局的方法



```
<html>
<head>
  <title>网页标题</title>
</head>
<body>
  <div class="mybody">
    <div class="header">
      <div class="nav"></div>
      <div class="logo"></div>
      <div class="banner"></div>
    </div>
    <div class="content">
      <div class="news"></div>
      <div class="menu"></div>
      <div class="link"></div>
    </div>
    <div class="footer"></div>
  </div>
</body>
</html>
```

例子：11\_1



# 布局的方法



<div>...</div>是块级元素，如何将两个块级元素并列显示？



## 流 ( Flow )

- **元素流**是浏览器在页面上摆放HTML元素的方法
- 浏览器从HTML文件最上面开始，**从上到下**沿着元素流逐个显示所遇到的各个元素
  - **块元素**，从上向下流，各元素之间有一个换行
  - **行内元素**，从左上方方向右下方流，各元素在水平方向上相互挨着。

## Float浮动定位

默认值，无浮动

向父元素右侧紧靠

float: none | left | right

向父元素左侧紧靠

## ► 布局的方法

- ✓ 浮动：定义元素在哪个方向浮动，改变页面中对象的前后流动顺序

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。
inherit	规定应该从父元素继承 <code>float</code> 属性的值。

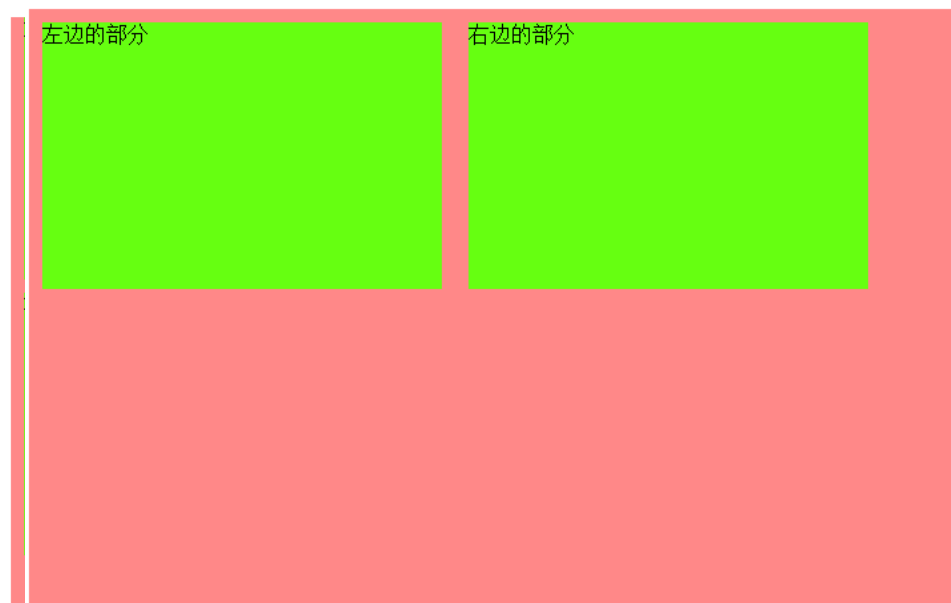
浮动元素不再是正常流的一部分！

float.html

## ► 布局的方法

```
<style type="text/css">
  .news{
    background-color:#f88;
    height:450px;
    width:700px
  }
  .nf{
    float:left;
    margin:10px;
    width:300px;
    height:200px;
    background-color:#6f1;
  }
</style>
```

```
<div class="news">
  <div class="nf">左边的部分</div>
  <div class="nf">右边的部分</div>
</div>
```



例子：11\_2



## Float应用 - 图文混排

图像浮动，文字会环绕在图像周围

南瓜煲鸡的做法其实及其简单！成本低廉，操作也并不复杂，而口感鲜美，也不失为节日菜谱！尤其是换季了，家中的



男女老少来份南瓜煲鸡，营养又易美味；本人特喜欢南瓜煲鸡的做法，感觉这是最好吃的家常菜了，很奇怪为什么会有那么多的人不喜欢吃南瓜煲鸡呢？后来才明白原来他们是因为做南瓜煲鸡的时候方法没有掌握好！南瓜煲鸡有很多的做法，正确的做法才能做出好吃的南瓜煲鸡！南瓜煲鸡的最好做法是能做到不油不腻，这样的南瓜煲鸡吃起来才会香，而且不用担心会长胖！不信做做看！

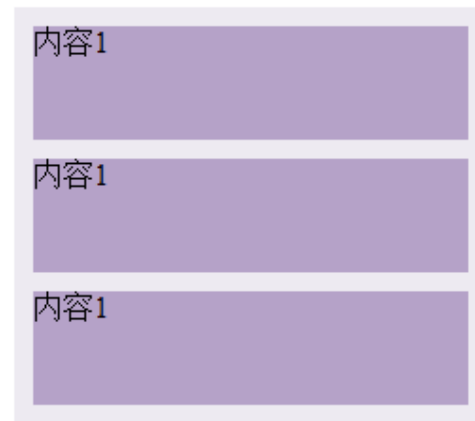
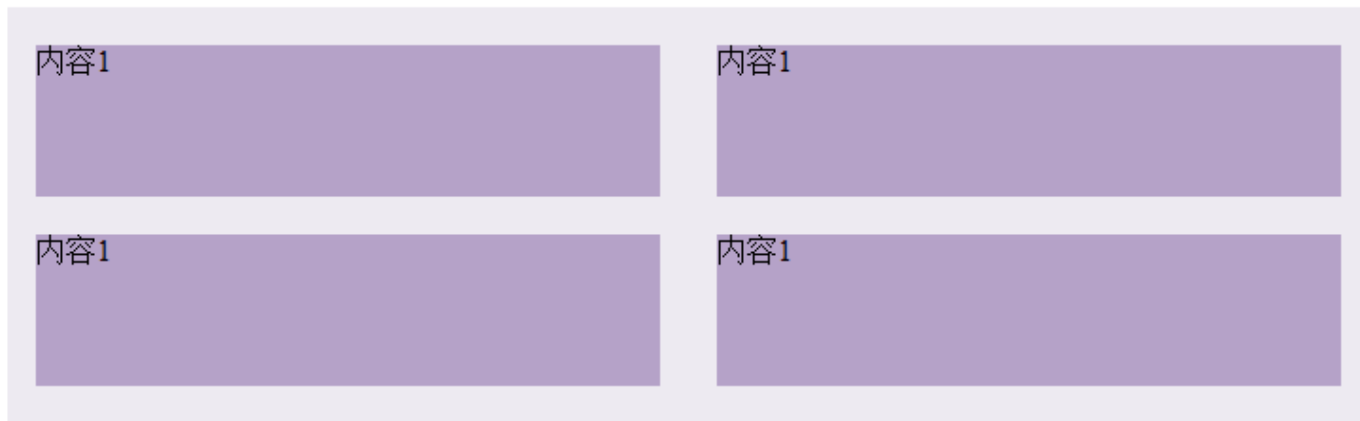
例子：11\_3

## ▶ Float应用

1. 浮动元素距离父元素边框的位置，是该侧的父元素padding+自身margin的值
2. 元素一旦浮动就不属于父元素了，且通常不会超过父元素的边界
3. 浮动元素不会相互重叠
4. 不能上下浮动，通常只设一种浮动即可
5. 如果父元素宽度不够，浮动元素会另起一行显示
6. 一个浮动元素会尽量向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止
7. 浮动元素之前的元素将不会受到影响，浮动元素之后的元素将围绕它
8. 如果图像是右浮动，下面的文本流将环绕在它左边

# ► 布局的方法

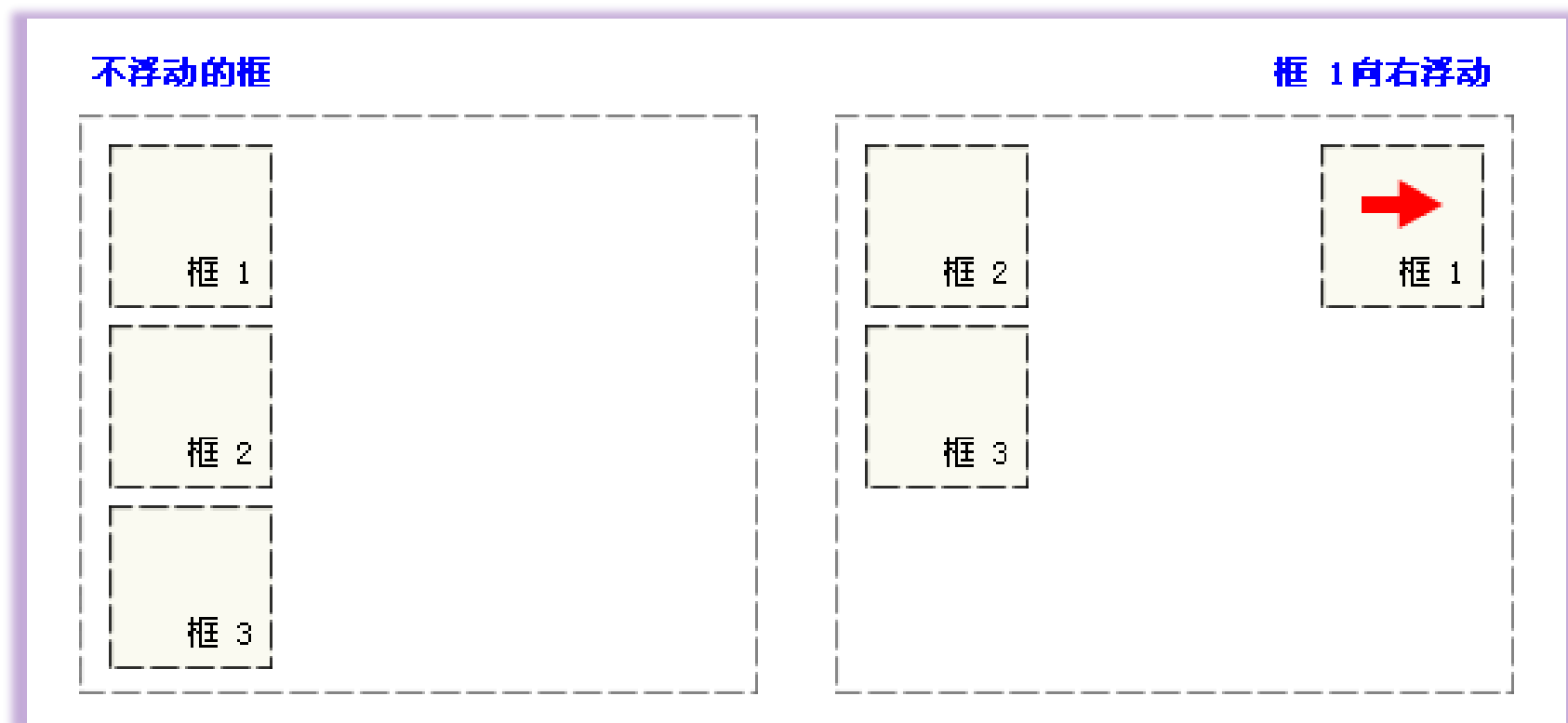
练习：



例子：11\_4



# 浮动

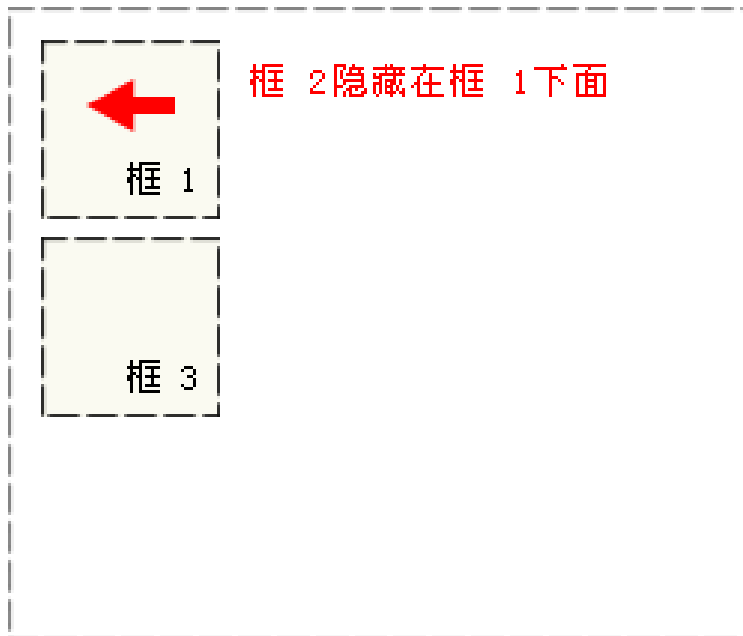


当框 1 向右浮动时，脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘。

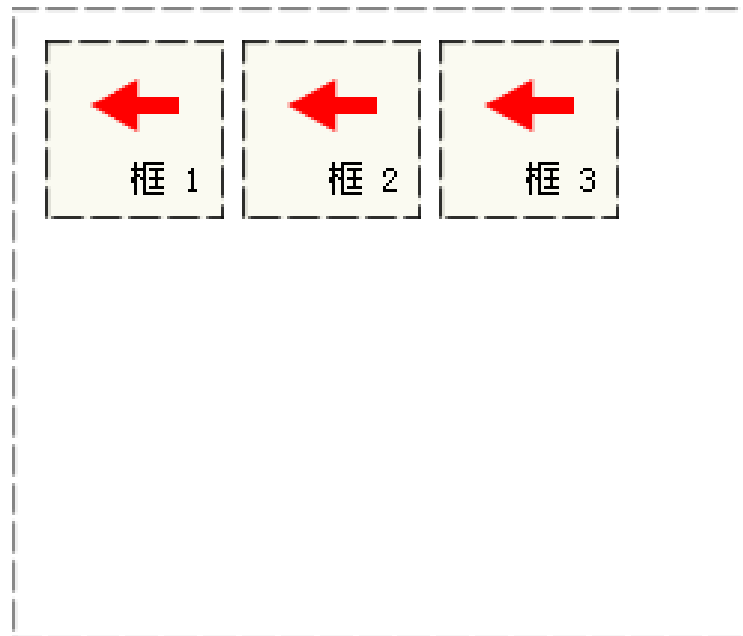


# 浮动

框 1 向左浮动



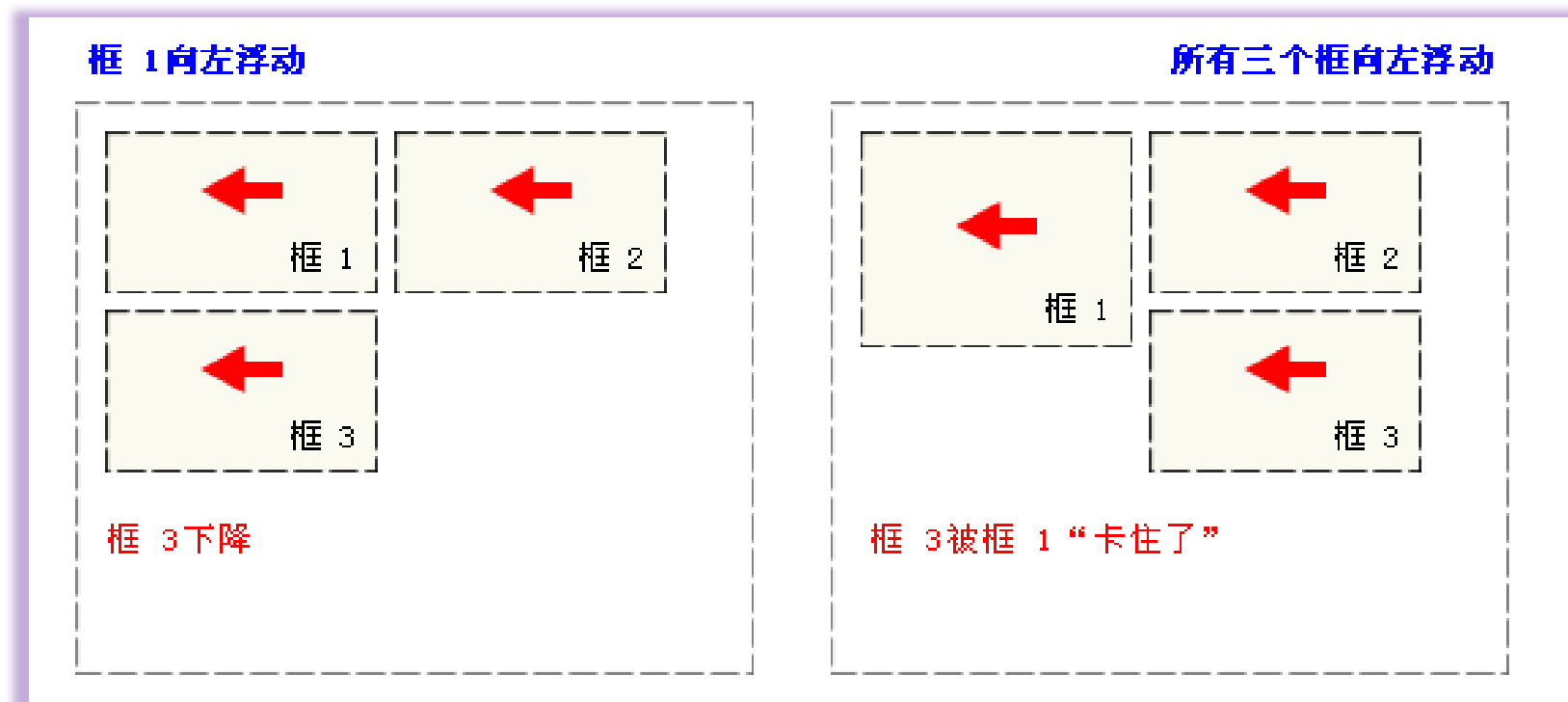
所有三个框向左浮动







# 浮动



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”。



# 浮动



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”。

## 溢出处理- overflow

例子：11\_5

定义溢出元素内容区的内容会如何处理

值	描述
visible	默认值。内容不会被修剪，会呈现在元素框之外。
hidden	内容会被修剪，并且其余内容是不可见的。
scroll	内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
auto	如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。
inherit	规定应该从父元素继承 overflow 属性的值。

### overflow

- 1、父元素~~不~~设高度并且子元素浮动时，使父元素高度自动适应子元素高度。
- 2、父元素~~设置~~的高度或宽度~~小~~于子元素时，使父元素出现相应滚动条或隐藏子元素超出部分内容。

## ▶ 布局的方法

### 清除浮动 - clear

用来设置该元素边上没有其他元素可以浮动

**clear:** left / right / both / none

**left** : 不允许左边有浮动的元素

**right** : 不允许右边有浮动的元素

**both** : 不允许左右两边有浮动的元素

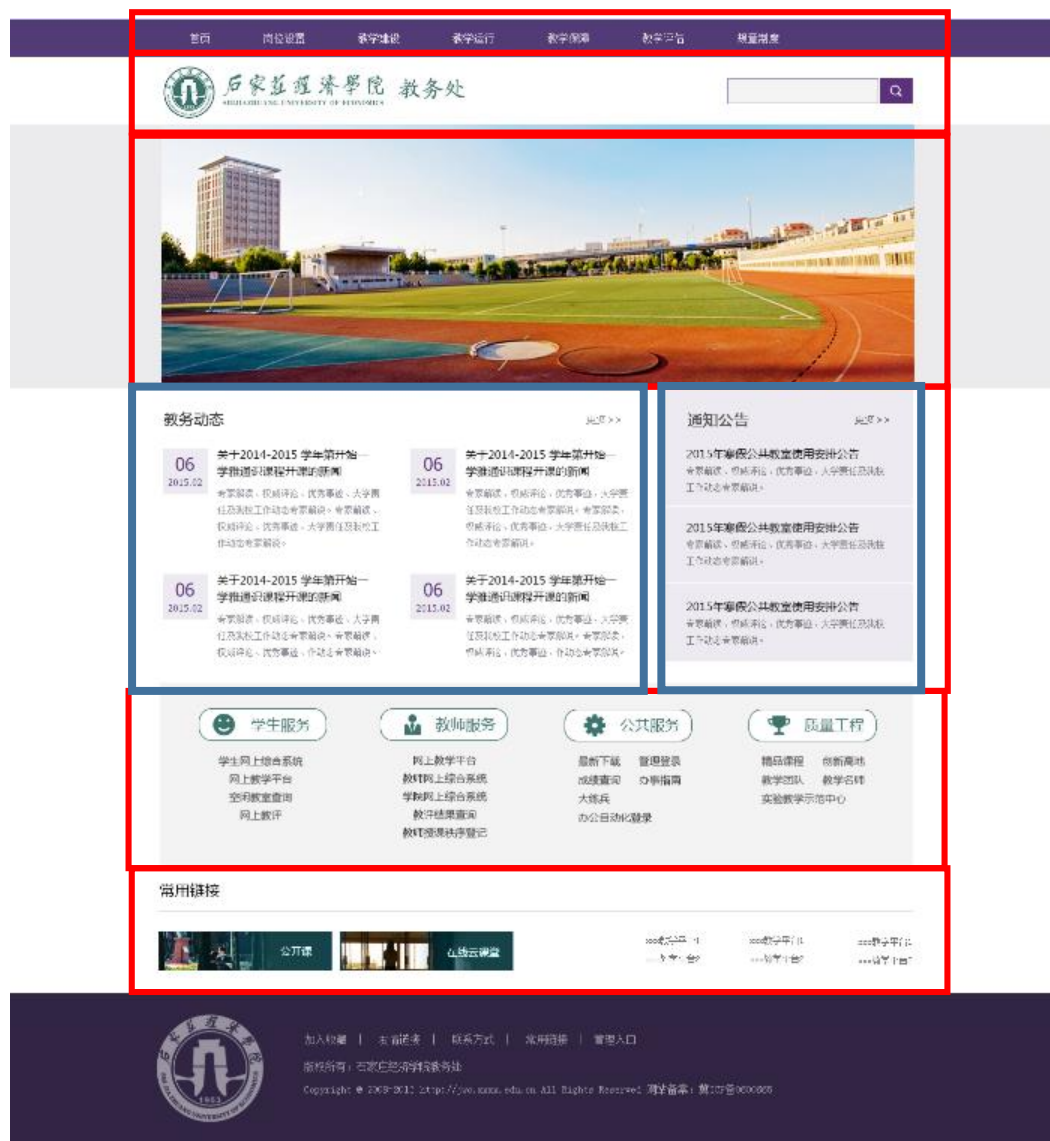
**none** : 默认值，可以浮动

例子：11\_6

例子：clear.html



# 布局的方法



```
<body>
  <div class="mybody">
    <div class="header">
      <div class="nav"></div>
      <div class="logo"></div>
      <div class="banner"></div>
    </div>
    <div class="content">
      <div class="news">
        <div class="left"></div>
        <div class="right"></div>
      </div>
      <div class="menu"></div>
      <div class="link"></div>
    </div>
    <div class="footer"></div>
  </div>
</body>
```

# 本节内容



- ▶ 布局的基本概念、版式
- ▶ 布局思想
- ▶ 如何布局
- ▶ 相对定位、绝对定位

## ▶ 相对定位、绝对定位

### position定位

值	描述
<code>absolute</code>	生成绝对定位的元素，相对于 <code>static</code> 定位以外的第一个父元素进行定位。 元素的位置通过 <code>"left"</code> , <code>"top"</code> , <code>"right"</code> 以及 <code>"bottom"</code> 属性进行规定。
<code>fixed</code>	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 <code>"left"</code> , <code>"top"</code> , <code>"right"</code> 以及 <code>"bottom"</code> 属性进行规定。
<code>relative</code>	生成相对定位的元素，相对于其正常位置进行定位。 因此， <code>"left:20"</code> 会向元素的 <code>LEFT</code> 位置添加 20 像素。
<code>static</code>	默认值。没有定位，元素出现在正常的流中（忽略 <code>top</code> , <code>bottom</code> , <code>left</code> , <code>right</code> 或者 <code>z-index</code> 声明）。
<code>inherit</code>	规定应该从父元素继承 <code>position</code> 属性的值。

## ▶ 相对定位、绝对定位

### ✓ 绝对定位

### ✓ `position:absolute`

- 1、与left , right , top , bottom等属性共同使用
- 2、如果父元素未设置position属性，则以浏览器窗口左上角为起始位置
- 3、如果父元素设置了position属性，则以父元素左上角为起始位置

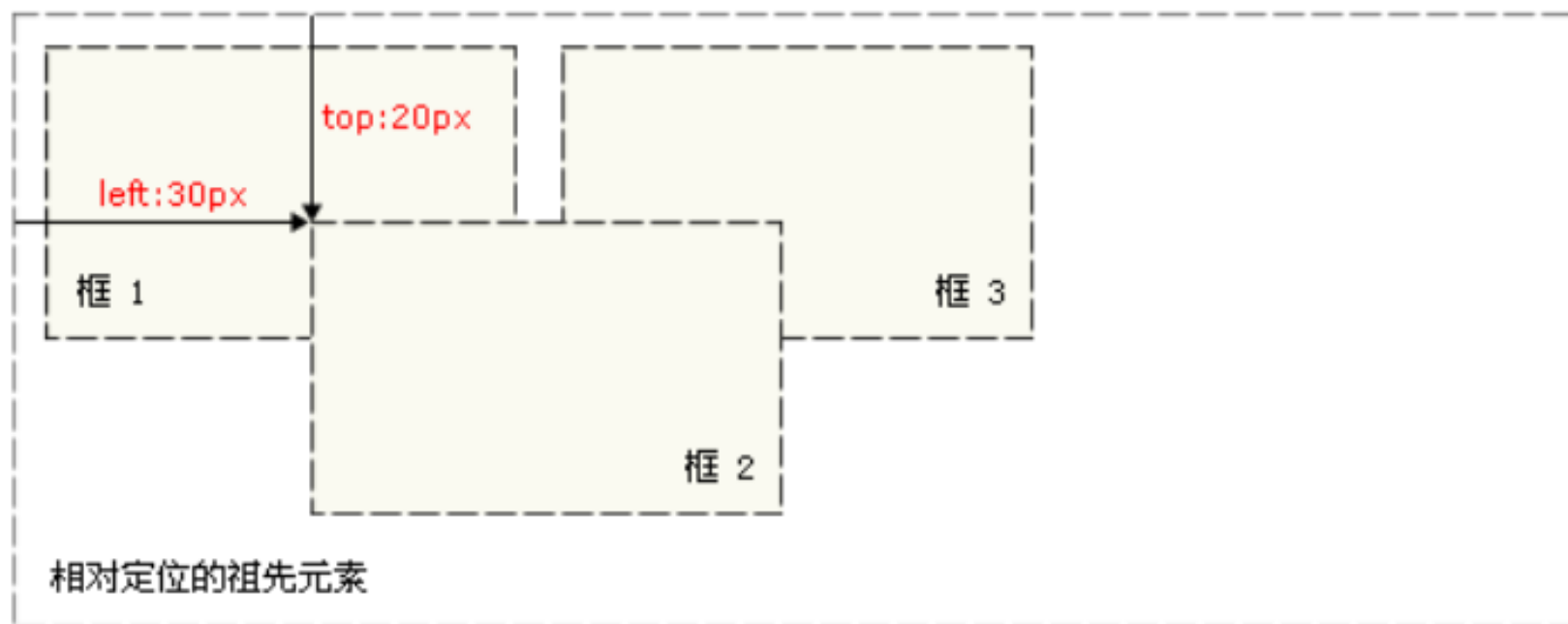
例子：11\_7

## ▶ 相对定位、绝对定位

✓ **绝对定位**

例子：11\_7

✓ **position:absolute**



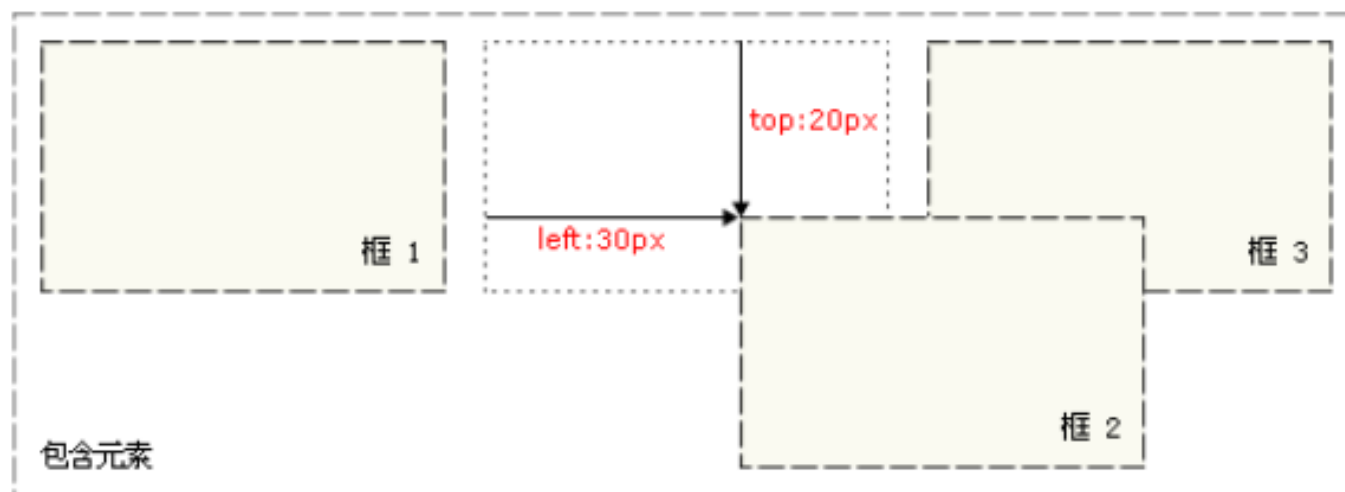
## ▶ 相对定位、绝对定位

### ✓ 相对定位

例子：11\_8

### ✓ `position:relative`

- 1、与left, right, top, bottom等属性共同使用
- 2、以自身本应在的位置为起始位置





## ▶ 相对定位、绝对定位

float和position相似点：

- 1、都是将元素浮动起来
- 2、元素一旦设置就与父元素没关系了

float和position不同点：

- 1、position与top、left等配套使用
- 2、float位置移动通过margin、padding等实现
- 3、overflow和clear对position无效
- 4、布局通常使用float，而不是position

### 顺序堆叠 **z-index**

- 1、用于体现元素在Z轴空间的堆叠关系
- 2、与 **position : absolute** 配套使用
- 3、大部分情况以数字为取值
- 4、可为多个元素设置
- 5、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上边。

例子：11\_9

## ▶ 本节小结

- 网页布局方法：应用盒子模型特征与浮动构建页面布局
- 网页布局思想：从大到小，整体到局部
- float：用于网页布局
- overflow：元素溢出处理
- clear：清除元素浮动
- position：相对定位和绝对定位
- z-index：元素顺序堆叠



谢谢  
Thank You