



# Web前端开发

---

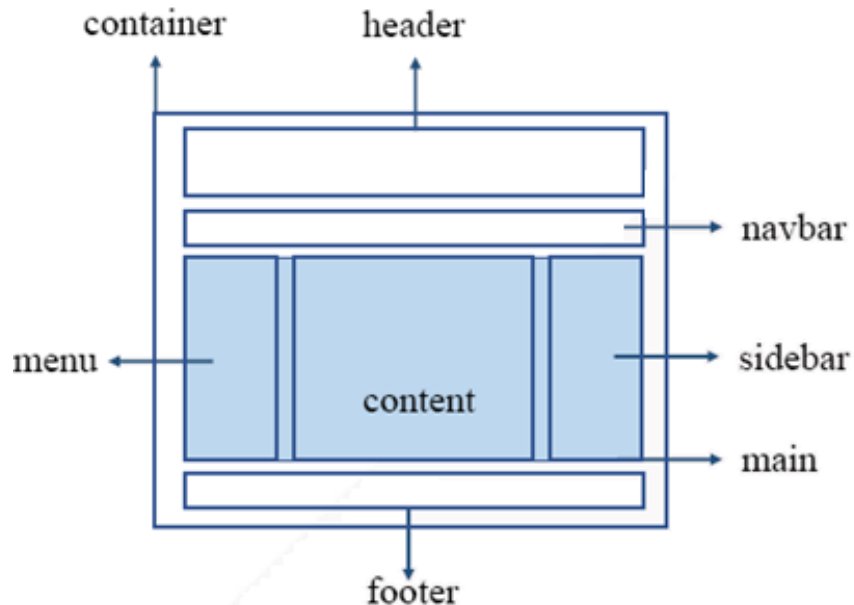
(*HTML5 + CSS3 + JAVASCRIPT*)  
■ 单元3 布局与定位

## 单元3 布局与定位

### 目 录

- 3.1 概述
- 3.2 盒子模型
- 3.3 CSS定位机制

# 3.1 概述



# 内容

## 盒子模型

页面元素的大小

边框

与其他元素的距离

元素什么样

## 定位机制

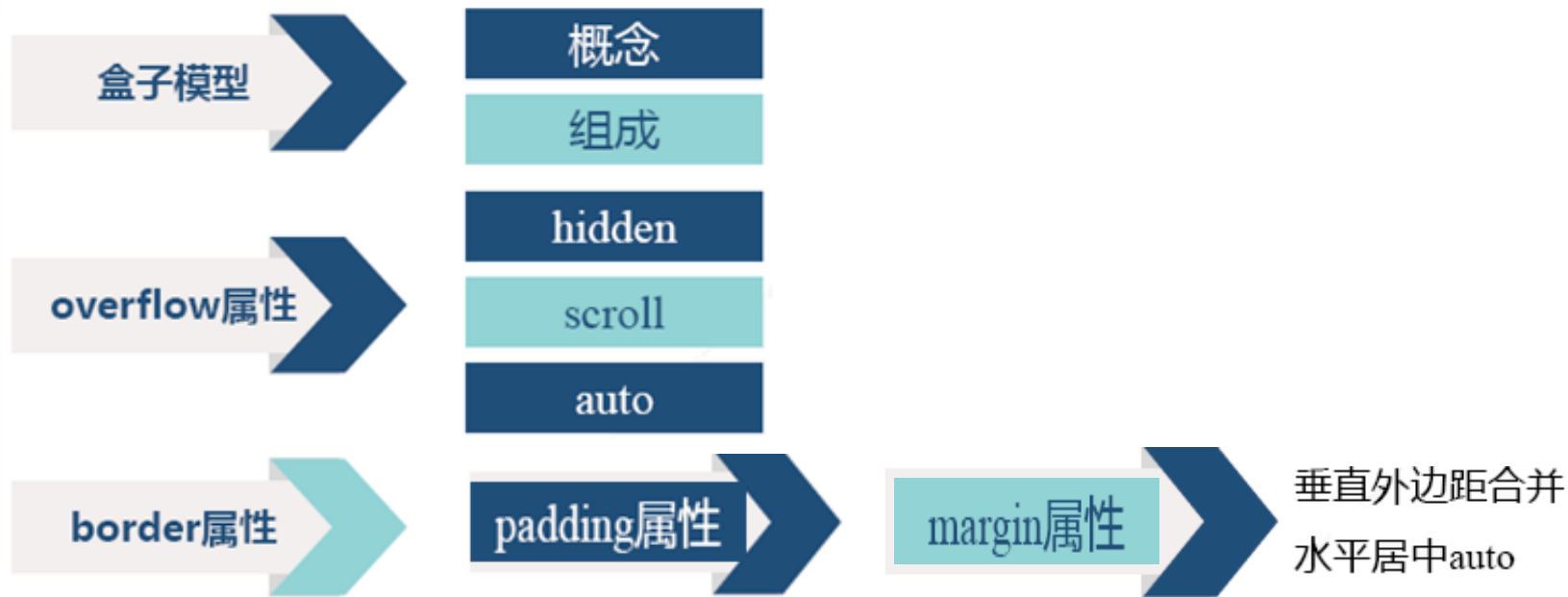
文档流

浮动定位

层定位

元素放在哪

## 3.2 盒子模型

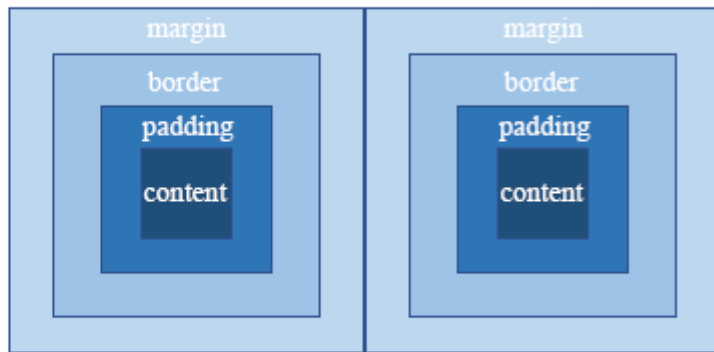
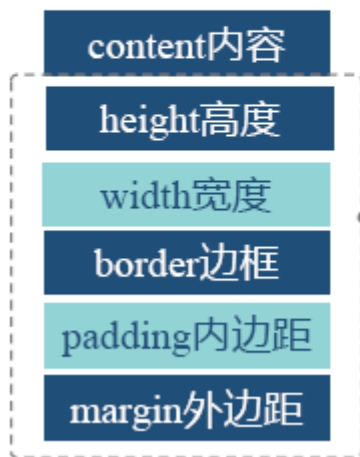


网页中的 区域、图片、导航、列表、段落，都可以是盒子

# 盒子模型

页面中的**所有元素**都可以看成一个盒子，占据着一定的页面空间

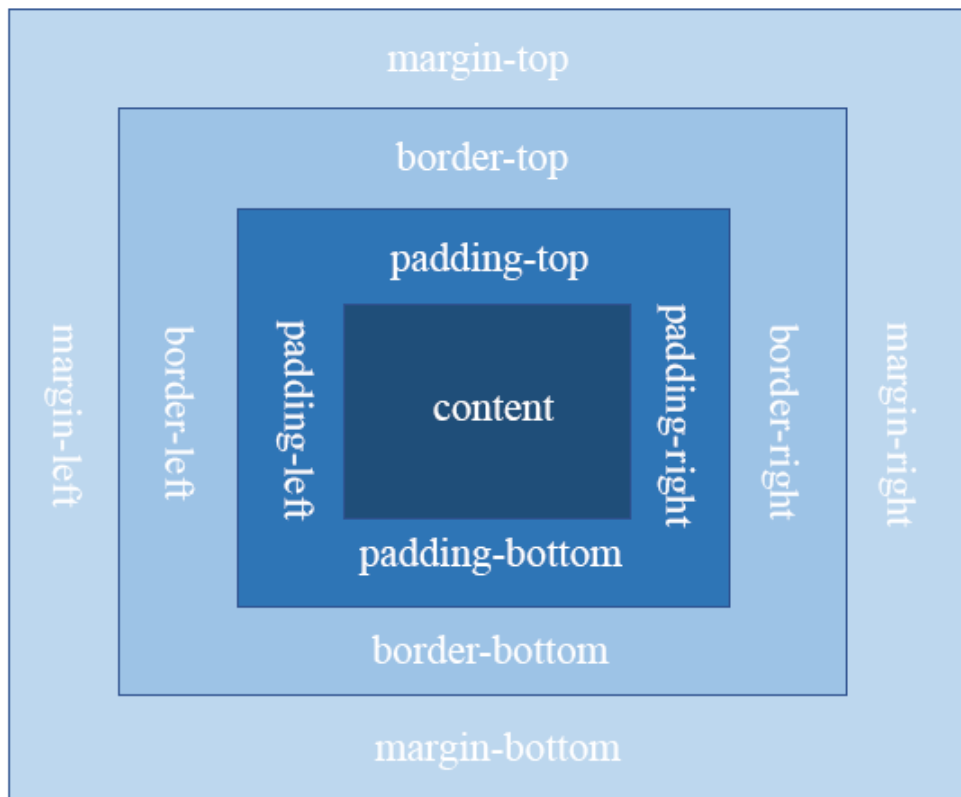
## 盒子模型组成



都是CSS样式的属性

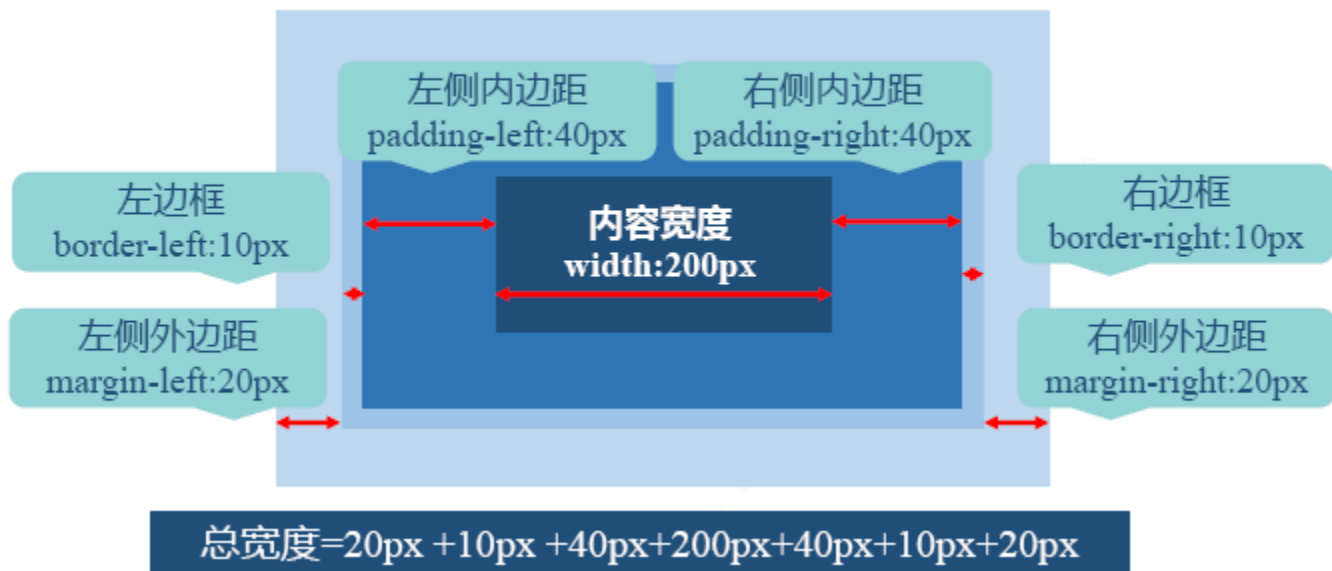


# 盒子模型



# 盒子模型

一个盒子的实际宽度、高度: **content+padding+border+margin**

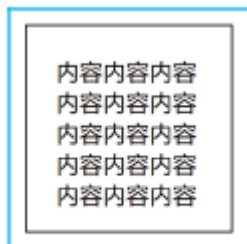




# 盒子模型



```
<html>
<head>
<style type="text/css">
#box{
  width:100px;
  height:100px;
  border:1px solid;
  padding:20px;
  margin:10px;
}
</style>
</head>
```



盒子以外的文字



```
<body>
  <div id="box">
    内容内容内容...
  </div>
</body>
</html>
```

# overflow属性

当内容溢出盒子框时，overflow属性取值

hidden

超出部分不可见

scroll

显示滚动条

auto

如果有超出部分，显示滚动条



内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容

内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容

scroll

内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容  
内容内容内容内容内容内容内容

hidden

# border属性

<b>border-width:</b>	px、thin、medium、thick
<b>border-style:</b>	dashed、dotted、solid、double
<b>border-color:</b>	颜色
<b>border:</b>	width style color

```
div{  
border-width:2px;  
border-style:solid;  
border-color:red;  
}
```

```
div{  
border:2px solid red;  
}
```

```
div{  
border-bottom:1px solid red;  
}
```



# border属性的一个应用



```
.line {  
    height: 1px;  
    width: 500px;  
    border-top: 1px solid #e5e5e5;  
}
```

水平分割线

## 习近平重访“海洋”全俄儿童中心：这牵挂，就像海洋一样深

08月11日 00:00 人民日报

1989

原标题：这牵挂，就像海洋一样深

“习主席来了！”

位于符拉迪沃斯托克市的“海洋”全俄儿童中心，顿时成了欢乐的海洋。

当地时间9月12日晚6时20分许，习近平主席在出席第四届东方经济论坛全会及相关活动后，在俄罗斯总统普京陪同下，来到阔别6年之久的“海洋”儿童中心访问。

# padding属性和 margin属性

取值: **px**, **%**(外层盒子的宽度和高度)

内边距

外边距

组成

padding: 5px;

margin: 5px;

上右下左

padding-top: 10%;

margin-top: 10%;

上

padding-left

margin-left

左

padding-right

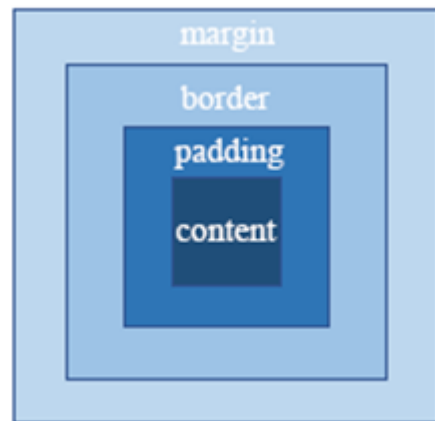
margin-right

右

padding-bottom

margin-bottom

下



对浏览器默认的设置清零

```
* {  
  margin: 0;  
  padding: 0;  
}
```

# margin属性

```
margin: 1px;
```

```
margin: 1px 1px 1px 1px;
```

```
margin: 1px 2px;
```

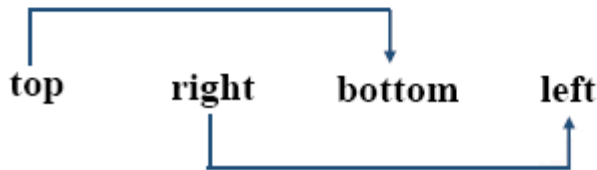
```
margin: 1px 2px 1px 2px;
```

```
margin: 1px 2px 3px;
```

```
margin: 1px 2px 3px 2px;
```

```
margin: 1px 2px 1px 3px;
```

注意，这里虽然上下边距都为1px  
但是这里不能缩写。



# margin属性

margin的合并：垂直方向合并，水平方向不合并

外边距合并形成  
一个外边距

合并之前



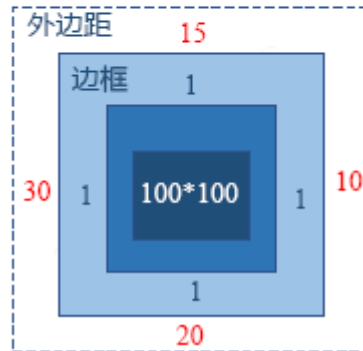
合并之后



```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
div{
width:100px;
height:100px;
margin:15px 10px 20px 30px;
border:1px solid red;
}
</style>
</head>
<body>
<div id="box1">box1</div>
<div id="box2">box2</div>
</body>
</html>
```

box1

box2



# margin属性的一个应用

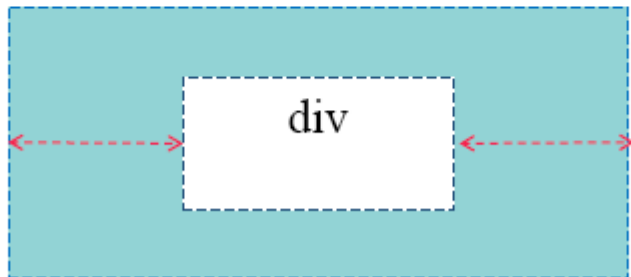
## 水平居中

图片、文字水平居中

`text-align:center;`

div水平居中

`margin:0 auto;` 浏览器自动计算





# 案例-图片列表

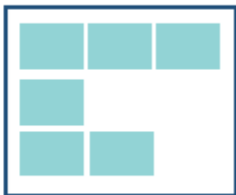
```
<div id="newsimglist">
  <img src = "images/crisis.jpg" />
  <img src = "images/crisis.jpg" />
  <img src = "images/crisis.jpg" />
</div>
#newsimglist{
  text-align:center;
  font-size:0; /*否则图间有空隙*/
}
#newsimglist img{
  height:100px;
  width:100px;
  margin-left:5px;
  border:1px solid #0cf;
  padding:5px;
}
```



## 3.3 定位机制

CSS中，存在3种的定位机制： 文档流 浮动定位 层定位

文档流 flow



浮动 float



层 layer



### 3.3.1 文档流定位

元素分类

block

inline

inline-block

元素类型转换

display属性

display: none

元素不会被显示

display: block

显示为block元素

display: inline

显示为inline元素

display: inline-block

显示为inline-block元素

# 元素分类-block

## block元素特点



- 独占一行
- 元素的height、width、margin、padding都可设置

## 常见的block元素

`<div>`、`<p>`、`<h1>...<h6>`、  
`<ol>`、`<ul>`、`<table>`、`<form>`

## 将元素显示为block元素

```
a{  
    display:block;  
}
```

inline元素a转换为block元素  
从而使a元素具有块状元素特点。

# 元素分类-inline

## inline元素特点

- 不单独占用一行
- width、height不可设置
- width就是它包含的文字或图片的宽度，不可改变

## 常见的inline元素

`<span>`、`<a>`

## 显示为inline元素

**display:inline;**

inline元素之间有一个间距问题

```
<style type="text/css">
a{
    background:pink;
}
</style>
<body>
  <a href="http://www.baidu.com">百度</a>
  <a href="http://www.mooc.com">慕课网</a>
</body>
```

百度 慕课网

# 元素分类-(inline-block)

## inline-block元素

就是同时具备inline元素、block元素的特点

- 不单独占用一行
- 元素的height、width、margin、padding都可设置

## 常见的inline-block元素

`<img>`

## 显示为inline-block元素

`display:inline-block;`

# 案例-水平导航栏



默认文档流定位下, HTML元素是有不同类型的, 不同类型相互之间也可以转换

# 文档流定位-小结

从上到下，从左到右，输出文档内容

block从上到下独占一行



`<div>`、`<p>`、`<h1>`、`<ul>`、`<ol>`、`<li>`、`<table>`、`<form>`

元素类型转换



display属性

inline从左到右占据一行



`<a>`、`<span>`

inline-block



`<img>`



## 3.3.2 浮动定位

float属性

left、right

clear属性

left、right、both

脱离默认文档流定位

### 3.3.2 浮动定位-float属性

div实现横向多列布局

float属性:

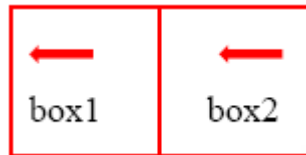
left—左浮动

right—右浮动

none—不浮动

```
<div id="box1"></div>  
<div id="box2"></div>
```

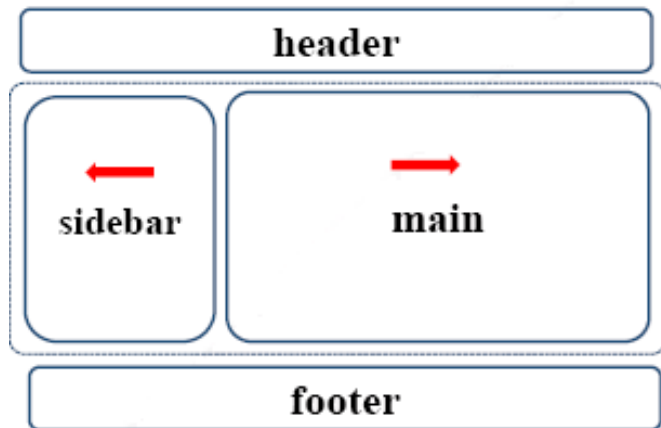
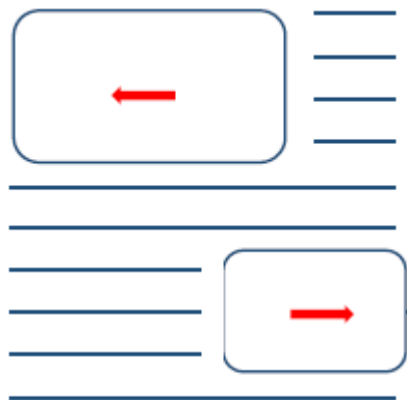
```
div{  
  width:200px;  
  height:200px;  
  border:1px solid red;  
  float:left;  
}
```



盒子box1与box2脱离了文档流原来位置

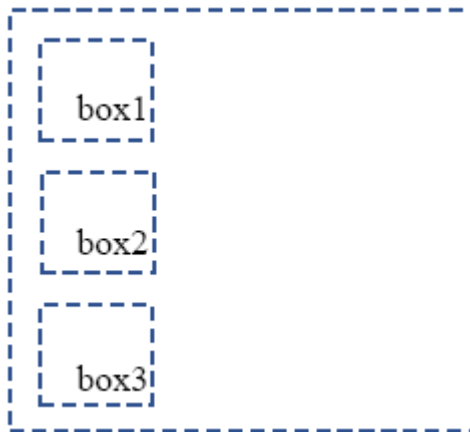
## 3.3.2 浮动定位-float属性

float用处

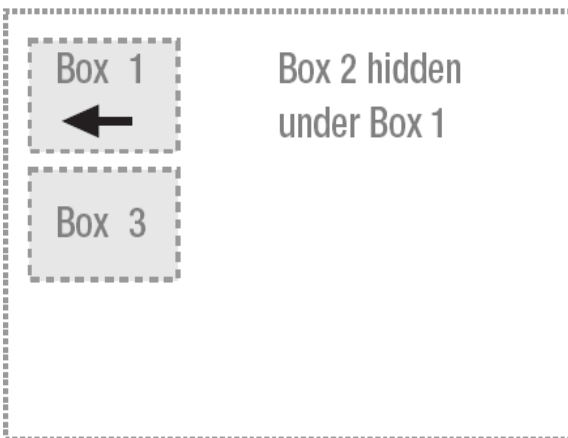


## 3.3.2 浮动定位-float属性

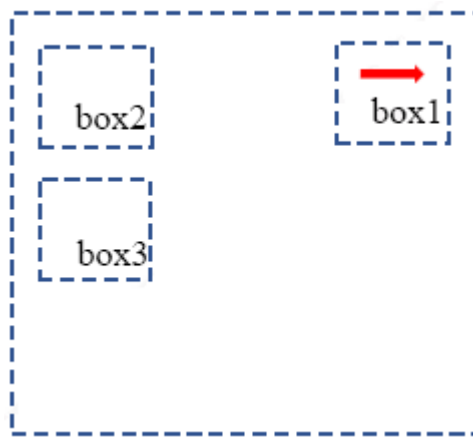
不浮动



框1向左浮动



框1向右浮动



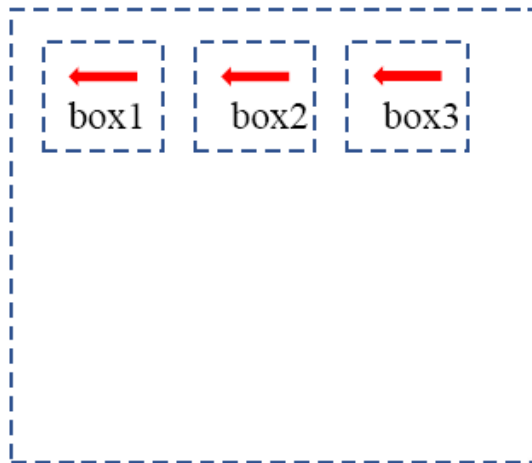
当box1左浮动，由于漂浮在上方，  
所以覆盖了box2

## 总结：一个盒子浮动后的特点

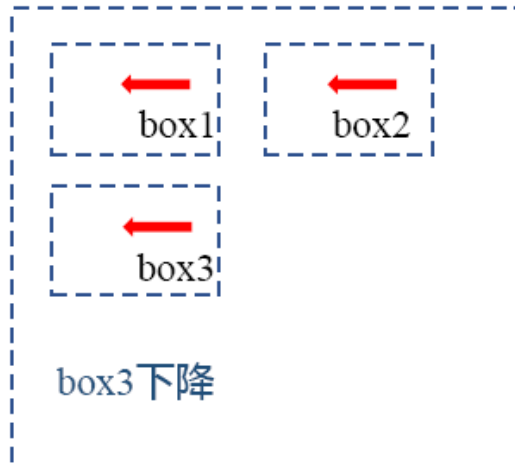
- (1) 浮动后的盒子将以行内块元素显示，但宽度不会自动伸展。
- (2) 浮动的盒子将脱离标准流，即不再占据浏览器原来分配给它的位置。
- (3) 未浮动的盒子将占据浮动盒子的位置，同时未浮动盒子内的内容会环绕浮动后的盒子。

### 3.3.2 浮动定位-float属性

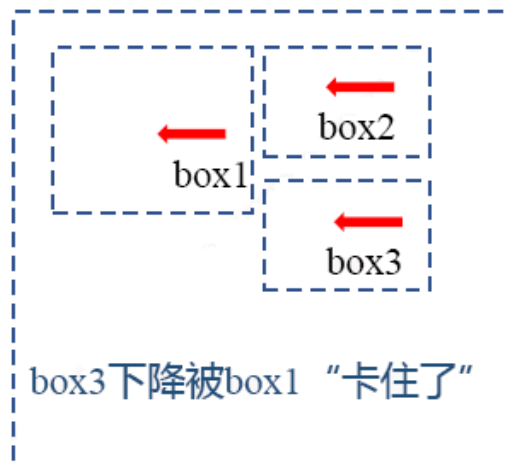
所有三个盒子向左浮动



所有三个盒子向左浮动  
宽度不够



所有三个盒子向左浮动  
宽度不够



**注意：** 使用浮动的元素和其周边的元素之间会产生环绕的效果。

在网页布局中，浮动元素会影响到其他元素的正常排列。因此设置元素浮动后应尽快清除该元素的浮动对其他元素的影响。

## 3.3.2 浮动定位-clear属性

clear

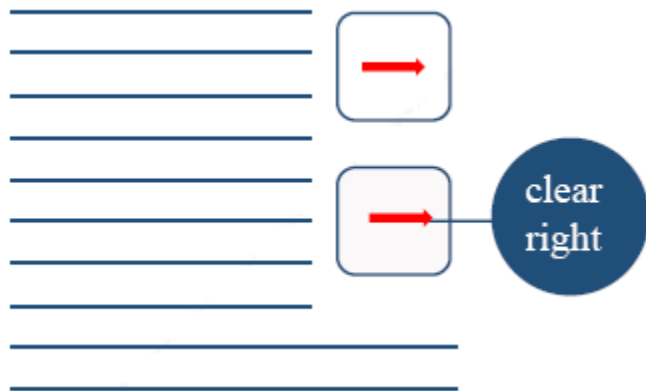
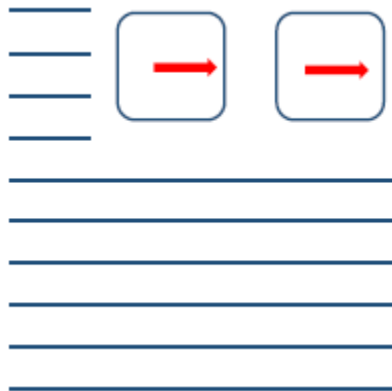
表示左边、右边或两边不允许有浮动的元素。

both, 清除左右两边的浮动。

left 和 right 只能清除一个方向的浮动。

none 是默认值, 只在需要移除已指定的清除值时用到。

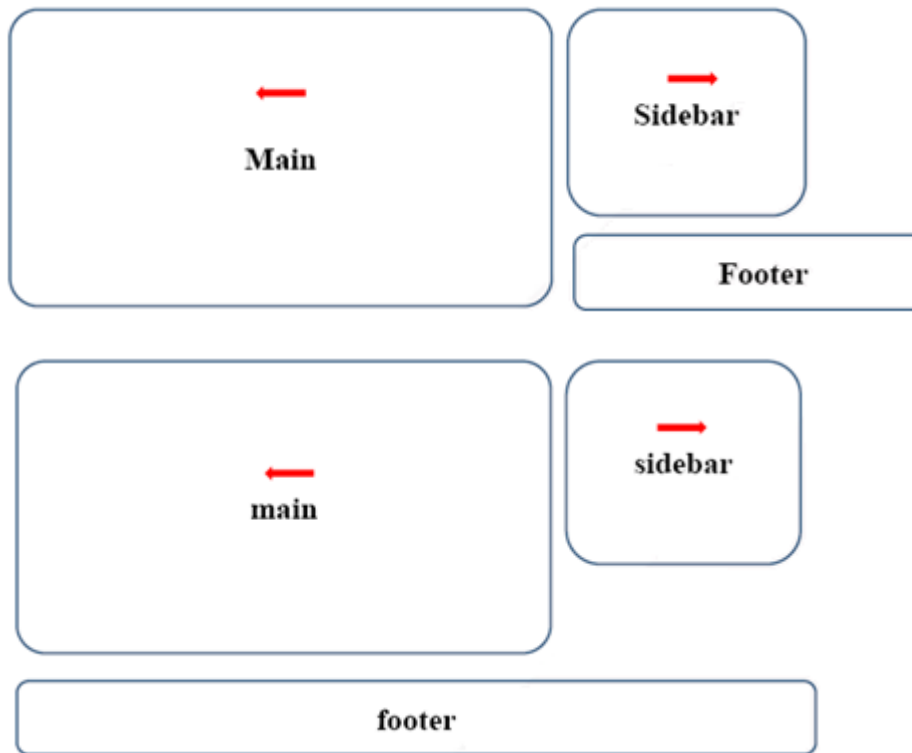
### 单方向清除浮动



## 3.3.2 浮动定位-clear属性

侧栏向右浮动，并且短于主内容区域。页脚(footer)于是按浮动所要求的向上跳到了可能的空间。

```
#footer{  
    clear:both;  
}
```





# 案例



### 3.3.3 层定位

像图像软件中的图层一样可以对每个layer能够精确定位操作

position属性

(相对于谁定位)

fixed 固定定位

relative 相对定位

absolute 绝对定位

通过以下属性定位

left属性

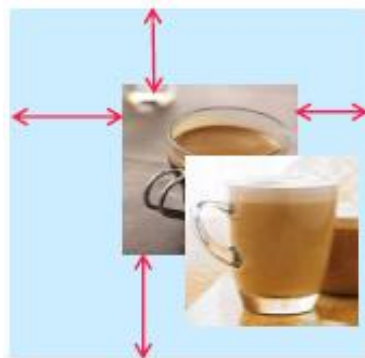
right属性

top属性

bottom属性

z-index属性

(位置在哪里)



z-index: 值大在上面

**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。

```
.zindex-1 {  
    position: fixed;  
    left: 20px;  
    top: 20px;  
    border: 1px dashed #00f;  
    z-index: 1;  
}
```

说明：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。默认为0。

z-index 仅能在定位元素上奏效（例如 position: absolute/relative/fixed;）

# P167: ch3-page-010.html

```
.zindex-1 {  
    position: fixed;  
    left: 20px;  
    top: 20px;  
    border: #00f 1px dashed;  
    z-index: 20;  
}
```

```
.zindex-2 {  
    position: fixed;  
    left: 20px;  
    top: 20px;  
    border: #f00 1px solid;  
    z-index: 10;  
}
```

### 3.3.3 层定位-position属性

static: 默认值

没有定位, 元素出现在正常的流中  
top, bottom, left, right , z-index 无效

fixed: 固定定位

相对于浏览器窗口进行定位  
top, bottom, left, right , z-index 有效

relative: 相对定位

相对于其在普通文档流中的位置进行定位。  
top, bottom, left, right , z-index 有效

absolute: 绝对定位

相对于 static 定位以外的第一个父元素进行定位  
top, bottom, left, right , z-index 有效



### 3.3.3 层定位-固定定位

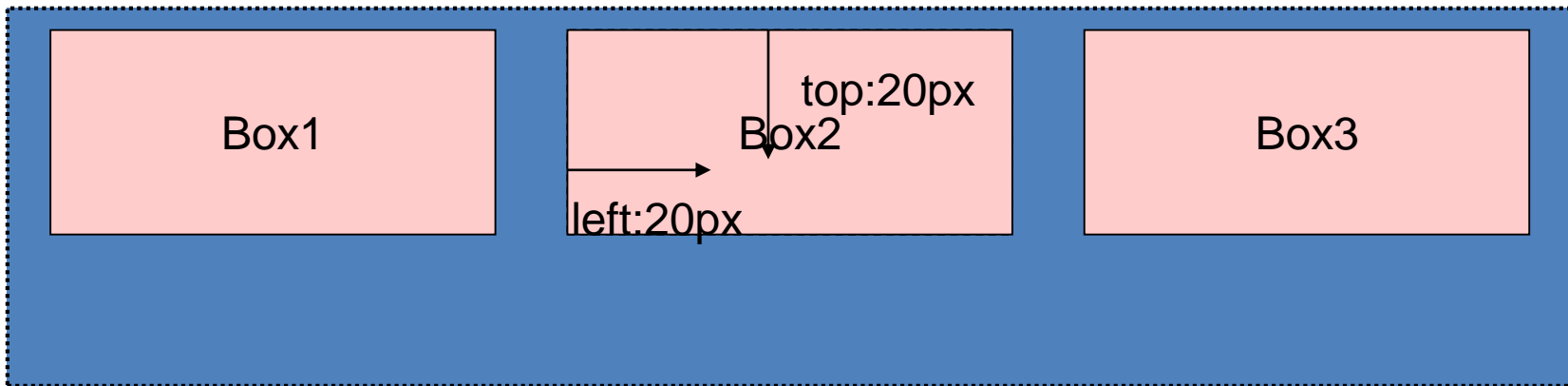


### 3.3.3 层定位-相对定位

relative

定义为relative的元素未脱离正常的文档流，  
其在文档流中的**原位置依然存在**

```
#Box2{  
    position: relative;  
    left: 20px;  
    top: 20px;  
}
```



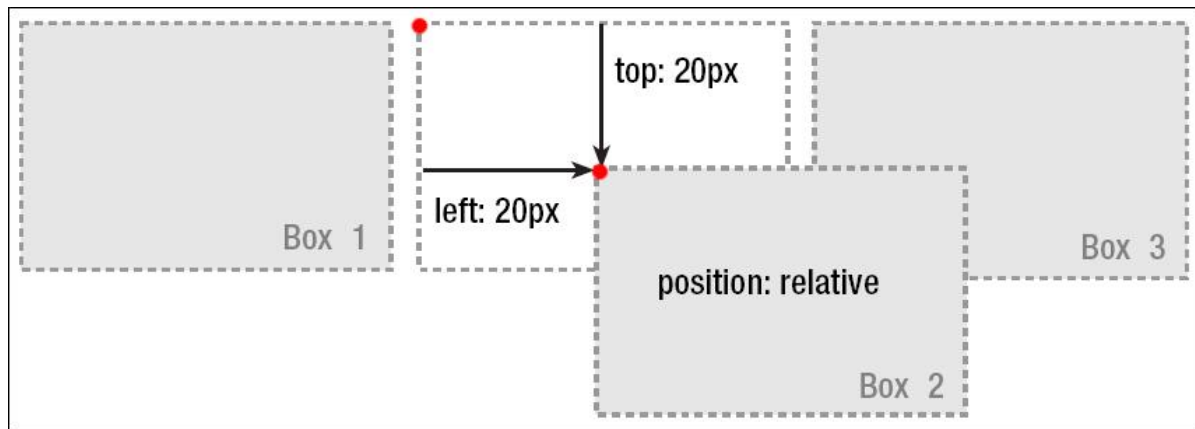
Box2设置了相对定位。但仍保留着原来的位置空间。



### 3.3.3 层定位-相对定位

relative

定义为relative的元素未脱离正常的文档流，  
其在文档流中的**原位置依然存在**

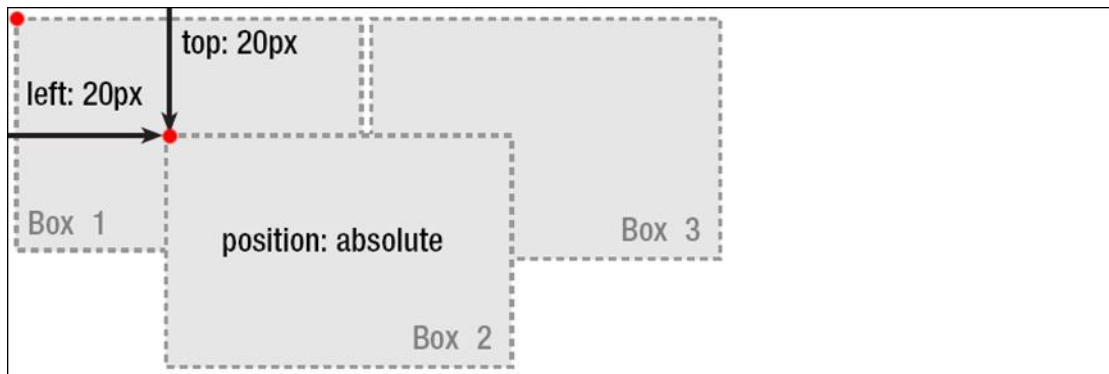


Box2设置了相对定位。  
但仍保留着原来的位置空间。

### 3.3.3 层定位-绝对定位

absolute

定位为absolute的层脱离正常文本流，但与relative的区别:其在正常流中的**原位置不再存在**

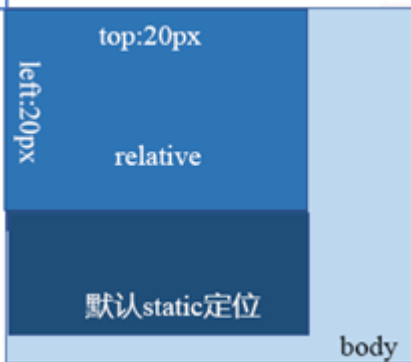


Box2设置了绝对定位，**已脱离了原来的文档流。**

```
#Box2{  
    position: absolute;  
    left: 20px;  
    top: 20px;  
}
```

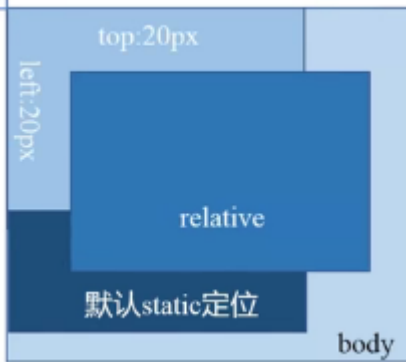
### 3.3.3 层定位-相对/绝对

(0.0)

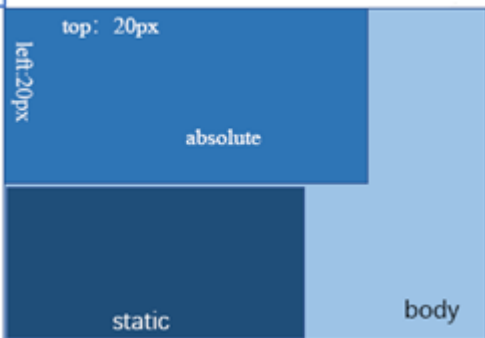


```
#box1{  
  width:170px;  
  height:190px;  
  position:relative;  
  top:20px;  
  left:20px;  
}
```

(0.0)



(0.0)



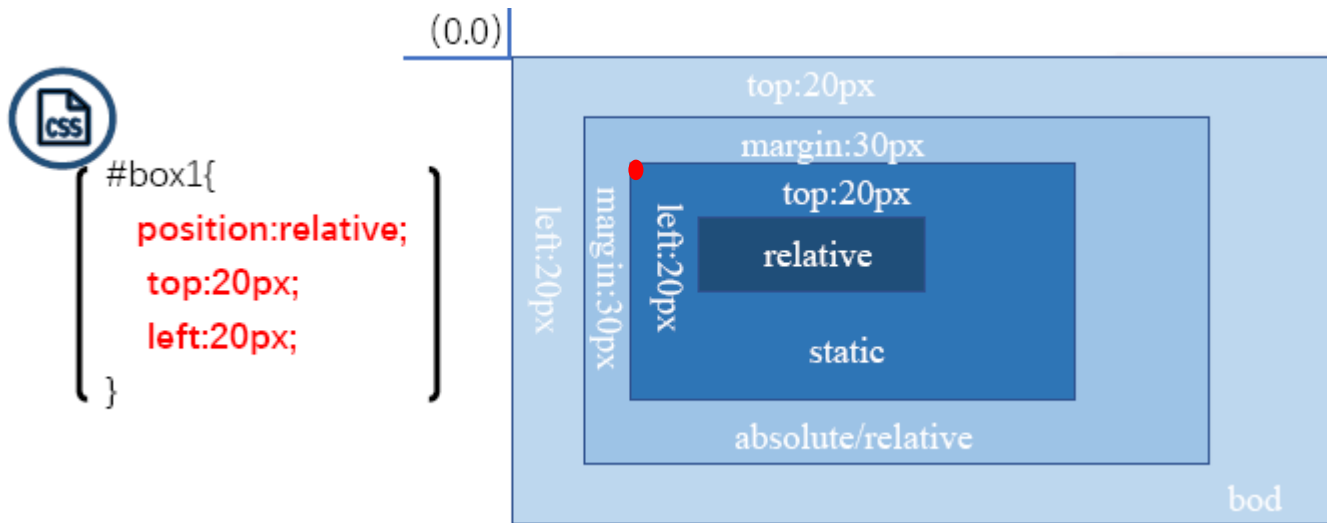
```
#box1{  
  position:absolute;  
  top:20px;  
  left:20px;  
}
```

(0.0)



### 3.3.3 层定位-相对/绝对

**relative** 相对于其在普通流中的位置进行定位。是以自身为参照，意思就是它原本该在什么位置，加上**relative**属性后，会以它原本按照文档流应该在的位置为参照进行定位



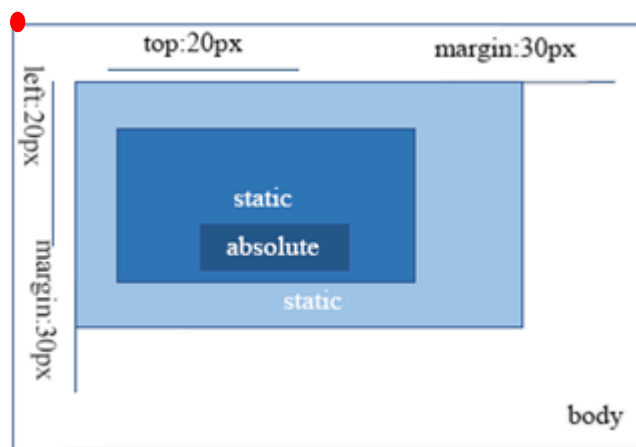
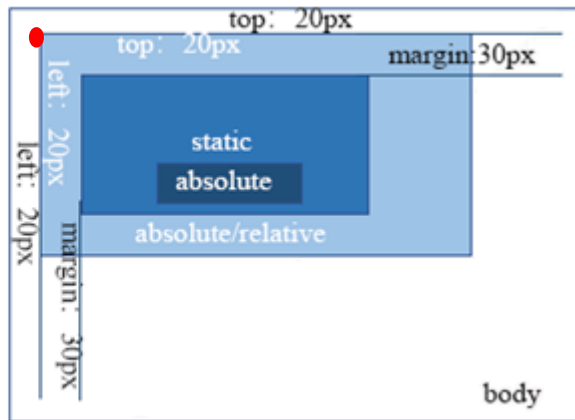
### 3.3.3 层定位-相对/绝对

对于absolute定位的层总是相对于其**最近的定义为absolute或relative的父层**，而这个父层并不一定是其直接父层。  
如果其父层中都未定义absolute或 relative，则其将相对**body**进行定位



```
#box1{  
  position:absolute;  
  top:20px;  
  left:20px;  
}
```

(0,0)



### 3.3.3 relative+absolute

父元素box1:

position:relative;

子元素box2:

position:absolute

top、bottom、left、right相对于父元素来进行偏移定位

```
<div id="box1">  
  <div id="box2">  
  </div>  
</div>
```

```
#box1 {  
  position:relative;  
}  
#box2 {  
  position:absolute;  
}
```

### 3.3.3 relative+absolute

```
div{
  border:1px solid red;
  color: #fff;
}
#box1{
  width:170px;
  height:190px;
  position:relative;
}
#box2{
  width:99%;
  position:absolute;
  bottom:0;
}
```



```
<div id="box1">
```

```
  
```

```
  <div id="box2">一起享受咖啡带来的温暖吧
```

```
  </div>
```

```
</div>
```

### 3.3.3 relative+absolute



父元素: relative  
子元素: absolute  
top取负值  
right取负值



### 3.3.3 层定位-小结

position属性

- fixed: 相对浏览器
- relative: 相对于其在普通流中的位置进行定位
- absolute: 相对于relative/absolute/body定位

left属性

right属性

top属性

bottom属性

z-index属性

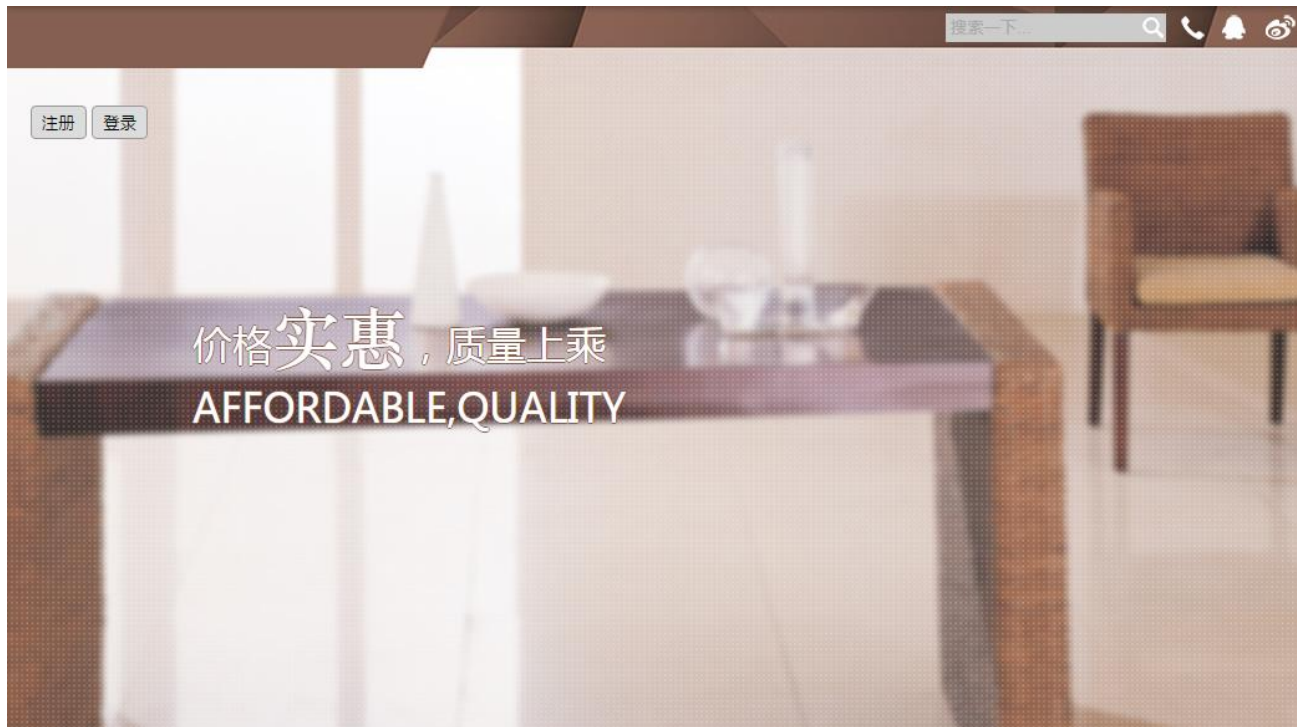
	相对定位	绝对定位
position取值	relative	absolute
文档流中原位置	保留	不保留
定位参照物	其在普通文档流中的位置	非static的父元素

相对定位 **VS** 绝对定位

# 案例：设计实现网页元素的定位

P160

使用CSS的position属性实现网页元素的“任意”摆放，增强网页元素的美化效果



# 综合案例-咖啡店





咖啡MENU | 咖啡COOK | 咖啡STORY | 咖啡NEWS | 咖啡PARTY

### 咖啡MENU

	拿铁	卡布奇诺	摩卡	浓缩咖啡
	Latte	Cappuccino	Mocha	Espresso
大杯	35	35	35	30
中杯	30	30	30	25
小杯	25	25	25	20



### 拿铁Caffè Latte

这是一种传统的经典饮料——浓咖啡注入热牛奶，其上覆盖一层轻盈的奶沫。

品尝此款咖啡时，您可以选择特别加入某种口味（如香草，焦糖或杏仁口味）的糖浆。



### 卡布奇诺Cappuccino

这款咖啡沿袭传统技法，由我们技艺娴熟的咖啡师将手工制作的奶与细腻奶泡轻柔地混在浓缩咖啡之上制作而成。



### 摩卡Caffè Mocha

这款咖啡由醇香的摩卡酱，浓缩咖啡和热奶相融合，上面裹以搅打奶沫。

寒冷的日子里，忧伤的时光中，任何人都无法抵抗她的诱惑。



### 浓缩咖啡Espresso

这是咖啡的粹粹，以最浓缩的方式呈现。浓缩咖啡带点焦糖味道，浓厚馥郁。



网站设计课程版权 2015-2020

# 课后测-盒子模型

- 1、盒子模型中的盒子，可以是任何页面元素。
- 2、盒子模型中的盒子是可以逐层嵌套的。
- 3、**CSS**进行布局与定位，首先用盒子将元素大小、边距等信息确定下来，然后用定位方式决定盒子的位置。
- 4、定位机制分为三种：盒子定位、文档流定位、浮动定位。
- 5、盒子模型中，边框的属性是可以分为4个方向分别设置的。
- 6、溢出盒子的部分，可以使用**overflow**属性，将其设置为**hidden**隐藏起来。

# 课后测-盒子模型

7、下列（ ）表示上边框线宽10px，下边框线宽5px，左边框线宽20px，右边框线宽1px。

- A.border-width:10px 1px 5px 20px
- B.border-width:10px 5px 20px 1px
- C.border-width:5px 20px 10px 1px
- D.border-width:10px 20px 5px 1px

8、在CSS语言中下列哪一项是"左边框"的语法（ ）。

- A.border-left-width: <值>
- B.border-top-width: <值>
- C.border-left: <值>
- D.border-top-width: <值>

# 课后测-盒子模型

9、CSS中，盒子模型的属性不包括（ ）。

- A.border
- B.padding
- C.content
- D.margin

10、下列哪些是设置有关边框的属性（ ）。

- A.border-color
- B.border-style
- C.border-width
- D.border-back

11.文字、图片、div区域水平居中，都可以使用text-align:center进行设置。

12. margin:1px 2px 1px 3px; 可以缩写成 margin:1px 2px 3px;

# 课后测-盒子模型

13、下列哪个属性能够设置盒模型的左侧外边距（ ）。

A.margin

B.indent

C.margin-left

D.padding-left

14、下列哪个CSS属性能够设置盒子模型的内边距为10、20、30、40、（顺时针方向）（ ）。

A.padding:10px 20px 30px 40px

B.padding:40px 30px 20px 10px

C.padding:10px 40px 30px 20px

D.padding:20px 10px 40px 30px

# 课后测-盒子模型

15、阅读下面HTML代码，两个div之间的垂直空白距离是（ ）。

```
<style type="text/css">  
.header { margin: 10px; border:1px solid #f00; }  
.container { margin: 15px; border:1px solid #f00; }  
</style>
```

.....

```
<div class="header"></div>  
<div class="container"></div>
```

- A.0px
- B.10px
- C.15px
- D.25px



# 课后测-盒子模型

16、阅读下面CSS代码，下面选项中与该代码段效果等同的是（ ）。

`.box { margin:10px 5px; margin-right:10px; margin-top:5px; }`

A..`box { margin:5px 10px 0px 0px; }`

B..`box { margin:5px 10px 10px 5px; }`

C..`box { margin:5px 10px; }`

D..`box { margin:10px 5px 10px 5px; }`

17、利用以下什么代码可以设置div区域的水平居中（ ）。

A..`div{margin:0}`

B..`div{margin:auto 100px}`

C..`div{margin:100px auto}`

D..`div{margin:100px 100px }`

# 课后测-定位概述

- 1、浮动定位通过float属性设置来实现。
- 2、默认定位方式就是文档流定位。
- 3、层定位通过position属性来设定。
- 4、对于一个盒子，文档流定位、浮动定位、层定位只能选择其中一个。
- 5、层定位是可以将盒子像图层一样处理，有前后层叠顺序。

# 课后测-文档流定位

- 1、块状元素在网页中就是以块的形式显示，所谓块状就是元素显示为矩形区域，常用的块状元素包括div、h1-h6、p、ul。
- 2、默认情况下，块状元素都会占据一行，通俗地说，两个相邻块状元素不会出现并列显示的现象；默认情况下块状元素会按顺序自上而下排列。
- 3、块状元素都不可以定义自己的宽度和高度。
- 4、块状元素可以作为其他元素的容器，它可以容纳其他行内元素和其他块状元素。

# 课后测-文档流定位

5、下列哪个样式定义后,行级元素可以定义宽度和高度( )。

- A.display:inline
- B.display:none
- C.display:block
- D.display:inheric

6、以下( )是HTML常用的块状标签。

- A.<span>
- B.<a>
- C.<br>
- D.<h1>

# 课后测-浮动定位

- 1、浮动元素在文档流中的原位置依然保留。
- 2、利用float属性将3个div盒子水平排列，可以将3个盒子都设置float:right。
- 3、如果一个父元素的所有子元素都浮动起来，则这个父元素默认高度坍缩为0。

# 课后测-浮动定位

4、clear属性取值说法正确的是( )。

- A.取值为all表示四周浮动元素被删除
- B.取值为left表示左侧浮动元素被删除
- C.取值为right表示右侧浮动元素被删除
- D.取值为both表示清除两侧浮动

5、float属性说法不正确的是( )。

- A.该属性可以用于图文混排
- B.该属性可以用于网页分栏
- C.该属性可以用于盒子层叠
- D.该属性可以用于浮动定位

# 课后测-层定位

- 1、position属性设定，表示定位的参照物是哪个元素；而定位位置则由top,bottom,left,right的取值决定。
- 2、通常将父元素设置为绝对定位，子元素设置为相对定位，就可以将子元素相对于父元素定位了。

# 课后测-层定位

3、如果要将网页中的两个div对象制作为重叠效果，（ ）。

A. 是不可能的

B. 利用表格标记<table>

C. 利用样式表定义中的绝对位置与相对位置属性

D. 利用样式表定义中的z-index属性

4、以下哪个选项是对对象进行定位的（ ）。

A.padding

B.margin

C.position

D.display



# 课后测-层定位

5、position属性取值（ ）表示固定定位。

A.absolute

B.relative

C.static

D.fixed

6、position属性取值（ ）表示相对定位。

A.relative

B.absolute

C.static

D.fixed



谢 谢