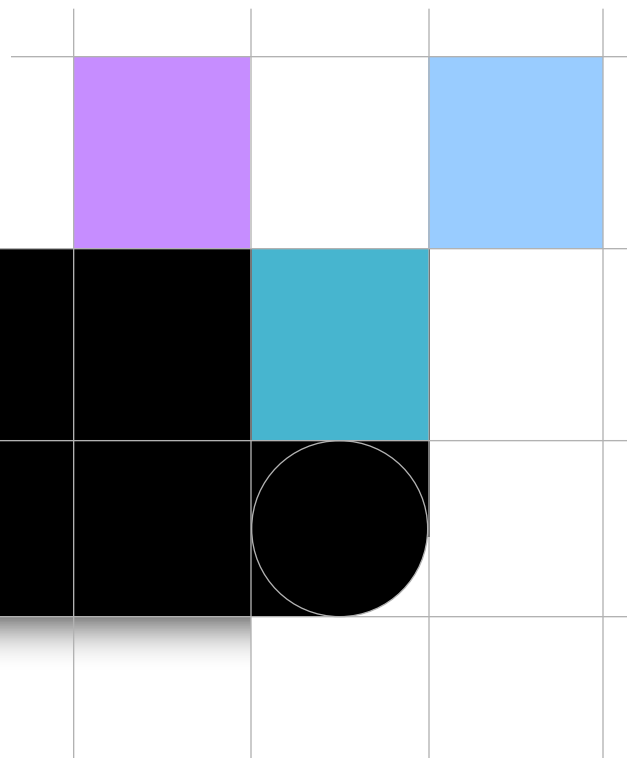
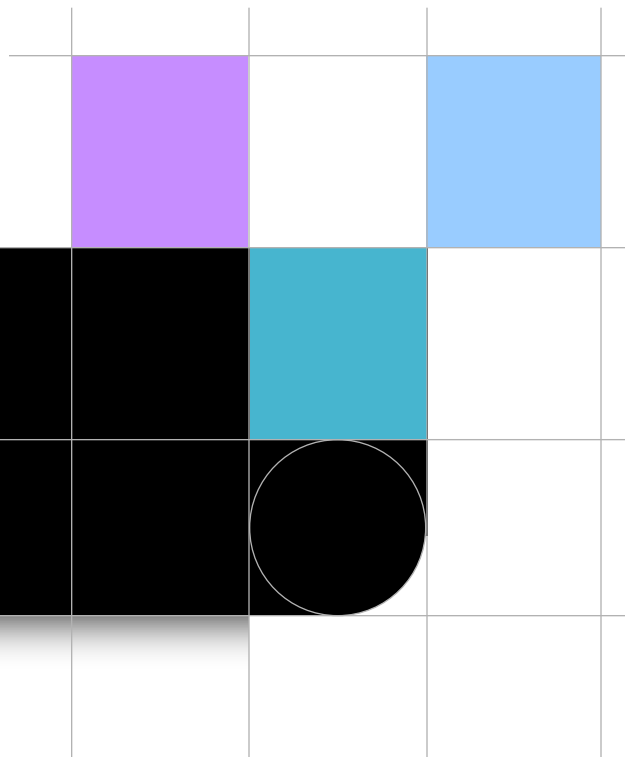


网站设计



信息学院
孙 俏

CSS 布局





Name Surname
Vivamus suscipit sem magna.
Donec euismod
Loremipsum@ipsum.com

My Profile

Your Project!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum a urna feugiat, malesuada nunc et, egestas magna. Quisque id quam sed massa pretium iaculis. Nulla turpis nunc, ultricies sit amet semper nec, eleifend ac nulla. Morbi eu tempor elit. Proin in semper nulla, sit amet lacinia nisi. Praesent ultricies dapibus condimentum. Phasellus eleifend aliquam diam, eu pulvinar odio elementum id. Mauris sagittis tempus lacinia.



Login

Home

Profile

Service

Photos

About Us

Sed faucibus urna a nisi sollicitudin in gravida nibh congue. In hac habitasse platea dictumst. Praesent neque nunc, laoreet id mattis nec, sagittis vel purus. Duis non hendrerit nisi. Curabitur condimentum dapibus lacinia.



Mail



Project

Support

Sed faucibus urna a nisi sollicitudin in gravida nibh congue. In hac habitasse platea dictumst. Praesent neque nunc, laoreet id mattis nec, sagittis vel purus. Duis non hendrerit nisi. Curabitur condimentum dapibus lacinia.



Services

Sed faucibus urna a nisi sollicitudin in gravida nibh congue. In hac habitasse platea dictumst. Praesent neque nunc, laoreet id mattis nec, sagittis vel purus. Duis non hendrerit nisi. Curabitur condimentum dapibus lacinia.



Gallery



Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



Nam cursus

Lorem ipsum dolor sit amet, consectetur adipiscing elit.



Sed accumsan

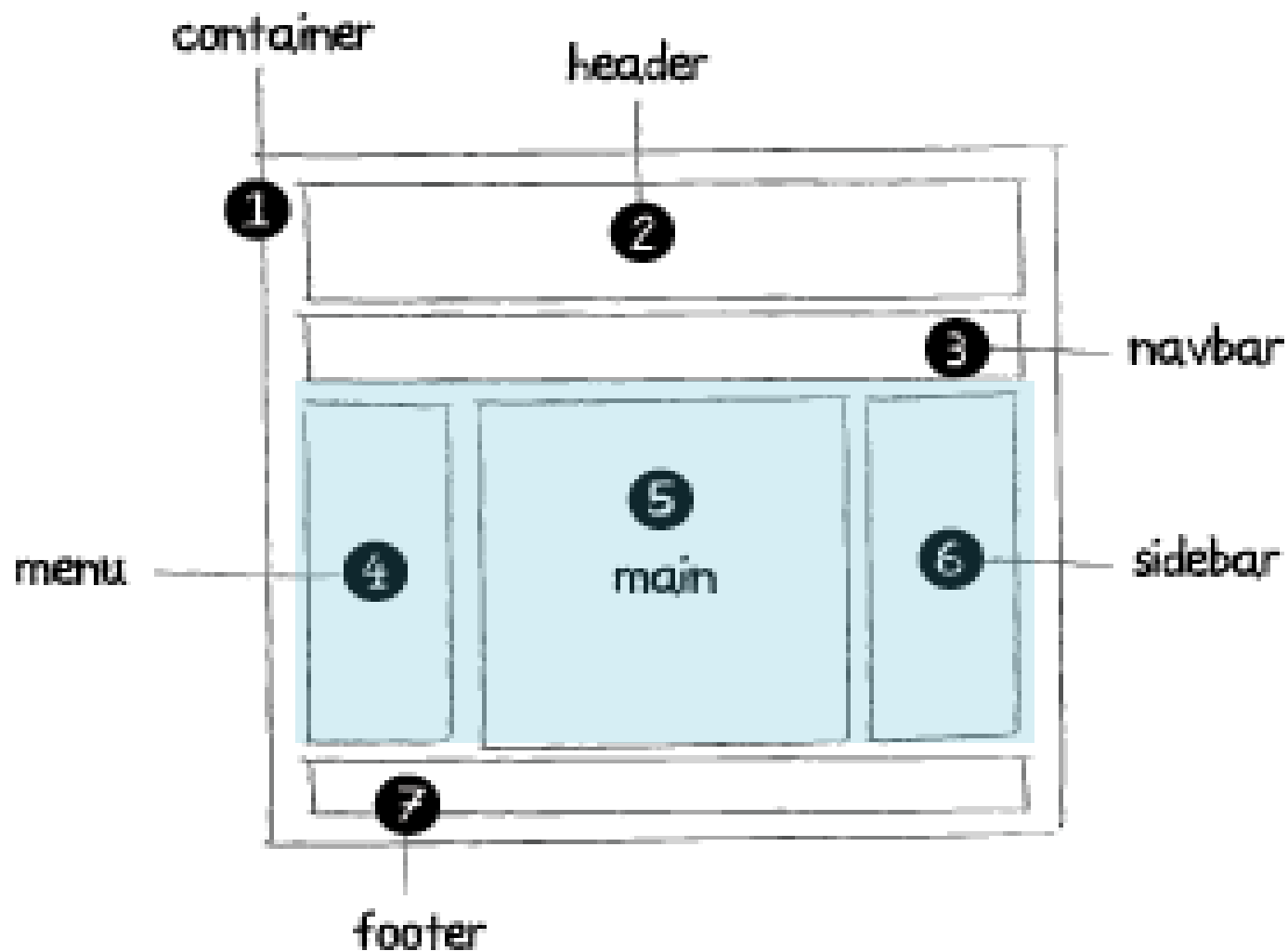
Lorem ipsum dolor sit amet, consectetur adipiscing elit.



Aliquam viverra

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

CSS布局



CSS布局

```
<div id="container">
```

```
  <div id="header">
```

```
  </div>
```

```
  <div id="navbar">
```

```
  </div>
```

```
  <div id="content">
```

```
    <div id="menu" class="aside">
```

```
    </div>
```

```
    <div id="main">
```

```
    </div>
```

```
    <div id="sidebar" class="aside">
```

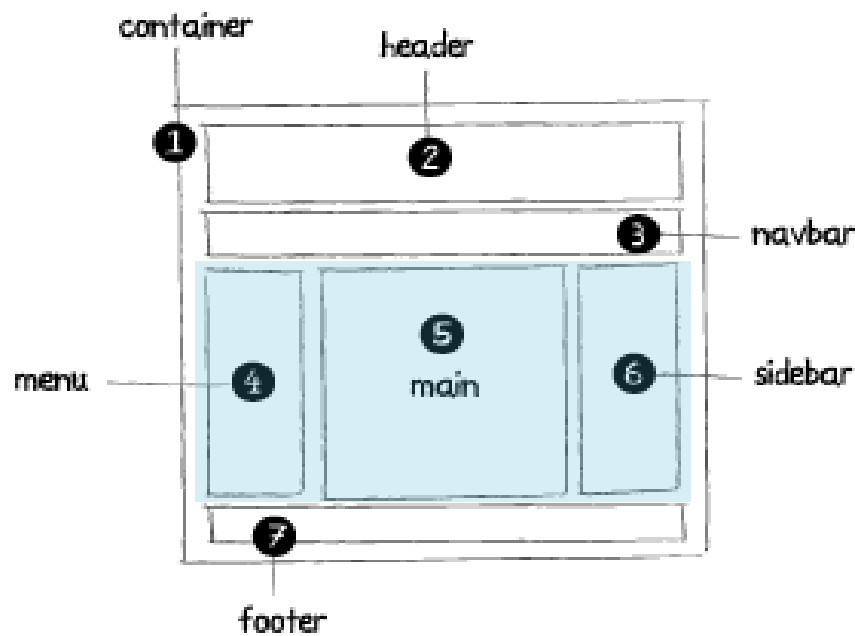
```
    </div>
```

```
  </div>
```

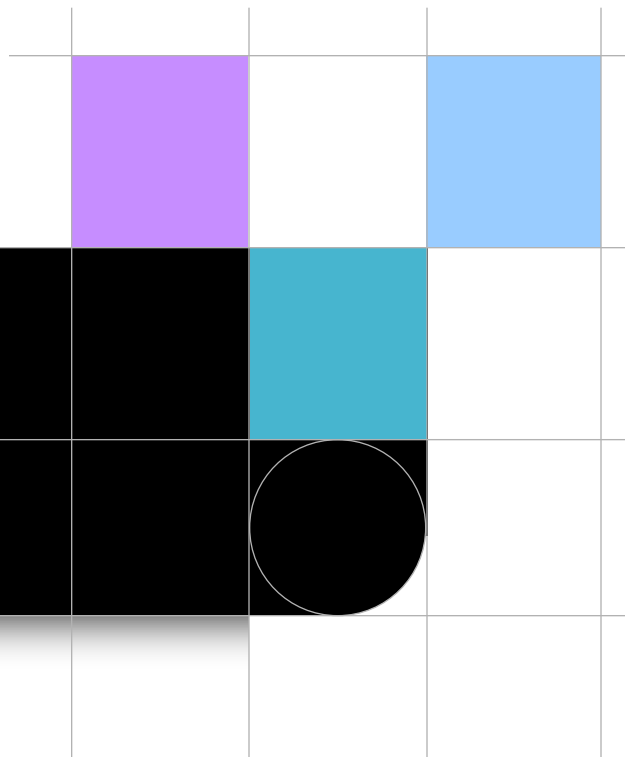
```
  <div id="footer">
```

```
  </div>
```

```
</div>
```



盒子模型



CSS布局——盒子模型

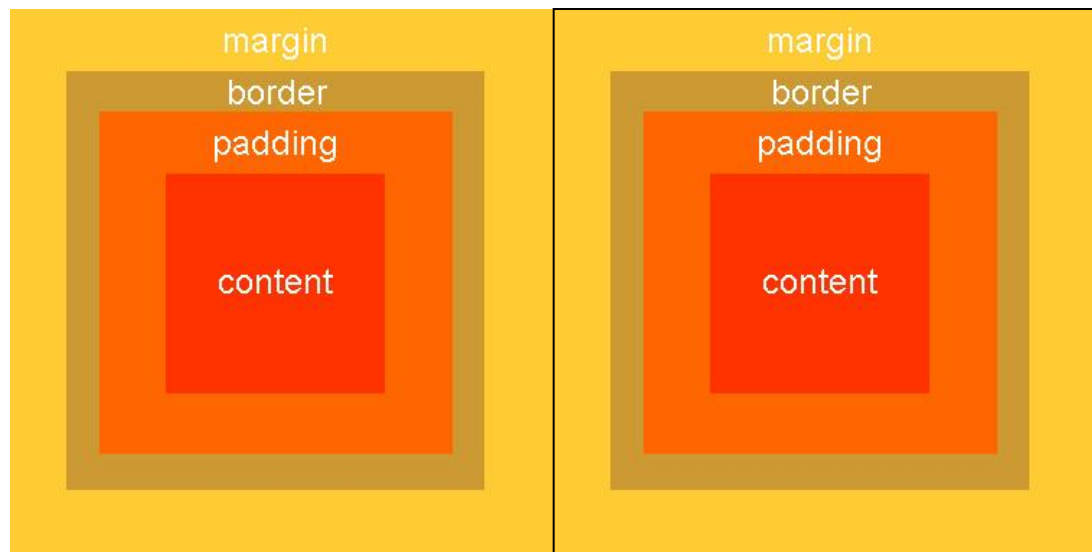


CSS布局——盒子模型

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

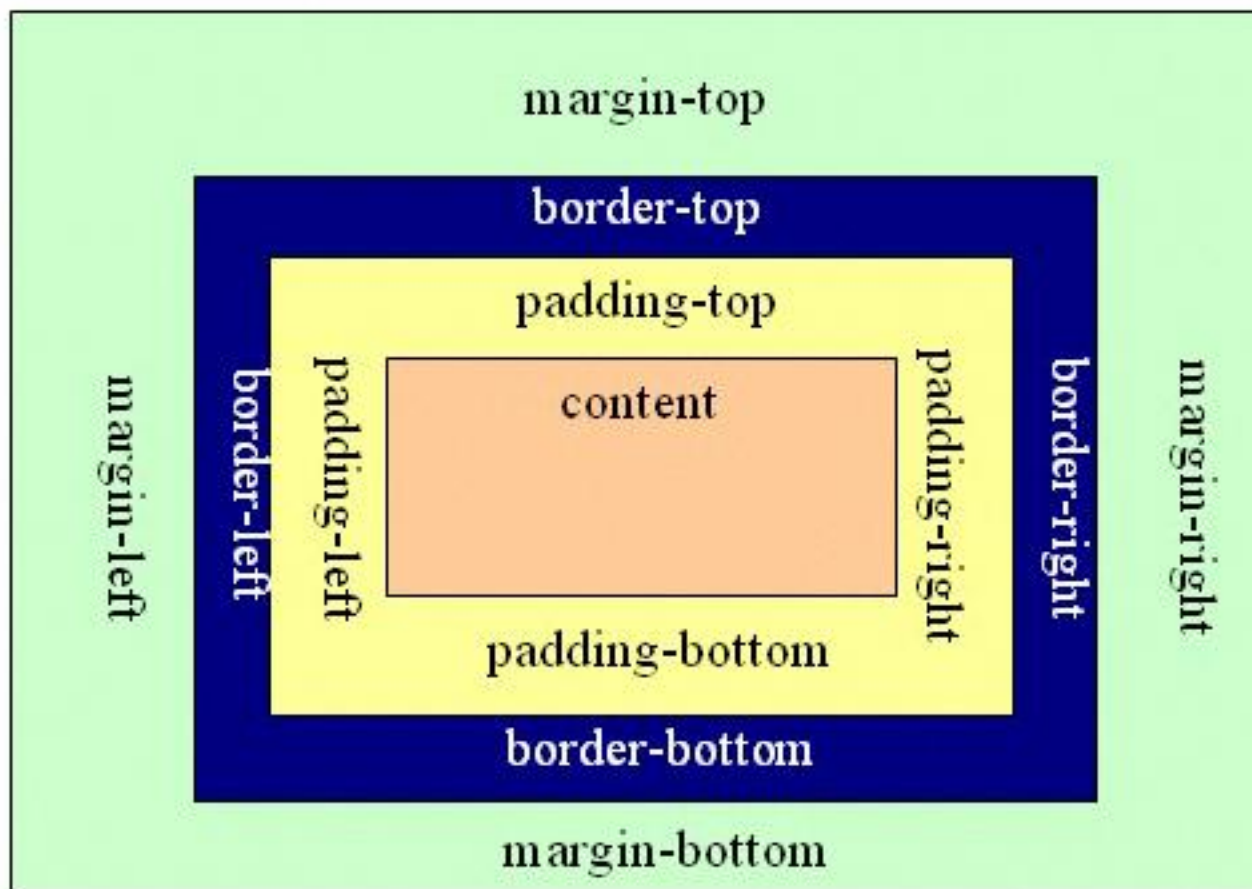
盒子模型组成

- content（内容）
- border（边框）
- padding（填充）
- margin（间隔）

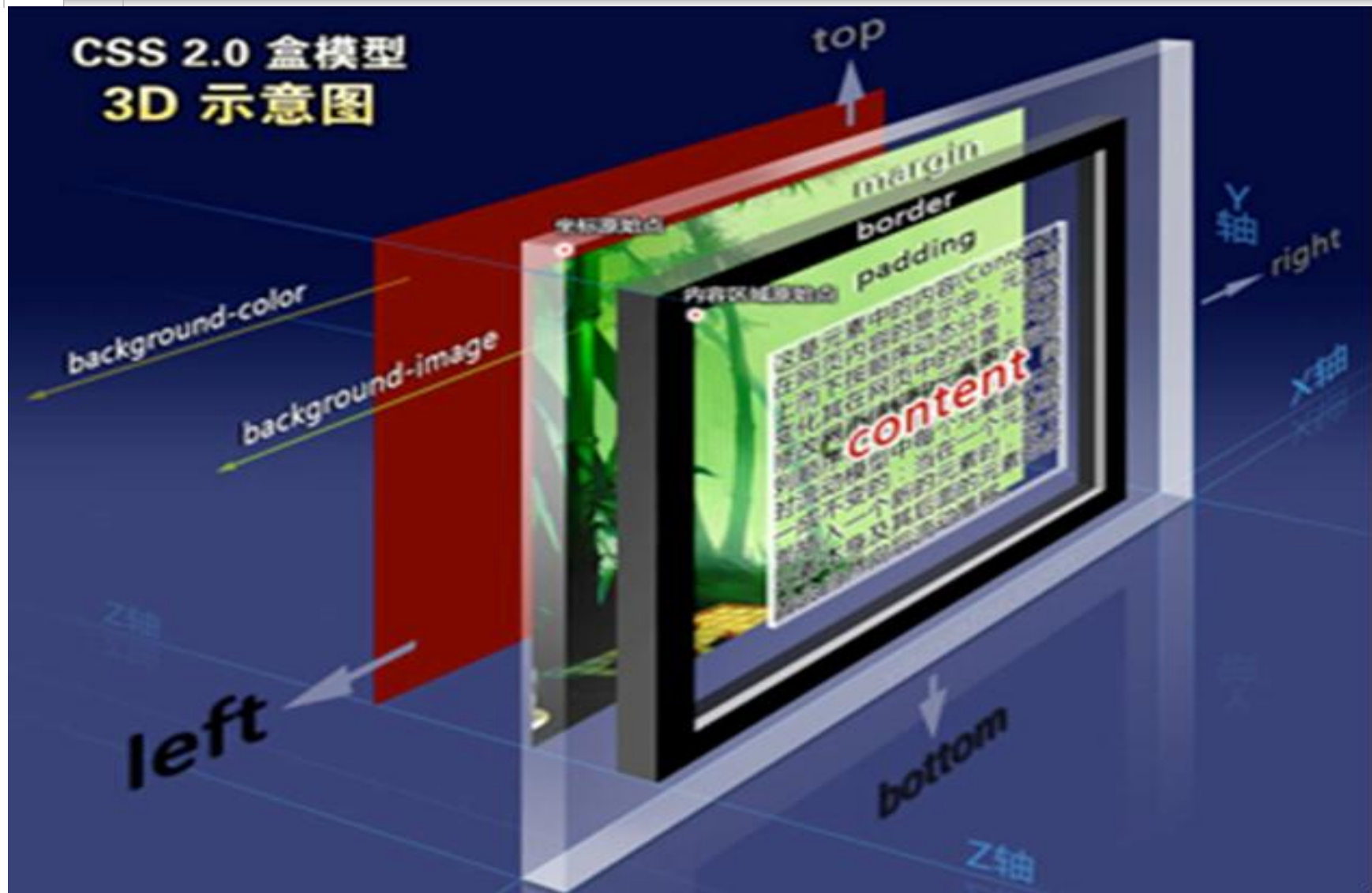


都是CSS样式的属性

CSS布局——盒子模型



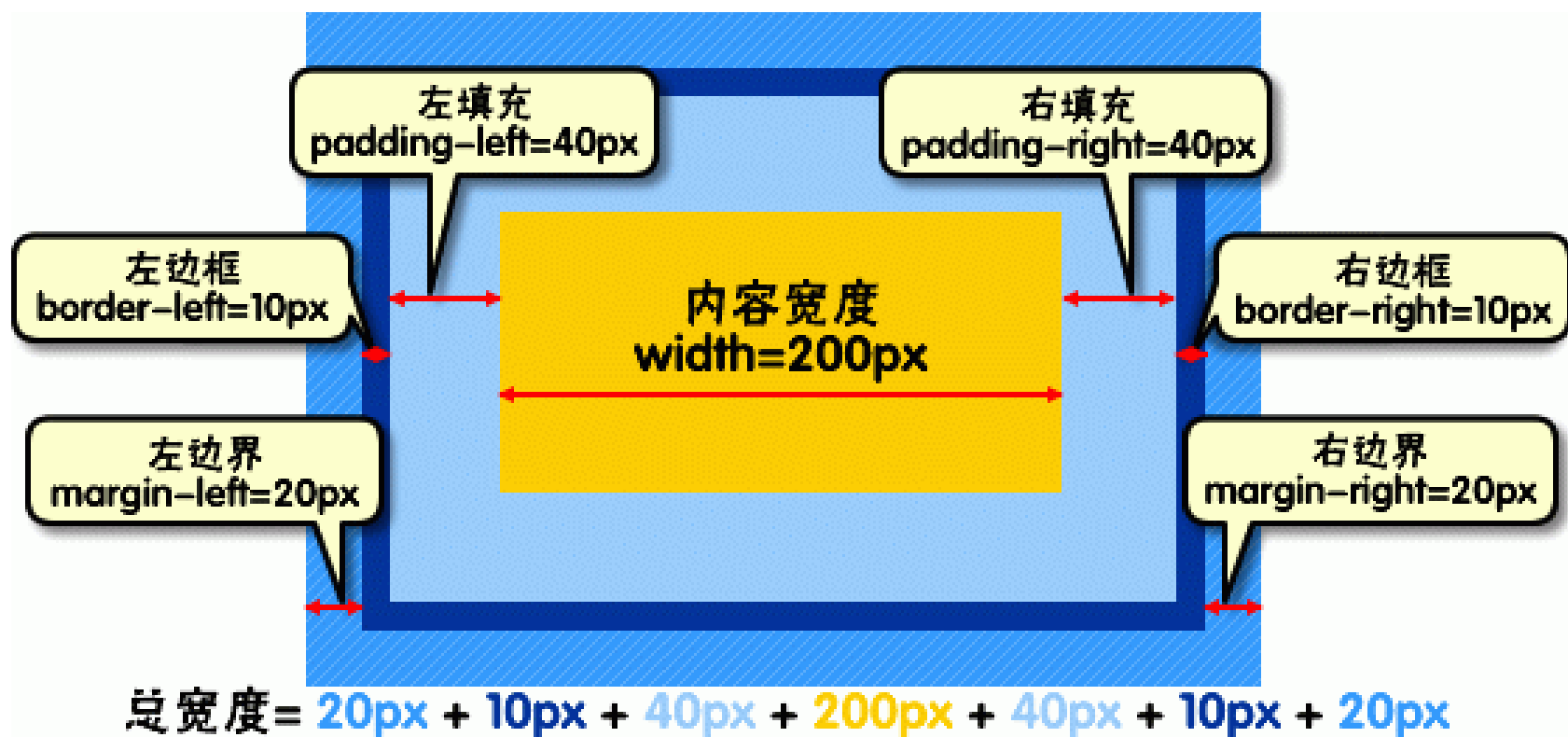
CSS布局——盒子模型



CSS布局——盒子模型

一个盒子的实际宽度、高度：


$\text{content} + \text{padding} + \text{border} + \text{margin}$





CSS布局——盒子模型

```
<html>
<head>
<style type="text/css">
#box1 {
    width:100px;
    height:100px;
    border:1px solid;
    padding:20px;
    margin:10px;
}
</style>
</head>
<body>
<div id="box1">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</div>
</body>
</html>
```





CSS布局——盒子模型 border

- border-width: px、thin、medium、thick
- border-style: dashed、dotted、solid
- border-color: 颜色
- border: width style color;

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

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

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



案例

- 水平分割线

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



CSS布局——盒子模型padding

- **Padding内边距** **margin外边距** (auto: IE自动计算)

属性

描述

padding

上 右 下 左

padding-bottom

下内边距

padding-left

左内边距

padding-right

右内边距

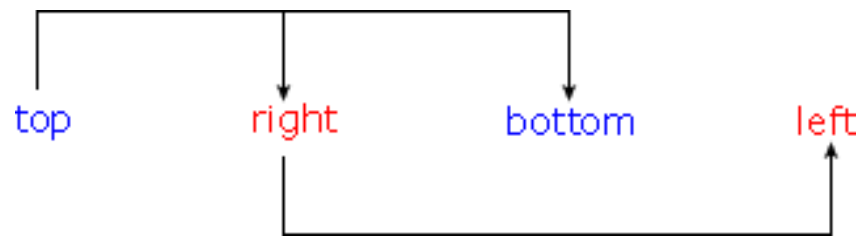
padding-top

上内边距



CSS布局——盒子模型 margin

- `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。

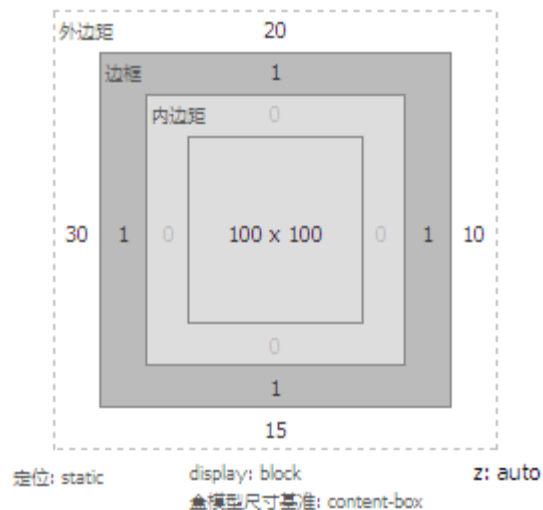
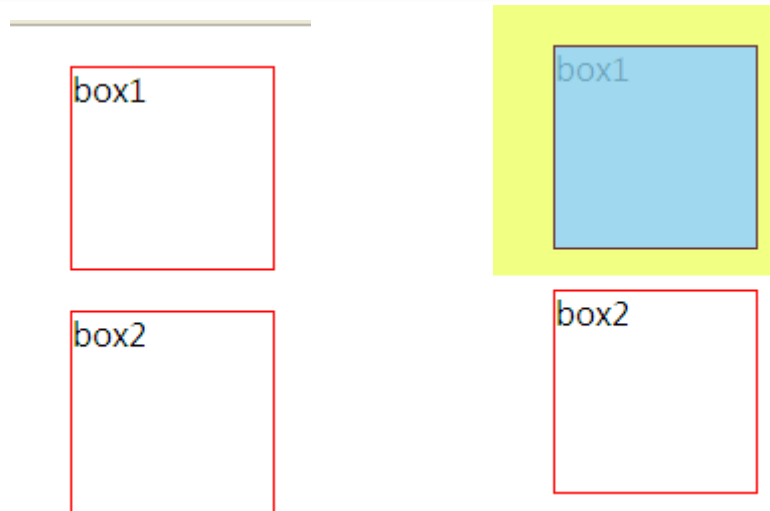


CSS布局——盒子模型 margin

```
<!DOCTYPE HTML>
<html>
<head>
<style type="text/css">
div{
    width:100px;
    height:100px;

    margin:20px 10px 15px 30px;
    border:1px solid red;
}
```

```
</style>
</head>
<body>
    <div id="box1">box1</div>
    <div id="box2">box2</div>
</body>
</html>
```



CSS布局——盒子模型margin

● Margin的合并

合并之前



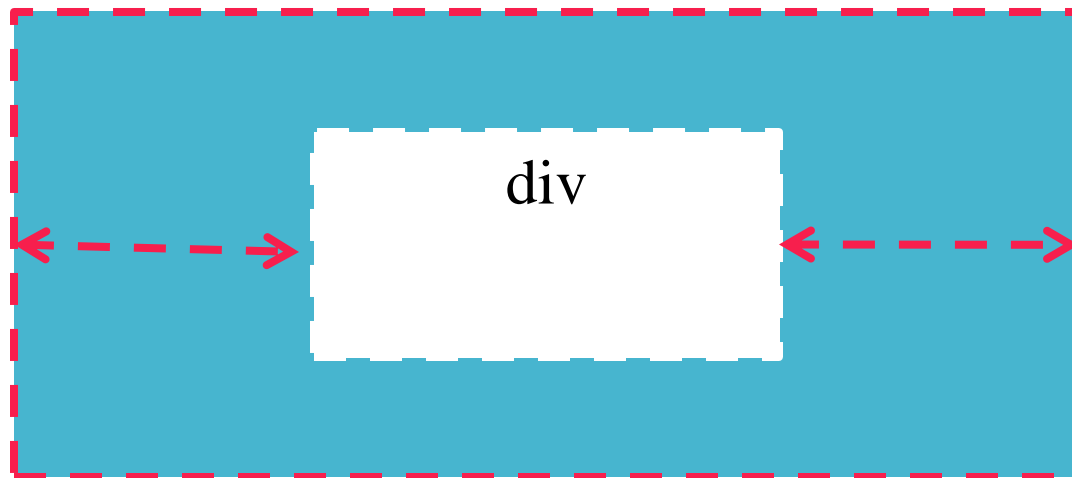
外边距合并
形成一个外边距

合并之后



CSS布局——盒子模型 margin

- 水平居中
 - 图片、文字水平居中
 - `Text-align:center;`
 - Div水平居中
 - `Margin:0 auto;`



案例

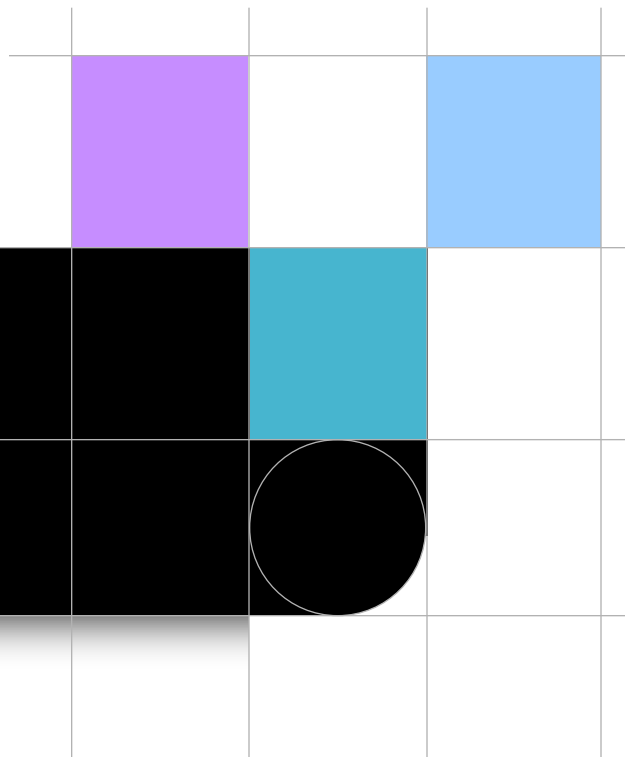




案例

```
#newsimglist {  
    text-align:center;  
    text-indent:0em; /*否则图片有两个字符的缩进*/  
    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;  
    vertical-align: middle; /*由于ie中有默认高度的空文本节点  
    调整图片上下与文本距离相等*/
```

定位机制





CSS布局——定位机制

- CSS中，存在3种的定位机制：
 - 文档流 `flow`
 - 浮动 `float`
 - 层 `layer`



CSS定位机制——文档流flow

- 文档流
- 特点
 - 从上到下，从左到右，输出文档内容
 - 由块级元素block和行级inline元素组成
 - block从上到下独占一行
 - Inline从左到右占据一行





CSS定位机制——文档流flow

- block元素与inline元素转换
- display属性：
 - none 不会被显示
 - block 显示为块级元素
 - inline 默认。显示为内联元素
 - Inline-block 兼具两者特征



练习

[链接1](#)

[链接2](#)

[链接3](#)



CSS定位机制——浮动布局float

- block实现横向多列布局
- float属性：
 - left—左浮动
 - right—右浮动
 - none—不浮动



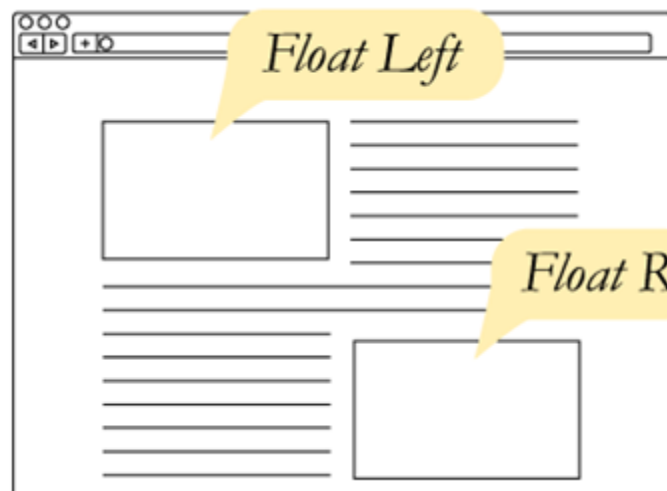
CSS定位机制——浮动布局float

- `div {`
- `width: 200px;`
- `height: 200px;`
- `border: 2px red solid;`
- `float: left;`
- `}`
- `<div id="div1"></div>`
- `<div id="div2"></div>`



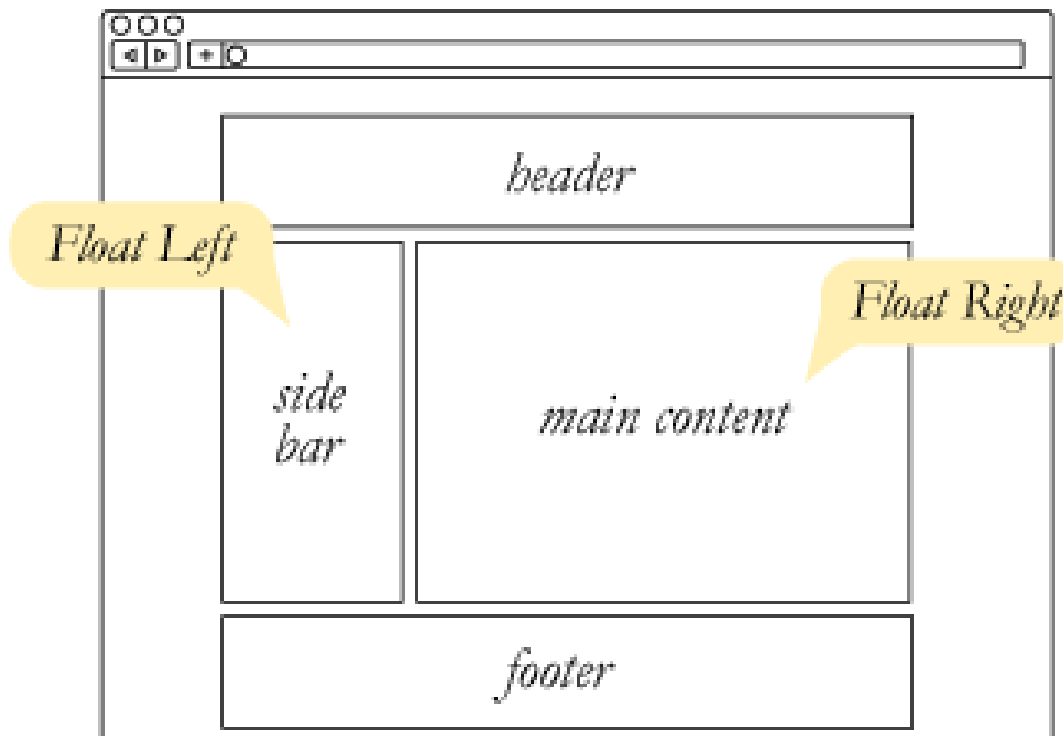
float 详解——用处

● float 用处



1

Web Layout



2

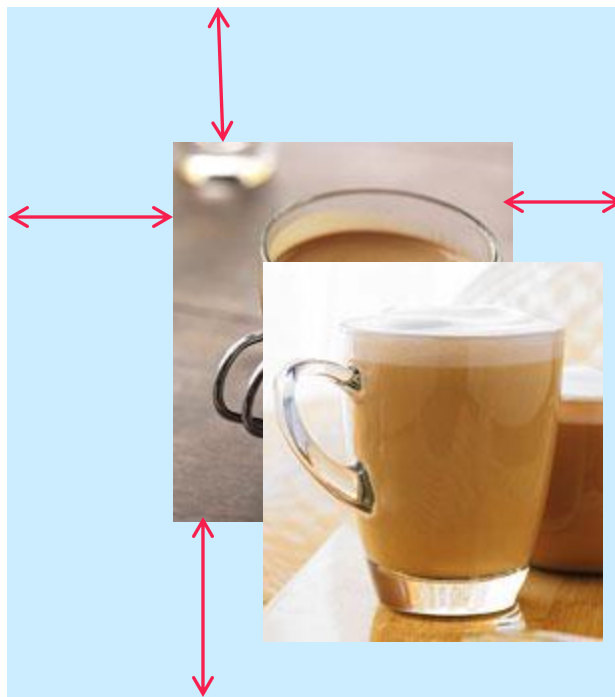


CSS定位机制——layer

- 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：
 - absolute 绝对定位
 - relative 相对定位
 - fixed 固定定位

CSS定位机制——layer

- 通过以下属性定位
 - top, bottom, left, right
 - z-index





position

- 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: 绝对定位
 - 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
 - 相对于其正常位置进行定位
 - 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



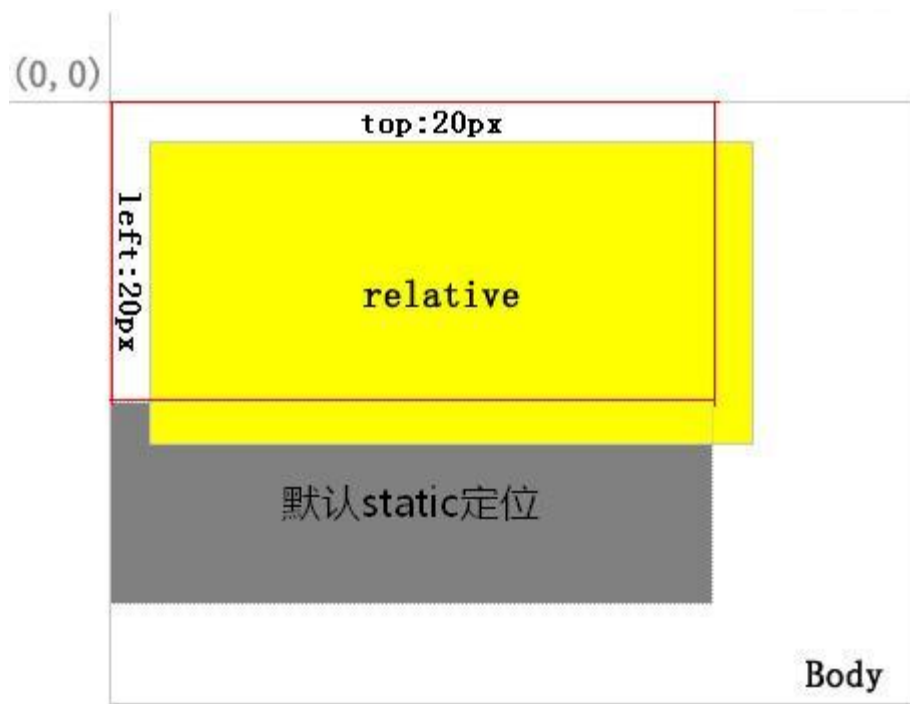
CSS定位机制——position:fix

不会随浏览器窗口的滚动条滚动而变化
总在视线里的元素

```
#div1 {  
    width:200px;  
    height:200px;  
    border:2px red solid;  
    position:fixed;  
    left:100px;  
    top:50px;  
}
```

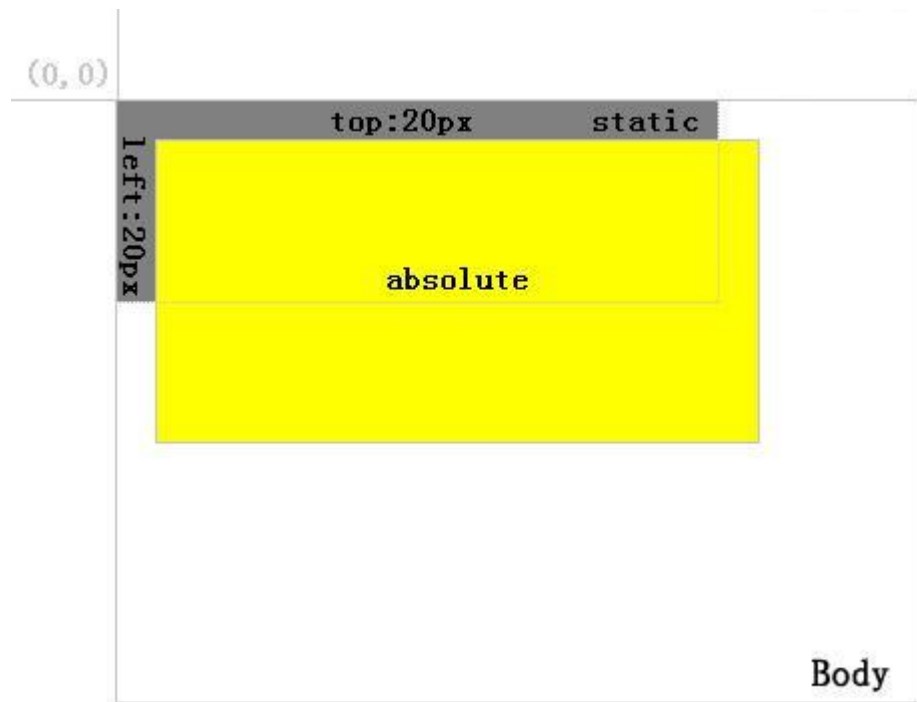
position——relative

- relative。定位为relative的元素脱离正常的文本流中，但其在文本流中的位置依然存在



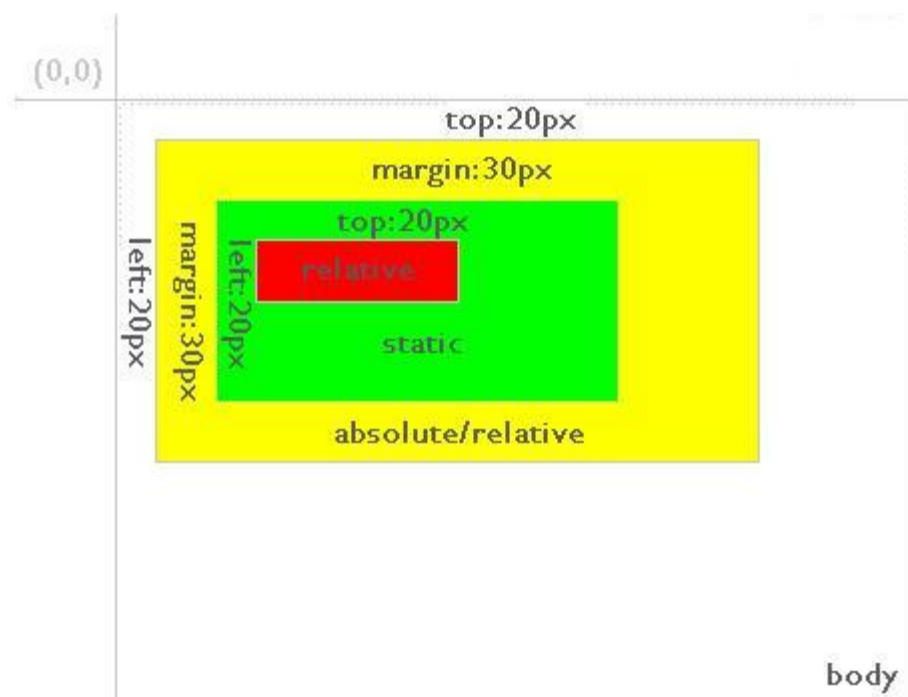
position——absolute

- absolute。定位为absolute的层脱离正常文本流，但与relative的区别是其在正常流中的位置不存在



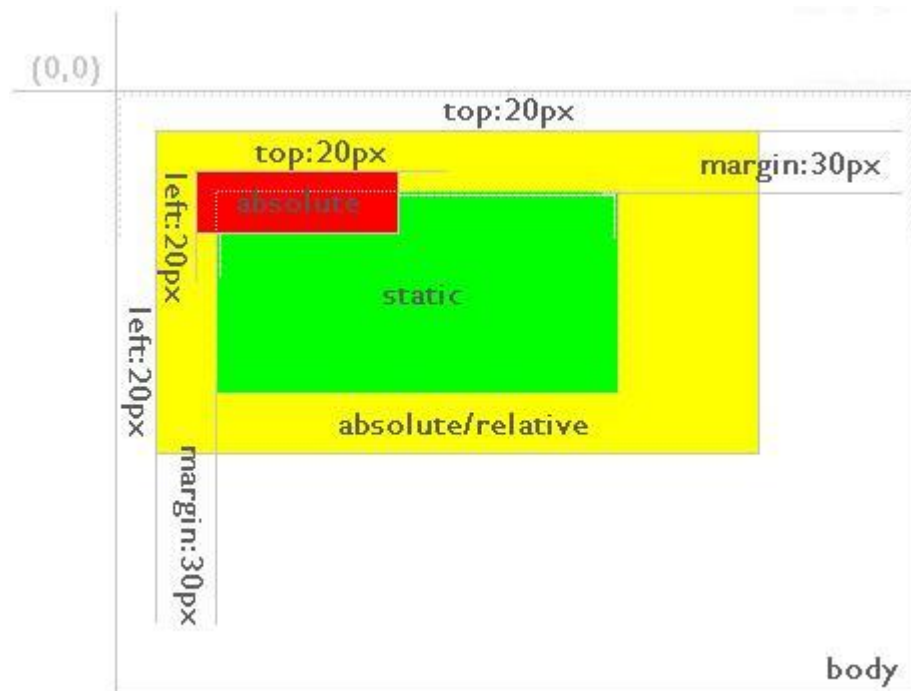
position——relative vs absolute

- relative与absolute的主要区别：
- 首先，是上面已经提到过的在正常流中的位置存在与否。
- 其次，relative定位的层总是相对于其最近的父元素，无论其父元素是何种定位方式。



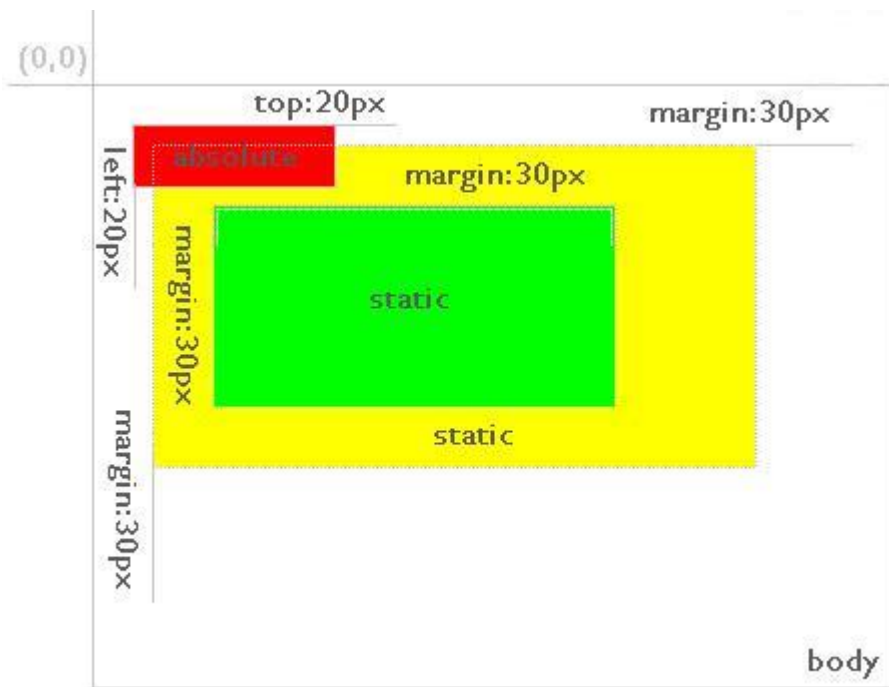
position——absolute

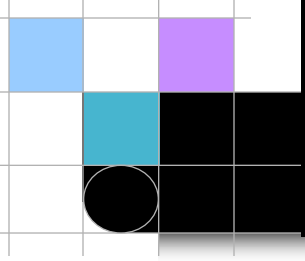
- 对于absolute定位的层总是相对于其最近的定义为absolute或relative的父层，而这个父层并不一定是其直接父层。



position——absolute

- 对于absolute定位的层, 如果其父层中都未定义absolute或relative, 则其将相对body进行定位





CSS定位机制—— relative+absolute

- 父元素box1: `position:relative;`
- 子元素box2: `position:absolute`
- 子元素: `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>
```


CSS定位机制—— relative+absolute

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

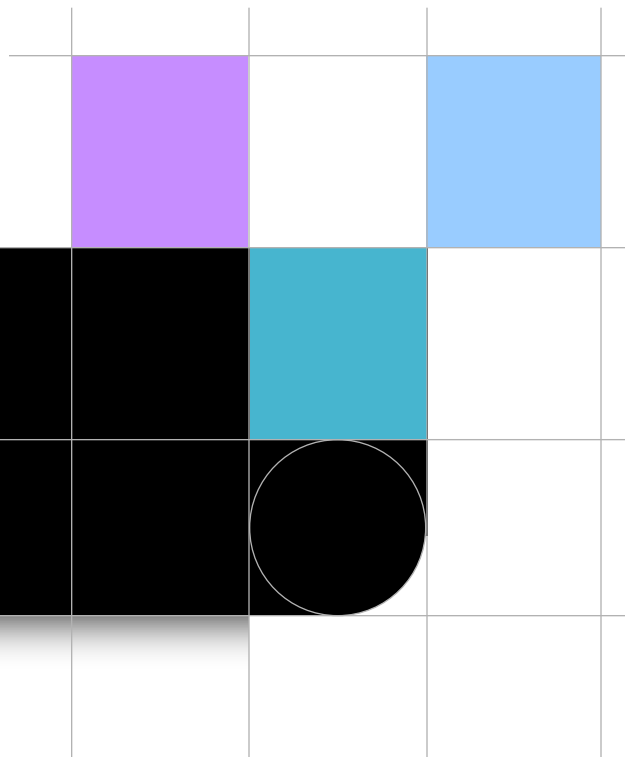
```
#box1{  
    width:200px;  
    height:200px;  
    position:relative;  
}
```

```
#box2{  
    width:99%;  
    position:absolute;  
    bottom:0;  
}
```

```
<div id="box1">  
      
    <div id="box2">当我还是三年级的学生时是一个害羞的小女生。  
</div>  
</div>
```



居中





文字与图片在div居中

```
<style>
```

```
.textcenter {text-align:center}
```

```
</style>
```

```
<body>
```

```
<div class="textcenter"> <img src= "1.gif" /></div>
```

```
</body>
```



文字与图片垂直居中

- 单独文字垂直居中

```
#footer {  
    height: 60px;  
    line-height: 60px;  
}
```

- 文字与图片同排

```
.yangshi img {vertical-align: middle;}
```

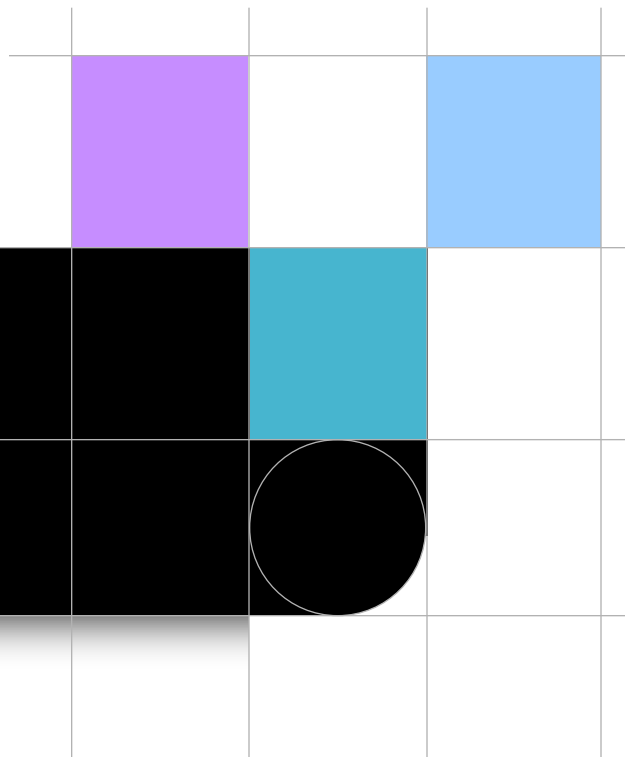


Div水平居中

- 对div对象设置margin

```
#container {  
    Width:900px;  
    margin:0 auto;  
}
```

元素分类





CSS定位机制——文档流flow

- 文档流
- 特点
 - 从上到下，从左到右，输出文档内容
 - 由块级元素block和行级inline元素组成
 - block从上到下独占一行
 - Inline从左到右占据一行





元素分类——block，inline

- 常用的块状元素有：

- <div>、<p>、<h1>、、、

- 常用的行级元素有：

- <a>、、

-

- 相互转换：

- display:block

- display:inline