



陆冬云 qq,wx: 184 3 185 83

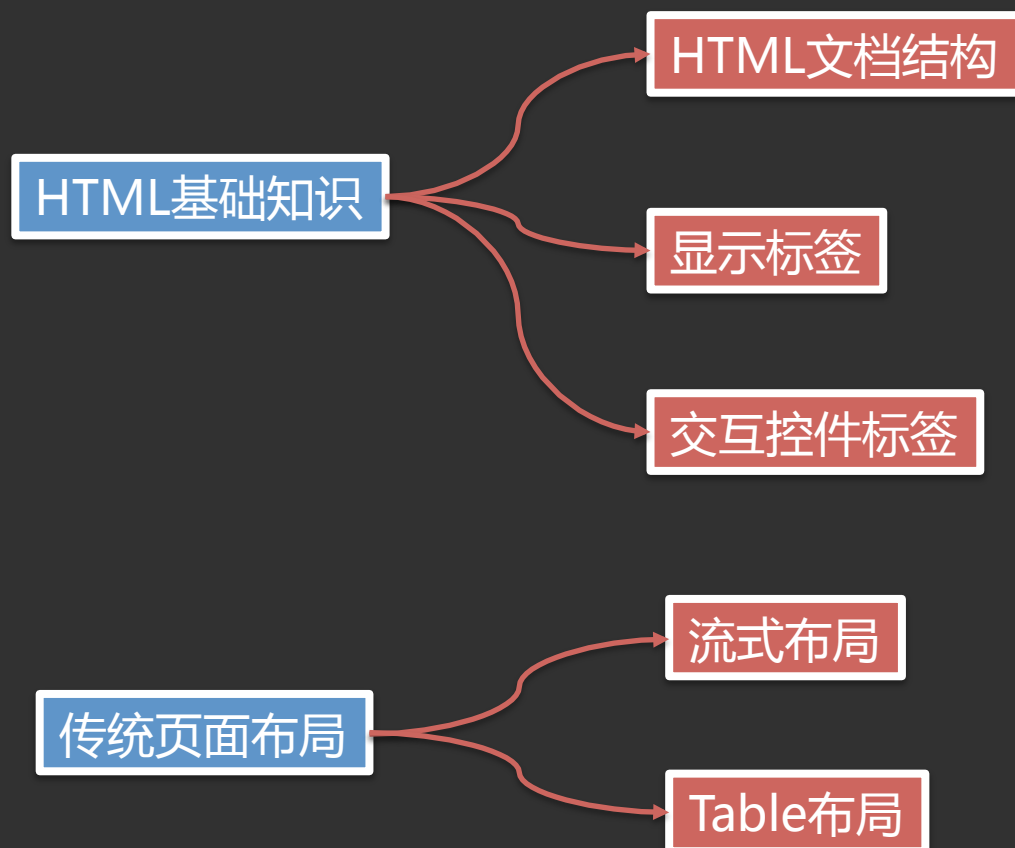
# *Python*开发

## HTML 基础

# HTML基础



陆冬云 qq,wx: 184 3 185 83



# HTML-标记语言



```
<Tag key= "value" ....>  
    Some-content
```

```
.....  
</Tag>
```

缩写的情况：

```
<Tag .... />
```

# HTML-文档结构



陆冬云 qq,wx: 184 3 185 83

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>越狱 第2季..</title>
  </head>

  <body >
    Hello World!
  </body>

</html>
```



# HTML-标签元素



1. body: 最大的容器
2. 文字：纯显示
3. 图片：纯显示
4. 超链接：点击跳转
5. 输入元素：
  - 输入框
  - 按钮等

# body标记



```
<body bgcolor= "red" >
```

```
Hello world
```

```
</body>
```

# 文本

```
<font size= "5"  color= "red" >  
    Hello world  
</font>
```

```
<b>..
```

```
<i>..
```

```
<h1>...</h1>
```

# 图片

```

```



# 超链接



```
<a href="aa/bb/cc.gif">click me</a>
```

# URL地址概念



src , href : 都是URL地址

有两种地址 :

1. 服务器地址 : 以 / 开始
2. 相对地址 : 以当前被访问文件所在目录开始

# 其他常用标记

`<br>` : 换行

`<hr>` : 水平线

`<pre>` : 保留文本内部格式

特殊字符 :

`&nbsp;`

`&gt;`

`&lt;`

`&#1234`



陆冬云 qq,wx: 184 3 185 83

# 输入元素



陆冬云 qq,wx: 184 3 185 83

```
<input type="text" name="xm" value="Alice" />
```

## Type :

- text
- password
- button
- submit
- radio
- checkbox
- hidden

账号:

密码:

☐ 男 ☒ 女

☒ 南京 ☒ 上海 ☐ 香港

# 制作输入表单



目标：使用input标签，完成注册页面。

1. 用户登录账号
2. 密码、确认密码
3. 性别，城市

参考form.html

# 基本页面制作



陆冬云 qq,wx: 184 3 185 83

目标：使用前面介绍的标签，完成一个产品的简介页面。

1. 产品标题
2. 产品图片
3. 产品简介内容

## <夏季清爽凉鞋>

---



塑料凉鞋适宜于夏天，一双凉鞋可以穿两个夏天。第一个夏天鞋子稍大一些，到了第二个夏天便有些紧了。没法穿的凉鞋可以剪下塑料，来修补那些穿坏了补补还能再穿的凉鞋。把烙铁在火塘里烧红，把鞋子上坏了的鞋口对接好，贴上剪来的塑料块，一阵浓浓的塑料味的青烟飘过，鞋子便粘好了。

参考[shoe\\_font.html](#)

# 布局元素：table



陆冬云 qq,wx: 184 3 185 83

```
<table border= "0|1" >  
  <tr>  
    <td>...</td>  
    <td colspan= "2" >...</td>  
  </tr>  
  ...  
</table>
```

# Table表格-学生清单



目标：制作表格

1. 创建行、列
2. 控制列数

参考



# table表格-图文布局



陆冬云 qq,wx: 184 3 185 83

## <夏季清凉凉鞋>



塑料凉鞋适宜于夏天，一双凉鞋可以穿两个夏天。第一个夏天鞋子稍大一些，到了第二个夏天便有些紧了。没法穿的凉鞋可以剪下塑料，来修补那些穿坏了补补还能再穿的凉鞋。把烙铁在火塘里烧红，把鞋子上坏了的茬口对接好，贴上剪来的塑料块，一阵浓浓的塑料味的青烟飘过，鞋子便粘好了。

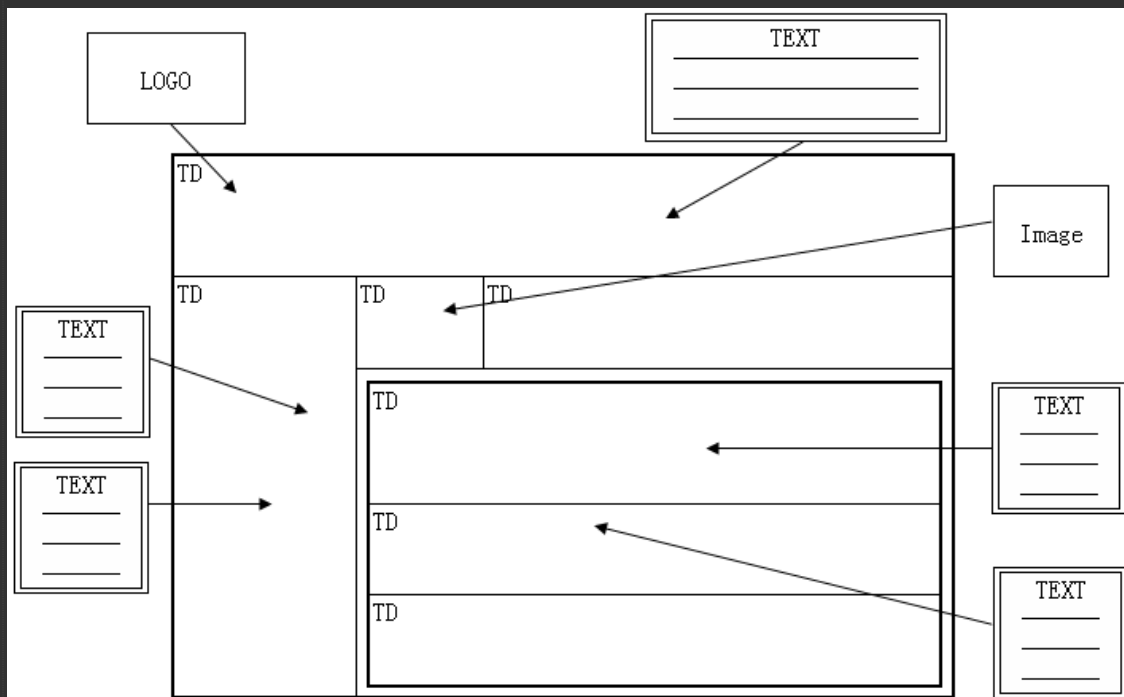
参考[table-shoe.html](http://table-shoe.html)

# HTML-Table的缺陷



陆冬云 qq,wx: 184 3 185 83

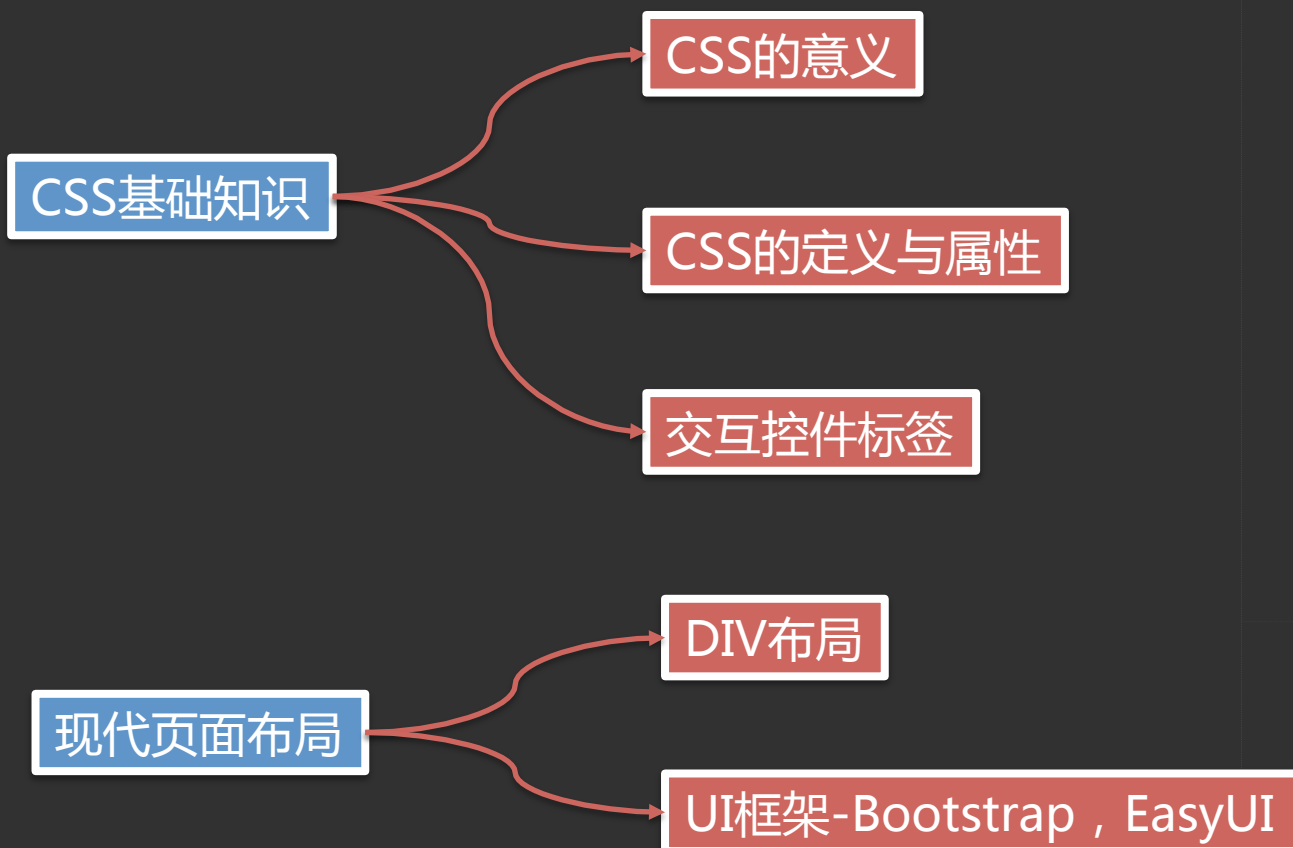
1. 结构复杂
2. 调整困难
3. 复用低效



# HTML基础知识



陆冬云 qq,wx: 184 3 185 83



# CSS的意义-形式与内容的分离



陆冬云 qq,wx: 184 3 185 83

## 无法分离的情况：

```
<font size=6 color=red>  
<b>游子吟</b>  
</font>  
<br/>  
<font size=3 color=green> <i>  
慈母手中线，游子身上衣。  
临行密密缝，意恐迟迟归。  
谁言寸草心，报得三春晖。  
</i> </font>
```

## 分离的情况：

```
<div class= "mytitle" >  
游子吟  
</div>  
<div class= "mytext" >  
慈母手中线，游子身上衣。  
临行密密缝，意恐迟迟归。  
谁言寸草心，报得三春晖。  
</div>
```

# CSS-标签元素



陆冬云 qq,wx: 184 3 185 83

## 行级元素

`<span>....`

`</span>`

## 块级元素

`<div style= "" class= "" >`

`....`

`</div>`

# CSS-作用范围

<style>

\*{...}

span{...}

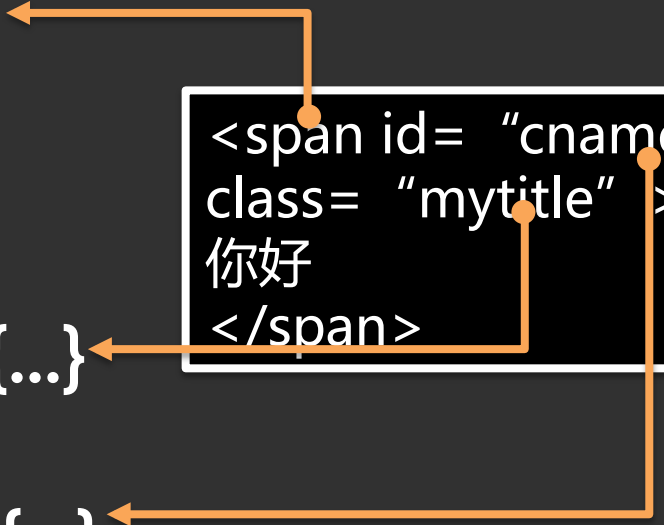
P H1 {...}

P, H1 {...}

.mytitle {...}

#cname {...}

</style>



```
<span id= "cname"  
class= "mytitle" >  
你好  
</span>
```

# CSS定位方式



陆冬云 qq,wx: 184 3 185 83

## 集体声明:并列关系

```
<style type= "text/css" >  
body,td,th,#header,.one{color:blue;font-size:12px;}  
</style>
```

## 嵌套声明:从属关系

```
<style type= "text/css" >  
P h1{color:blue;font-size:14px;}  
</style>
```

## 混合声明:并列及从属关系

```
<style type= "text/css" >  
P H1,#header ul{color:blue;font-size:12px;}  
</style>
```

# CSS-属性



陆冬云 qq,wx: 184 3 185 83

## 样式

背景、文本、字体、链接、列表、表格、轮廓

## 框模型

内边距、边框、外边距

## 定位

相对定位、绝对定位、浮动



# CSS-属性



陆冬云 qq,wx: 184 3 185 83

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>测试页</title>
<link href="css/style1.css" rel="stylesheet" type="text/css" /> <!-- 此为链接
( 外部 ) 样式表 >
<style type="text/css">
<!--此为内嵌 ( 内部 ) 样式表
body,td,th {font-size: 12px;}
p{font-size: 24px;color:#FFFF00;}
-->
</style>
</head>
<body>
<div id="container">
<div id="header"> <p>头部</p> </div>
<div id="content">
<p style="color:#FF0000;font-size:36px">主体</p> <!-- 此为行内样式表 >
</div>
<div id="footer"> <p>尾部</p> </div>
</div>
</body>
</html>
```

# CSS-属性



陆冬云 qq,wx: 184 3 185 83

## 背景：

```
background-color: gray;  
background-image: url(/i/eg_bg_04.gif);  
background-repeat: repeat-y|no-repeat;  
background-position: 50px 100px;  
background-attachment: fixed
```

## 文本：

```
text-align: center  
text-indent: 5em;  
letter-spacing: 20px  
text-transform: uppercase|lowercase|capitalize;  
text-decoration: none|underline|overline|line-through|blink;  
white-space: nowrap|pre
```

# CSS-属性



陆冬云 qq,wx: 184 3 185 83

## 字体：

```
font-family: Georgia, serif, sans-serif;  
font-style:normal|italic|oblique;  
font-weight:normal|bold|900;  
font-size:3.75em;  
color:red
```

## 链接：

```
a:link {color:#FF0000;}  
a:visited {color:#00FF00;}  
a:hover {color:#FF00FF;}  
a:active {color:#0000FF;}
```

## 列表：

```
ul {list-style-type : square}  
ul li {list-style-image : url(xxx.gif)}  
li {list-style : url(example.gif) square inside}
```

# CSS-清零

```
body , * {
```

```
margin: 0 auto;  
padding: 0;
```

```
}
```

# 制作《游子吟》

目标：css的基本结构与用法

1. 在哪里定义
2. 在哪里使用
3. 属性、值与效果

学生目标

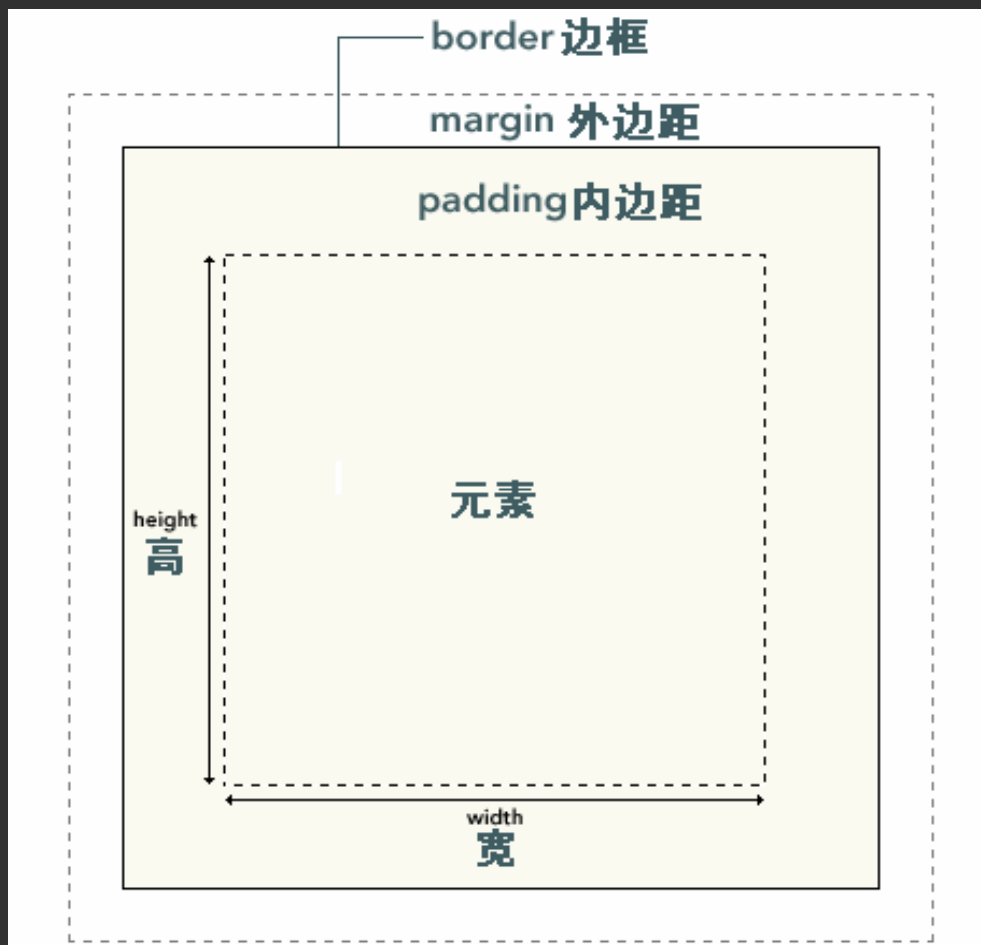
1. 突出放大显示标题
2. 浅色显示诗词部分

现场发挥练习

# CSS-盒子模型



陆冬云 qq,wx: 184 3 185 83



left、top、  
right bottom、

color、 style

# CSS-属性



陆冬云 qq,wx: 184 3 185 83

## 盒子参数

```
width:100px;  
height:100px;  
margin: 0 0 0 0;  
margin-left: 10px;  
margin-top: 10px;  
margin-right: 10px;  
margin-bottom: 10px;  
  
padding: 0 0 0 0;  
border: 1px solid red;  
color: red;  
background-color: red;  
word-wrap:break-word;
```

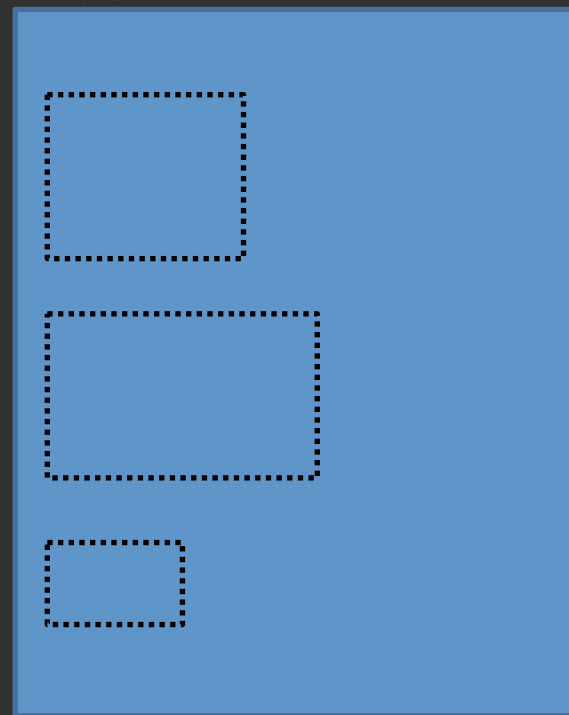
# CSS-布局



陆冬云 qq,wx: 184 3 185 83

核心原则：

1. 块元素包括：div , p , h1等
2. 块元素排列：占单独一行





# Div基本用法



目标：div的空间尺寸与默认排列

1. 定义大小
2. 控制定位
3. 填充文字内容

现场发挥练习

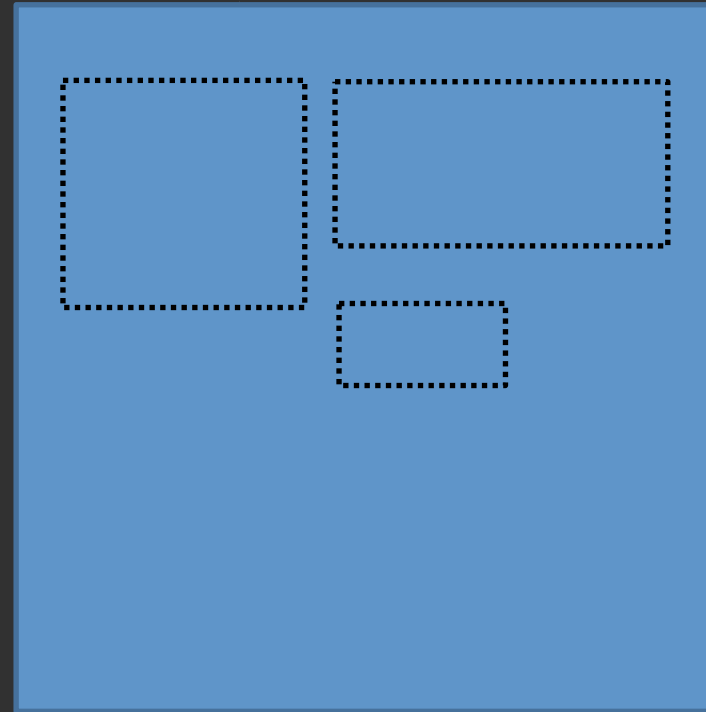
# Div-浮动



陆冬云 qq,wx: 184 3 185 83

float: left | right

1. 浮动不占空间
2. 浮动水平排列
3. 浮动换行



# Div-浮动清除

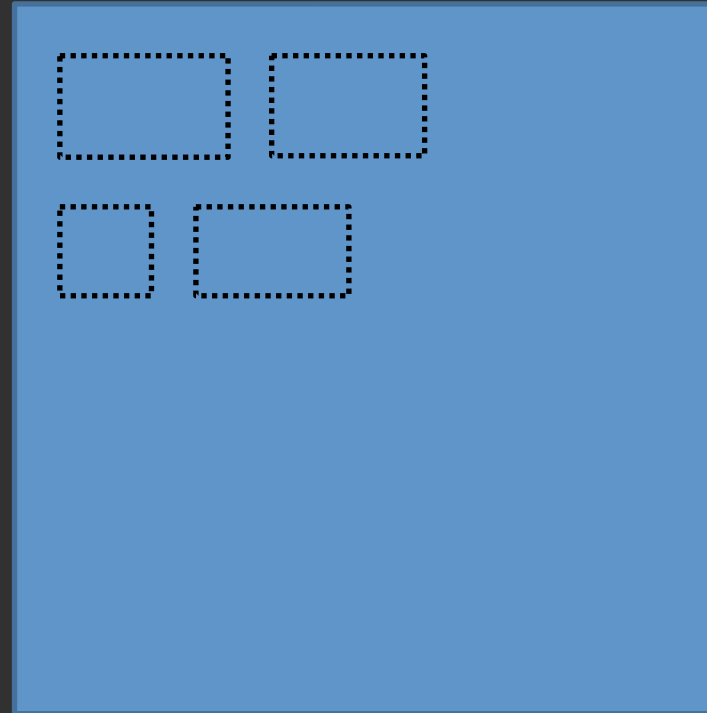


陆冬云 qq,wx: 184 3 185 83

clear: left | right | both

1. 控制换行

2. 父容器自动高度延展



# Div浮动与清除用法



目标：水平排列控制

1. 通过浮动水平排列
2. 控制块元素的换行

参考[clear1.html](#)

# Div浮动与清除用法



目标：父容器的高度

1. 子容器通过浮动水平排列
2. 通过添加空div清除浮动
3. 强制扩展了父容器的高度，及其后div的排列

参考[clear2.html](#)

# Div-浮动与文本排版



1. 文本环绕浮动的块元素排版
2. div与img都是块元素，但有一个不同

# 使用CSS制作鞋子的详情图



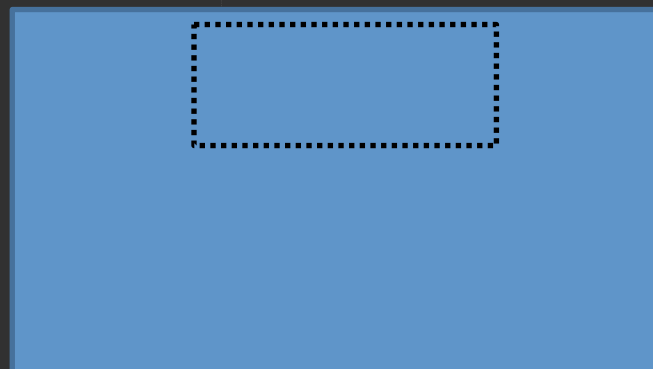
目标：使用浮动，观察文本与图片的关系

1. 图片标题的居中
2. 图片浮动
3. 文字环绕

参考[shoe\\_css.html](#)

# 布局-块元素的水平居中

margin: 0 auto





# 文本的水平居中



陆冬云 qq,wx: 184 3 185 83

`text-align: center`

你好

# 水平居中练习



目标：

让文本在块容器内或者独立水平居中

参考[center\\_h.html](#)

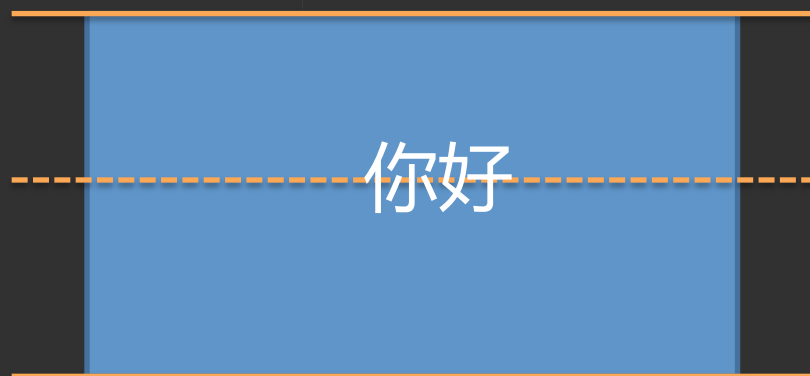
# 文本的垂直居中

行高=div高度

line-height: 80px



陆冬云 qq,wx: 184 3 185 83



# 水平垂直练习



陆冬云 qq,wx: 184 3 185 83

目标：  
让文本在块容器内垂直居中

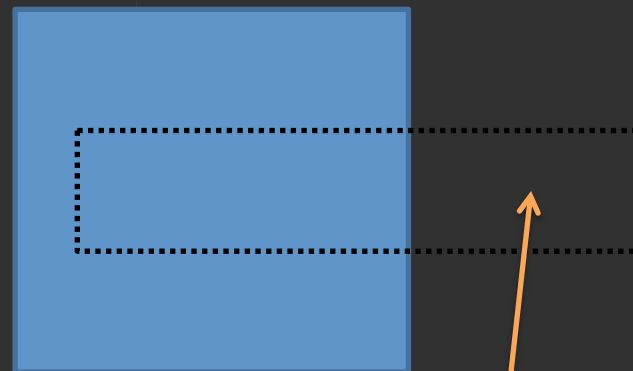
参考[center\\_v.html](#)

# Overflow属性:裁剪功能



陆冬云 qq,wx: 184 3 185 83

父容器设置此属性时，超出父容器的部分不显示



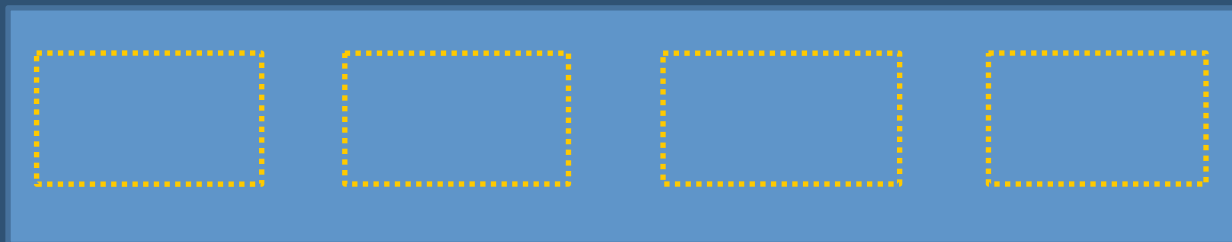
不显示

# 高级：等距排列子容器



目标：在屏幕正中的容器内均匀排列4个方块

1. 浮动、margin
2. 各尺寸估计与计算
3. 多层容器裁剪



参考[even.html](http://even.html)

# 综合运用-制作电商首页



目标：制作首页

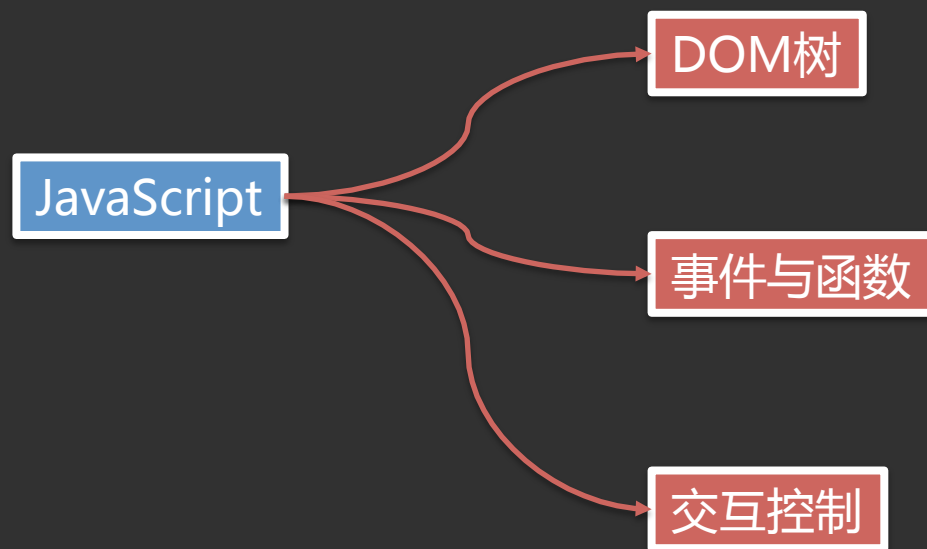
1. 990px宽度
2. 使用head.png做头部推广图片
3. 制作菜单
4. 制作2行8件商品

参考ecom.html

# HTML JS编程



陆冬云 qq,wx: 184 3 185 83





# Javascript-基本语法



变量定义：

```
var a=1;
```

万能类型：布尔，整数，实数，字符串

对象类型：Json对象

# Javascript-基本语法



陆冬云 qq,wx: 184 3 185 83

函数定义：

```
function getinfo(){  
    ...;  
}
```

# JS程序放在哪里



陆冬云 qq,wx: 184 3 185 83

```
<html>
<head...
  <script>
    document.write( "hello world" );
  </script>
</head>

<body>
  <script>
    document.write( "this is a test!" );
  </script>
</body>
</html>
```

程序可以放置在  
任何地方

# JS程序运行时机



陆冬云 qq,wx: 184 3 185 83

```
<html>
<head...
  <script>
    function abc(){
      ....}
  </script>
</head>
<body>
```

1.立即运行  
2.事件触发

```
<input ... onclick= "abc()" onfocus= ".." />
<img ... onclick= "abc()" />
<tr ... onclick= "abc()" />
< a href= "javascript: abc();,..;" >...</a>
</body>
</html>
```

# 学习JS的执行时机



目标：使用document对象测试write的工作效果

1. 在head去区使用write
2. 在body区使用write

参考[document.html](#)

# Javascript-Dom树



window.

只有一个，全局对象，提供全局方法

document.

可以有多个，对应body元素，是视窗区域  
form.

一个想服务器传递信息的单元

\$element.

任意元素，如div，tr，button

# JS提供的方法



陆冬云 qq,wx: 184 3 185 83

`alert()` 显示带有一段消息和一个确认按钮的警告框。  
`blur()` 把键盘焦点从顶层窗口移开。  
`clearInterval()` 取消由 `setInterval()` 设置的 timeout。  
`clearTimeout()` 取消由 `setTimeout()` 方法设置的 timeout。  
`close()` 关闭浏览器窗口。  
`confirm()` 显示带有一段消息以及确认按钮和取消按钮的对话框。  
`createPopup()` 创建一个 pop-up 窗口。  
`focus()` 把键盘焦点给予一个窗口。  
`moveBy()` 可相对窗口的当前坐标把它移动指定的像素。  
`moveTo()` 把窗口的左上角移动到一个指定的坐标。  
`open()` 打开一个新的浏览器窗口或查找一个已命名的窗口。  
`print()` 打印当前窗口的内容。  
`prompt()` 显示可提示用户输入的对话框。  
`resizeBy()` 按照指定的像素调整窗口的大小。  
`resizeTo()` 把窗口的大小调整到指定的宽度和高度。  
`scrollBy()` 按照指定的像素值来滚动内容。  
`scrollTo()` 把内容滚动到指定的坐标。  
`setInterval()` 按照指定的周期(以毫秒计)来调用函数或计算表达式。  
`setTimeout()` 在指定的毫秒数后调用函数或计算表达式。

# JS的document对象



陆冬云 qq,wx: 184 3 185 83

`close()` 关闭用 `document.open()` 方法打开的输出流，并显示选定的数据。

`getElementById()` 返回对拥有指定 id 的第一个对象的引用。

`getElementsByName()` 返回带有指定名称的对象集合。

`getElementsByTagName()` 返回带有指定标签名的对象集合。

`open()` 打开一个流，以收集来自任何 `document.write()` 或 `document.writeln()` 方法的输出。

`write()` 向文档写 HTML 表达式 或 JavaScript 代码。

`writeln()` 等同于 `write()` 方法，不同的是在每个表达式之后写一个换行符。



# Javascript-document对象



陆冬云 qq,wx: 184 3 185 83

- all[] 提供对文档中所有 HTML 元素的访问。
- anchors[] 返回对文档中所有 Anchor 对象的引用。
- applets 返回对文档中所有 Applet 对象的引用。
- forms[] 返回对文档中所有 Form 对象引用。
- images[] 返回对文档中所有 Image 对象引用。
- links[] 返回对文档中所有 Area 和 Link 对象引用。

# 学习dom定位元素并操纵



目标：使用document定位操作input组件

1. 获取输入框的内容
2. 修改输入框的内容
3. 修改radio的选择

参考[\*test2\\_dom\\_input.html\*](#)

# JS-全局对象



陆冬云 qq,wx: 184 3 185 83

window.

open,close,alert...

document.

write, getElementById(),  
getElementByName()

history.

back,go

Math.

PI, sqrt, sin,cos,log,random...

Location.

reload, replace

# JS-事件



陆冬云 qq,wx: 184 3 185 83

onblur 元素失去焦点。  
onchange 域的内容被改变。  
onclick 当用户点击某个对象时调用的事件句柄。  
ondblclick 当用户双击某个对象时调用的事件句柄。  
onfocus 元素获得焦点。  
onreset 重置按钮被点击。  
onresize 窗口或框架被重新调整大小。  
onselect 文本被选中。  
onsubmit 确认按钮被点击。

onload 一张页面或一幅图像完成加载。  
onabort 图像的加载被中断。  
onerror 在加载文档或图像时发生错误。  
onunload 用户退出页面。

# 学习dom定位元素并操纵



陆冬云 qq,wx: 184 3 185 83

目标：使用document定位select列表组件

1. 获取当前选择项
2. 修改默认选择项
3. 感知用户的选择变化

*参考test3\_dom\_select.html*

# 学习dom定位元素并操纵



目标：使用blur事件校验用户输入

1. 输入框失去焦点时触发
2. 校验

参考[test4\\_dom\\_blur\\_verify.html](http://test4_dom_blur_verify.html)

# JS-鼠标事件



陆冬云 qq,wx: 184 3 185 83

键盘事件：

onkeydown 某个键盘按键被按下

onkeypress 某个键盘按键被按下并松开

onkeyup 某个键盘按键被松开

鼠标事件：

onmousedown 鼠标按钮被按下

onmousemove 鼠标被移动

onmouseout 鼠标从某元素移开

onmouseover 鼠标移到某元素之上

onmouseup 鼠标按键被松开

# JS-事件



陆冬云 qq,wx: 184 3 185 83

鼠标事件：

onclick, onmouseover, onmousemove, onmouseout  
onmousedown, onmouseup  
ondrag(H5)\*\*\*

键盘事件：

onkeypress, onkeydown, onkeyup

时机事件：

onload , onsubmit, onselected

定时器事件：

setTimeout, setInterval



# 简单计算器的实现



目标：制作一个基本计算器

1. eval函数的使用
2. 液晶显示控件
3. 按键处理
4. 计算处理

参考test5\_calculator1 , test5\_calculator2

# 定时器的使用-settimeout



目标：制作一个自动计数器

1. 定时
2. 溢出处理

参考test6\_count

# 定时器的使用-settimeout



目标：漂浮动画

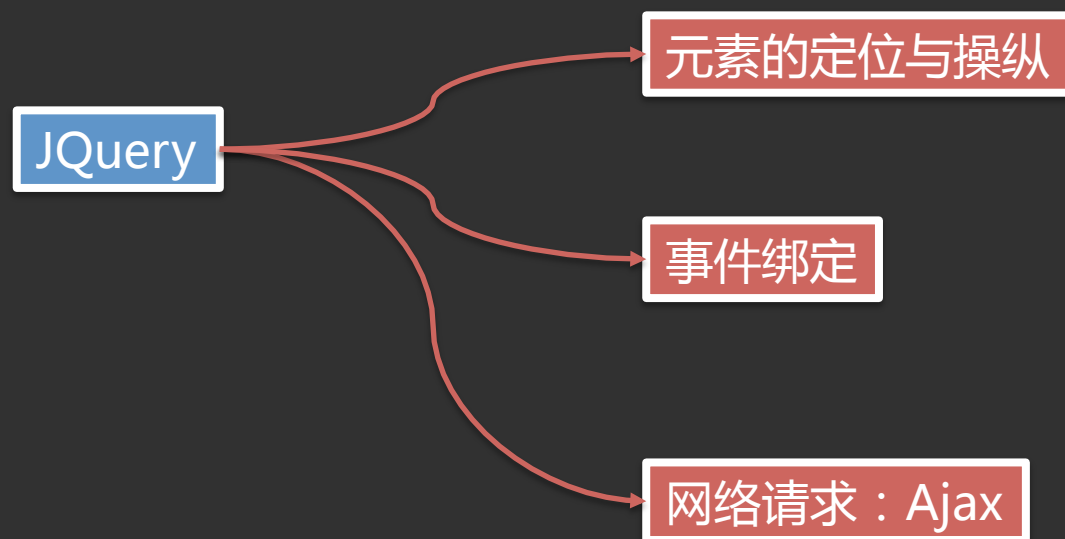
1. 定时
2. 浮动的位置计算与处理

参考test6\_settime\_layer

# JS之Jquery编程



陆冬云 qq,wx: 184 3 185 83



# Jquery-定位元素



与CSS的表达方式相似：

`$( "xxx" )`：通过tag定位元素

`$( "#xxx" )`：通过id定位元素

`$( ".xxx" )`：定位class定位元素

Jquery还有很多复杂的定位能力，不详述

# Jquery-通用功能



操纵内容：`$("#xx").html`

操纵css属性：`$("#xx").css`

操纵对象的css类：`$("#xx").class`

操纵对象的属性：`$("#xx").attr`

# 联系：操纵内容



目标：获取内容，修改内容

1. 修改内容
2. 追加内容
3. 在前面添加内容

参考[\*test1\\_html.html\*](#)

# 操纵css风格



目标：获取对象，修改其风格

1. 直接修改css属性
2. 通过修改元素属性，改变class

参考[test2\\_css.html](#)



# Jquery-功能的友好封装



陆冬云 qq,wx: 184 3 185 83

`$( "#xx" ).hide` : 隐藏元素

`$( "#xx" ).show` : 显示元素

`$( "#xx" ).toggle` : 切换显示状态

# 显示隐藏



目标：隐藏元素

1. 观察元素隐藏后的占位情况
2. 比较display:none 与invisible的区别

参考[test3\\_show.html](#)

# Jquery-属性动画



陆冬云 qq,wx: 184 3 185 83

```
$("#id1").animate({width: 500}, "slow");
```

```
$("#id1").animate({height: 500}, "slow");
```

```
$( "#xx" ).animate({"height":"300",  
"width":"500", "font-size":"80px"}, 2000);
```

# 练习：使用属性动画



目标：控制动画效果

1. 动画的顺序
2. 并发动画

参考[\*test4\\_animation.html\*](#)

# Jquery-事件绑定

方法1：直接按事件类型定义

```
$( "#id1" ).click(function(){...});
```

方法2：统一定义任意事件

```
$( "#id1" ).bind( "click" ,function(){...});
```

# 使用动态绑定添加控件响应



目标：操纵元素的响应能力

1. 观察什么元素可以绑定事件
2. 观察绑定前后的元素能力

参考[\*test5\\_event.html\*](#)

# 使用动态绑定添加控件响应



目标：改元素添加鼠标滑动响应能力

1. 使用hover , mouse事件绑定处理
2. 观察元素的效果随鼠标移动的变化

参考[test6\\_hover.html](#)

# Jquery-事件发生时的定位



- 1、\$(this)的含义
- 2、动态添加元素
- 3、为动态元素绑定事件



# 制作一个搜索条件选择



目标：建议标签式检索条件

1. 动态添加与消除元素
2. 动态绑定事件
3. 元素的清除

参考[test9\\_search.html](#)



陆冬云 qq,wx: 184 3 185 83

谢谢！