目錄

课程介绍

前端开发系统化学习教程，前端开发是后端程序员必修的课程，本课程开展注重两点：

1、实际开发中要用到的知识

2、面试中要用到的知识

课程介绍

介绍html文档的基本结构，html常用标签的使用，理解html语言制作网页基本原理，理解css的基本语法，css的引入方式，css选择器，css基本属性的使用等等。

html概述及**html**文档基本结构

html概述

HTML是 HyperText Mark-up Language 的首字母简写，意思是超文本标记语言，超文本指的是超链接，标记指的是标签，是一种用来制作网页的语言，这种语言由一个个的标签组成，用这种语言制作的文件保存的是一个文本文件，文件的扩展名为html或者htm。

html文档基本结构

一个html的基本结构如下：

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>网页标题</title>
    </head>
    <body>
          网页显示内容
    </body>
</html>
```

第一行是文档声明，第二行“<html>”标签和最后一行“</html>”定义html文档的整体，“<head>”标签和“<body>”标签是它的第一层子元素，“<head>”标签里面负责对网页进行一些设置以及定义标题，设置包括定义网页的编码格式，外链css样式文件和javascript文件等，设置的内容不会显示在网页上，标题的内容会显示在标题栏，“<body>”内编写网页上显示的内容。

一个html文件就是一个网页，html文件用编辑器打开显示的是文本，可以用文本的方式编辑它，如果用浏览器打开，浏览器会按照标签描述内容将文件渲染成网页。

html文档快速创建

新建一个html文档后，可以用快捷键的方式快速创建html文档。快捷键：!+tab键，或者 html:5+tab键

html标签入门

标签语法：

学习html语言就是学习标签的用法，html常用的标签有20多个，学会这些标签的使用，就基本上学会了HTML的使用。

标签的的使用方法：

```
<!-- 1、成对出现的标签：-->

<h1>h1标题</h1>
<div>这是一个div标签</div>
<p>这个一个段落标签</p>


<!-- 2、单个出现的标签： -->
<br>
<img src="images/pic.jpg" alt="图片">

<!-- 3、带属性的标签，如src、alt 和 href等都是属性 -->
<img src="images/pic.jpg" alt="图片">
<a href="http://www.baidu.com">百度网</a>

<!-- 4、标签的嵌套 -->
<div>
    <img src="images/pic.jpg" alt="图片">
    <a href="http://www.baidu.com">百度网</a>
</div>
```

块元素标签(行元素)和内联元素标签(行内元素)

标签在页面上会显示成一个方块。除了显示成方块，它们一般分为下面两类：
块元素：在布局中默认会独占一行，块元素后的元素需换行排列，块元素默认宽度等于父元素的宽度，即使设置了很小宽度，也占用一行。
内联元素：元素之间可以排列在一行，设置宽高无效，它的宽高由内容撑开，内联元素之间默认会有小间距。

常用块元素标签

1、标题标签，表示文档的标题，除了具有块元素基本特性外，还含有默认的外边距和字体大小

```
<h1>一级标题</h1>
<h2>二级标题</h2>
<h3>三级标题</h3>
<h4>四级标题</h4>
<h5>五级标题</h5>
<h6>六级标题</h6>
```

2、段落标签，表示文档中的一个文字段落，除了具有块元素基本特性外，还含有默认的外边距

```
<p>本人叫张山，毕业于某大学计算机科学与技术专业，今年23岁，本人性格开朗、
稳重、待人真诚、热情。有较强的组织能力和团队协作精神，良好的沟通能力和社
交能力，善于处理各种人际关系。能迅速适应环境，并融入其中。</p>
<p>本人热爱研究技术，热爱编程，希望能在努力为企业服务的过程中实现自身价值。</p>
```

3、通用块容器标签，表示文档中一块内容，具有块元素基本特性，没有其他默认样式

```
<div>这是一个div元素</div>
<div>这是第二个div元素</div>
<div>
    <h3>自我介绍</h3>
    <p>本人叫张山，毕业于某大学计算机科学与技术专业，今年23岁，本人性格开朗、
稳重、待人真诚、热情。有较强的组织能力和团队协作精神，良好的沟通能力和社
交能力，善于处理各种人际关系。能迅速适应环境，并融入其中。</p>
</div>
```

常用内联元素标签

1、超链接标签，链接到另外一个网页，具有内联元素基本特性，默认文字蓝色，有下划线

```
<a href="02.html">第二个网页</a>
<a href="http://www.baidu.com">百度网</a>
<a href="http://www.baidu.com"><img src="images/logo.png" alt="logo"></a>
```

2、通用内联容器标签，具有内联元素基本特性，没有其他默认样式

```
<p>这是一个段落文字，段落文字中有<span>特殊标志或样式</span>的文字</p>
```

3、图片标签，在网页中插入图片，具有内联元素基本特性，但是它支持宽高设置。

```
<img src="images/pic.jpg" alt="图片" />
```

其他常用功能标签

1、换行标签

```
<p>这是一行文字，<br>这是一行文字</p>
```

2、**html**注释：

html文档代码中可以插入注释，注释是对代码的说明和解释，注释的内容不会显示在页面上，html代码中插入注释的方法是：

```
<!-- 这是一段注释  -->
```

常用**html**字符实体

代码中成段的文字，如果文字间想空多个空格，在代码中空多个空格，在渲染成网页时只会显示一个空格，如果想显示多个空格，可以使用空格的字符实体,代码如下：

```
<!--  在段落前想缩进两个文字的空格，使用空格的字符实体：   -->
<p>
  一个html文件就是一个网页，html文件用编辑器打开显示的是文本，可以用<br />
文本的方式编辑它，如果用浏览器打开，浏览器会按照标签描述内容将文件<br />
渲染成网页，显示的网页可以从一个网页链接跳转到另外一个网页。</p>
```

在网页上显示 “<” 和 “>” 会误认为是标签，想在网页上显示“<”和“>”可以使用它们的字符实体，比如：

```
<!-- “<” 和 “>” 的字符实体为 &lt; 和 &gt;  -->
<p>
    &lt;div&gt;是一个html的一个标签<br>
    3 &lt; 5 <br>
    10 &gt; 5
</p>
```

html布局初步

网页布局原理

标签在网页中会显示成一个个的方块，先按照行的方式，把网页划分成多个行，再到行里面划分列，也就是在表示行的标签中再嵌套标签来表示列，标签的嵌套产生叠加效果。

布局示例

根据网页布局的原理以及上面的实例，写出网页的html结构代码。

标签语义化

在布局中需要尽量使用带语义的标签，使用带语义的标签的目的首先是为了让搜索引擎能更好地理解网页的结构，提高网站在搜索中的排名(也叫做SEO)，其次是方便代码的阅读和维护。

带语义的标签

1、h1~h6：表示标题

2、p：表示段落

3、img：表示图片

4、a：表示链接

不带语义的标签

1、div：表示一块内容

2、span：表示行内的一块内容

所以我们要根据网页上显示的内容，使用适合的标签，可以优化之前的代码。

css介绍

css概述

为了让网页元素的样式更加丰富，也为了让网页的内容和样式能拆分开，CSS由此思想而诞生，CSS是 Cascading Style Sheets 的首字母缩写，意思是层叠样式表。有了CSS，html中大部分表现样式的标签就废弃不用了，html只负责文档的结构和内容，表现形式完全交给CSS，html文档变得更加简洁。

css基本语法

css的定义方法是：

选择器 { 属性：值； 属性：值； 属性：值；}

选择器是将样式和页面元素关联起来的名称，属性是希望设置的样式属性，每个属性有一个或多个值。属性和值之间用冒号，一个属性和值与下一个属性和值之间用分号，最后一个分号可以省略，代码示例：

```
div{
    width:100px;
    height:100px;
    background:gold;
}
```

css引入方式

css引入页面的方式有三种：

1、内联式：通过标签的style属性，在标签上直接写样式。

```
<div style="width:100px; height:100px; background:red ">......</div>
```

2、嵌入式：通过style标签，在网页上创建嵌入的样式表。

```
<style type="text/css">
    div{ width:100px; height:100px; background:red }
    ......
</style>
```

3、外链式：通过link标签，链接外部样式文件到页面中。

```
<link rel="stylesheet" type="text/css" href="css/main.css">
```

css常用选择器一

1、标签选择器

标签选择器，此种选择器影响范围大，一般用来做一些通用设置，或用在层级选择器中。
举例：

```
div{color:red}
......
<div>这是第一个div</div>   <!-- 对应以上样式 -->
<div>这是第二个div</div>   <!-- 对应以上样式 -->
```

2、类选择器

通过类名来选择元素，一个类可应用于多个元素，一个元素上也可以使用多个类，应用灵活，可复用，是css中应用最多的一种选择器。
举例：

```
.blue{color:blue}
.big{font-size:20px}
.box{width:100px;height:100px;background:gold}
......
<div class="blue">....</div>
<h3 class="blue big box">....</h3>
<p class="blue box">....</p>
```

3、层级选择器

主要应用在标签嵌套的结构中，层级选择器，是结合上面的两种选择器来写的选择器,它可与标签选择器结合使用，减少命名，同时也可以通过层级，限制样式的作用范围。
举例：

```
.con{width:300px;height:80px;background:green}
.con span{color:red}
.con .pink{color:pink}
.con .gold{color:gold}
......
<div class="con">
    <span>....</span>
    <a href="#" class="pink">....</a>
    <a href="#" class="gold">...</a>
</div>
<span>....</span>
<a href="#" class="pink">....</a>
```

css属性入门

布局常用样式属性：

- width 设置元素(标签)的宽度，如：width:100px;
- height 设置元素(标签)的高度，如：height:200px;
- background 设置元素背景色或者背景图片，如：background:gold; 设置元素背景色为金色
- border 设置元素四周的边框，如：border:1px solid black; 设置元素四周边框是1像素宽的黑色实线

以上也可以拆分成四个边的写法，分别设置四个边的：

- border-top 设置顶边边框，如：border-top:10px solid red;
- border-left 设置左边边框，如：border-left:10px solid blue;
- border-right 设置右边边框，如：border-right:10px solid green;
- border-bottom 设置底边边框，如：border-bottom:10px solid pink;
- padding 设置元素包含的内容和元素边框的距离，也叫内边距，如padding:20px;padding是同时设置4个边的，也可以像border一样拆分成分别设置四个边:padding-top、padding-left、padding-right、padding-bottom。
- margin 设置元素和外界的距离，也叫外边距，如margin:20px;margin是同时设置4个边的，也可以像border一样拆分成分别设置四个边:margin-top、margin-left、margin-right、margin-bottom。
- float 设置元素浮动，浮动可以让块元素排列在一行，浮动分为左浮动：float:left; 右浮动：float:right;

文本常用样式属性一：

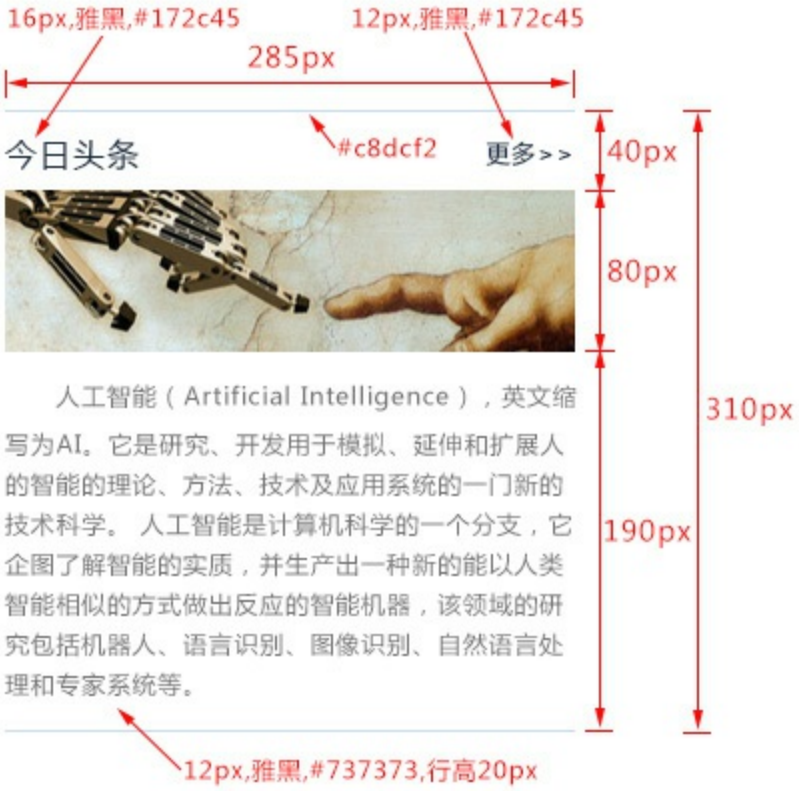
- color 设置文字的颜色，如： color:red;
- font-size 设置文字的大小，如：font-size:12px;
- font-family 设置文字的字体，如：font-family:'微软雅黑';为了避免中文字不兼容，一般写成：font-family:'Microsoft Yahei';
- font-weight 设置文字是否加粗，如：font-weight:bold; 设置加粗 font-weight:normal 设置不加粗
- line-height 设置文字的行高，如：line-height:24px; 表示文字高度加上文字上下的间距是24px，也就是每一行占有的高度是24px
- text-decoration 设置文字的下划线，如：text-decoration:none; 将文字下划线去掉

样式中的注释

```
/* 设置头部的样式 */
.header{
    width:960px;
    height:80px;
    background:gold;
}
```

css布局演示

通过样式，并且参照下图，可以把之前写的布局做进一步的调整，完成最终的布局效果：



html和**css**进阶

本节讲述路径相关知识、css属性进阶及选择器进阶、css盒子模型相关样式、盒模型使用技巧及相关问题、以及Photoshop辅助测量和辅助取色。

相对地址与绝对地址

网页上引入或链接到外部文件，需要定义文件的地址，常见引入或链接外部文件包括以下几种：

```
<!-- 引入外部图片   -->
<img src="images/001.jpg" alt="图片" />

<!-- 链接到另外一个网页   -->
<a href="002.html">链接到网页2</a>

<!-- 外链一个css文件   -->
<link rel="stylesheet" type="text/css" href="css/main.css" />

<!-- 外链一个js文件   -->
<script type="text/javascript" src="js/jquery.js"></script>
```

这些地址分为相对地址和绝对地址:

相对地址
相对于引用文件本身去定位被引用的文件地址，以上的例子都是相对地址，相对地址的定义技巧：

- “ ./ ” 表示当前文件所在目录下，比如：“./pic.jpg” 表示当前目录下的pic.jpg的图片，这个使用时可以省略。
- “ ../ ” 表示当前文件所在目录下的上一级目录，比如：“../images/pic.jpg” 表示当前目录下的上一级目录下的images文件夹中的pic.jpg的图片。

绝对地址
相对于磁盘的位置去定位文件的地址，比如： 绝对地址在整体文件迁移时会因为磁盘和顶层目录的改变而找不到文件，相对地址就没有这个问题。

html标签提高

无序列表标签

无序列表一般应用在布局中的新闻标题列表和文章标题列表，它是含有语义的，标签结构如下：

```
<ul>
    <li>列表标题一</li>
    <li>列表标题二</li>
    <li>列表标题三</li>
</ul>
```

列表的内容一般是可以链接的，点击链接到新闻或者文章的具体内容，所以具体结构一般是这样的：

```
<ul>
    <li><a href="#">列表标题一</a></li>
    <li><a href="#">列表标题二</a></li>
    <li><a href="#">列表标题三</a></li>
</ul>
```

css常用选择器二

1、id选择器

通过id名来选择元素，元素的id名称不能重复，所以一个样式设置项只能对应于页面上一个元素，不能复用，id名一般给程序使用，所以不推荐使用id作为选择器。
举例：

```
#box{color:red}
......
<p id="box">这是一个段落标签</p>   <!-- 对应以上一条样式，其它元素不允许应用此样式 -->
<p>这是第二个段落标签</p> <!-- 无法应用以上样式，每个标签只能有唯一的id名 -->
<p>这是第三个段落标签</p> <!-- 无法应用以上样式，每个标签只能有唯一的id名  -->
```

2、组选择器

多个选择器，如果有同样的样式设置，可以使用组选择器。
举例：

```
.box1,.box2,.box3{width:100px;height:100px}
.box1{background:red}
.box2{background:pink}
.box2{background:gold}

<div class="box1">....</div>
<div class="box2">....</div>
<div class="box3">....</div>
```

3、伪类及伪元素选择器

常用的伪类选择器有hover，表示鼠标悬浮在元素上时的状态，伪元素选择器有before和after,它们可以通过样式在元素中插入内容。

```
.box1:hover{color:red}
.box2:before{content:'行首文字';}
.box3:after{content:'行尾文字';}


<div class="box1">....</div>
<div class="box2">....</div>
<div class="box3">....</div>
```

css属性提高

文本常用样式属性二：

- text-align 设置文字水平对齐方式，如text-align:center 设置文字水平居中
- text-indent 设置文字首行缩进，如：text-indent:24px; 设置文字首行缩进24px
- font-style 设置字体是否倾斜，如：font-style:'normal'; 设置不倾斜，font-style:'italic';设置文字倾斜
- font 同时设置文字的几个属性，写的顺序有兼容问题，建议按照如下顺序写： font：是否加粗 字号/行高 字体；如： font:normal 12px/36px '微软雅黑';

其他常用样式属性：

- list-style 设置无序列表中的小圆点，一般把它设为"无"，如：list-style:none

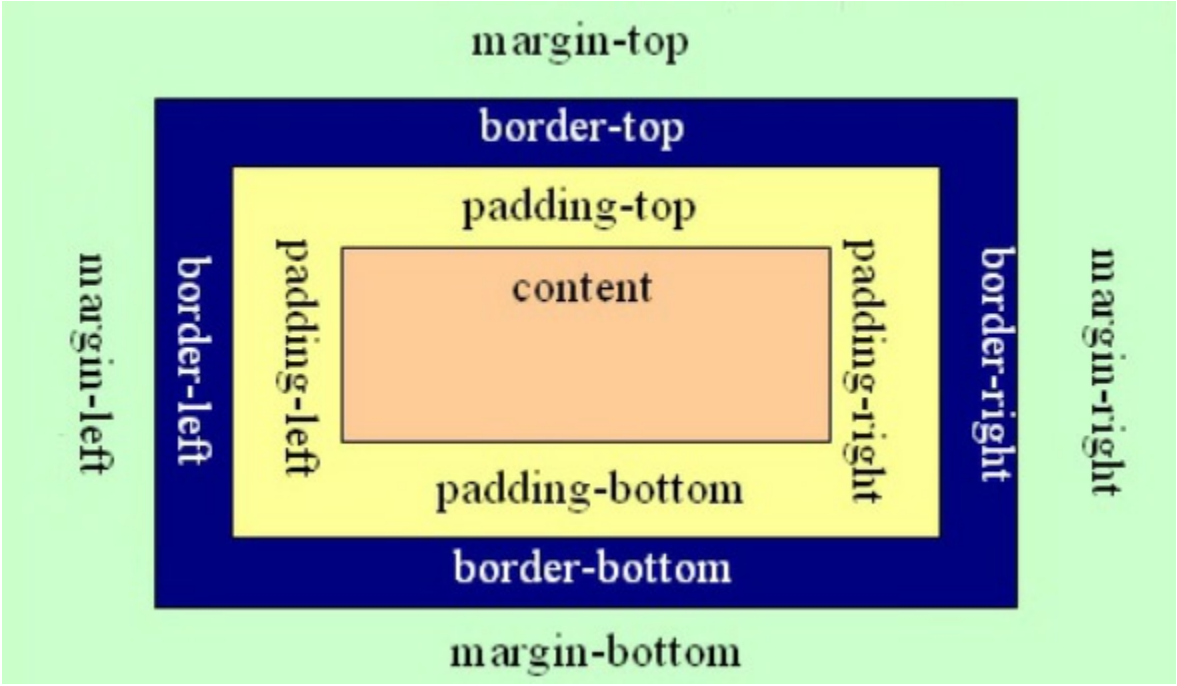
css元素溢出

当子元素的尺寸超过父元素的尺寸时，需要设置父元素显示溢出的子元素的方式，设置的方法是通过overflow属性来设置。

overflow的设置项：
1、visible 默认值。内容不会被修剪，会呈现在元素框之外。
2、hidden 内容会被修剪，并且其余内容是不可见的，此属性还有清除浮动、清除margin-top塌陷的功能。
3、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。
4、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。

CSS盒子模型

盒子模型解释
元素在页面中显示成一个方块，类似一个盒子，CSS盒子模型就是使用现实中盒子来做比喻，帮助我们设置元素对应的样式。盒子模型示意图如下：



把元素叫做盒子，设置对应的样式分别为：盒子的宽度(width)、盒子的高度(height)、盒子的边框(border)、盒子内的内容和边框之间的间距(padding)、盒子与盒子之间的间距(margin)。

设置宽高

```
width:200px;  /* 设置盒子的宽度，此宽度是指盒子内容的宽度，不是盒子整体宽度(难点) */
height:200px; /* 设置盒子的高度，此高度是指盒子内容的高度，不是盒子整体高度(难点) */
```

设置边框
设置一边的边框，比如顶部边框，可以按如下设置：

```
border-top:10px solid red;
```

其中10px表示线框的粗细；solid表示线性，常用的有：solid(实线) 、dashed(虚线)dotted(点线)；red表示线的颜色是红色。

设置其它三个边的方法和上面一样，把上面的'top'换成'left'就是设置左边，换成'right'就是设置右边，换成'bottom'就是设置底边。

四个边如果设置一样，可以将四个边的设置合并成一句：

```
border:10px solid red;
```

设置内间距**padding**

设置盒子四边的内间距，可设置如下：

```
padding-top：20px;     /* 设置顶部内间距20px */
padding-left:30px;     /* 设置左边内间距30px */
padding-right:40px;    /* 设置右边内间距40px */
padding-bottom:50px;   /* 设置底部内间距50px */
```

上面的设置可以简写如下：

```
padding：20px 40px 50px 30px; /* 四个值按照顺时针方向，分别设置的是 上 右 下 左
四个方向的内边距值。 */
```

padding后面还可以跟3个值，2个值和1个值，它们分别设置的项目如下：

```
padding：20px 40px 50px; /* 设置顶部内边距为20px，左右内边距为40px，底部内边距为50px */
padding：20px 40px; /* 设置上下内边距为20px，左右内边距为40px*/
padding：20px; /* 设置四边内边距为20px */
```

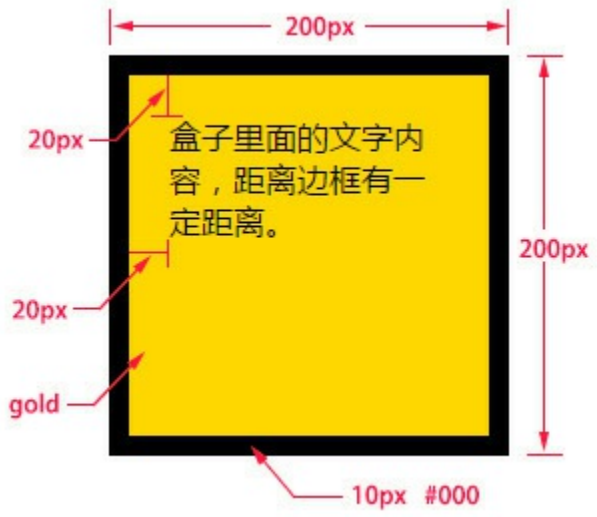
设置外间距**margin**

外边距的设置方法和padding的设置方法相同，将上面设置项中的'padding'换成'margin'就是外边距设置方法。

盒子的真实尺寸**(**难点**)**
盒子的width和height值固定时，如果盒子增加border和padding，盒子整体的尺寸会变大，所以盒子的真实尺寸为：

- 盒子宽度 = width + padding左右 + border左右
- 盒子高度 = height + padding上下 + border上下

理解练习
通过盒子模型的原理，制作下面的盒子：



chrome开发者工具
chrome开发者工具可以辅助开发，可以迅速查看元素的结构，样式，以及盒子模型结构和尺寸。

盒模型使用技巧及相关问题

margin相关技巧

1、设置不浮动的元素相对于父级水平居中： margin:x auto;

2、margin负值让元素位移及边框合并

垂直外边距合并

垂直外边距合并指的是，当两个不浮动的元素，它们的垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者，实际开发中一般只设置margin-top来避开这种合并的问题，或者将元素浮动，也可以避开这种问题。

margin-top 塌陷

在两个不浮动的盒子嵌套时候，内部的盒子设置的margin-top会加到外边的盒子上，导致内部的盒子margin-top设置失败，解决方法如下：

1、外部盒子设置一个边框

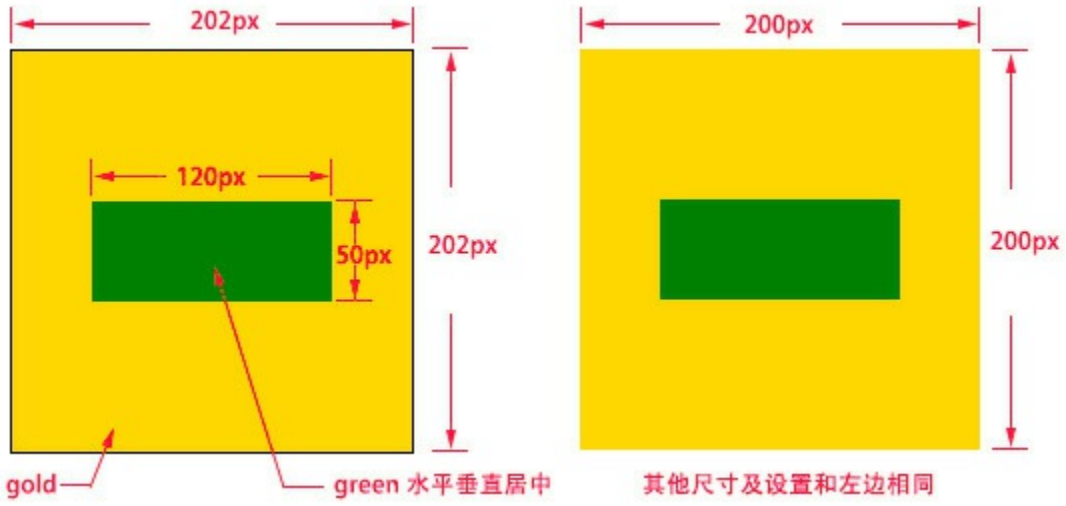
2、外部盒子设置 overflow:hidden

3、使用伪元素类：

```
.clearfix:before{
    content: '';
    display:table;
}
```

理解练习

使用margin间距制作下面的例子：



常用图片格式

图片是网页制作中很重要的素材，图片有不同的格式，每种格式都有自己的特性，了解这些特效，可以方便我们在制作网页时选取适合的图片格式，图片格式及特性如下：

1、psd

photoshop的专用格式。

优点：完整保存图像的信息，包括未压缩的图像数据、图层、透明等信息，方便图像的编辑。

缺点：应用范围窄，图片容量相对比较大。

2、jpg

网页制作及日常使用最普遍的图像格式。

优点：图像压缩效率高，图像容量相对最小。

缺点：有损压缩，图像会丢失数据而失真，不支持透明背景，不能制作成动画。

3、gif

制作网页小动画的常用图像格式。

优点：无损压缩，图像容量小、可以制作成动画、支持透明背景。

缺点：图像色彩范围最多只有256色，不能保存色彩丰富的图像，不支持半透明，透明图像边缘有锯齿。

4、png

网页制作及日常使用比较普遍的图像格式。

优点：无损压缩，图像容量小、支持透明背景和半透明色彩、透明图像的边缘光滑。

缺点：不能制作成动画

总结

在网页制作中，如何选择合适的图片格式呢？

1、使用大幅面图片时，如果要使用不透明背景的图片，就使用jpg图片；如果要使用透明或者半透明背景的图片，就使用png图片；

2、使用小幅面图片或者图标图片时，使用png图片；如果图片是动画，可以使用gif。

Photoshop辅助测量与取色

图片预览的方法
1、图片放缩
2、图片平移

尺寸测量方法
1、设置单位
2、矩形框测量、调整矩形框
3、文字大小的测量

取色方法
1、取色工具
2、前景色按钮

css颜色表示法
css颜色值主要有三种表示方法：
1、颜色名表示，比如：red 红色，gold 金色
2、rgb表示，比如：rgb(255,0,0)表示红色
3、16进制数值表示，比如：#ff0000 表示红色，这种可以简写成 #f00

html和**css**高级

本节讲述html表单的使用、元素类型总结及类型转换、以及浮动的定位的综合应用。

html表单

表单用于搜集不同类型的用户输入，表单由不同类型的标签组成，相关标签及属性用法如下：

1、<form>标签 定义整体的表单区域

- action属性 定义表单数据提交地址
- method属性 定义表单提交的方式，一般有“get”方式和“post”方式

2、<label>标签 为表单元素定义文字标注

3、<input>标签 定义通用的表单元素

- type属性
  - type="text" 定义单行文本输入框
  - type="password" 定义密码输入框
  - type="radio" 定义单选框
  - type="checkbox" 定义复选框
  - type="file" 定义上传文件
  - type="submit" 定义提交按钮
  - type="reset" 定义重置按钮
  - type="button" 定义一个普通按钮
- value属性 定义表单元素的值
- name属性 定义表单元素的名称，此名称是提交数据时的键名

4、<textarea>标签 定义多行文本输入框

5、<select>标签 定义下拉表单元素

6、<option>标签 与<select>标签配合，定义下拉表单元素中的选项

注册表单实例：

```
<form action="http://www..." method="get">
<p>
<label>姓名：</label><input type="text" name="username" />
</p>
<p>
<label>密码：</label><input type="password" name="password" />
</p>
<p>
<label>性别：</label>
<input type="radio" name="gender" value="0" /> 男
<input type="radio" name="gender" value="1" /> 女
</p>
<p>
<label>爱好：</label>
<input type="checkbox" name="like" value="sing" /> 唱歌
<input type="checkbox" name="like" value="run" /> 跑步
<input type="checkbox" name="like" value="swiming" /> 游泳
</p>
<p>
<label>照片：</label>
<input type="file" name="person_pic">
</p>
<p>
<label>个人描述：</label>
<textarea name="about"></textarea>
```

```
</p>
<p>
<label>籍贯：</label>
<select name="site">
    <option value="0">北京</option>
    <option value="1">上海</option>
    <option value="2">广州</option>
    <option value="3">深圳</option>
</select>
</p>
<p>
<input type="submit" name="" value="提交">
<input type="reset" name="" value="重置">
</p>
</form>
```

表单常用样式、属性及示例

- outline 设置input框获得焦点时，是否显示凸显的框线，一般设置为没有,比如：outline:none;
- placeholder 设置input输入框的默认提示文字。

表单布局实例

块元素类型及特性

块元素特性

块元素，也可以称为行元素，布局中常用的标签如：div、p、ul、li、h1~h6等等都是块元素，它在布局中的行为：

- 支持全部的样式
- 如果没有设置宽度，默认的宽度为父级宽度100%
- 盒子占据一行、即使设置了宽度

包含默认样式的块元素

上面讲的块标签中，有些标签是包含默认的样式的，这个含默认样式的有

- p标签：含有默认外边距
- ul：含有默认外边距和内边距，以及条目符号
- h1~h6标签：含有默认的外边距和字体大小
- body标签：含有默认的外边距

实际开发中，我们会把这些默认的样式在样式定义开头清除掉，清除掉这些默认样式，方便我们写自己的定义的样式，这种做法叫样式重置。

```
/* 清除标签默认的外边和内边距 */
body,p,h1,h2,h3,h4,h5,h6,ul{
    margin:0px;
    padding:0px;
}

/* 清除标签默认条目符号 */
ul{
    list-style:none;
}

/* 将h标签的文字大小设置为默认大小 */
h1,h2,h3,h4,h5,h6{
    font-size:100%;
    /* 根据实际需求来加   */
    font-weight:normal;
}
```

内联元素类型及特性

内联元素特性

内联元素，也可以称为行内元素，布局中常用的标签如：a、span等等都是内联元素，它们在布局中的行为：

- 不支持宽、高、margin上下、padding上下
- 宽高由内容决定
- 盒子并在一行
- 代码换行，盒子之间会产生间距
- 子元素是内联元素，父元素可以用text-align属性设置子元素水平对齐方式

解决内联元素间隙的方法

1、去掉内联元素之间的换行
2、将内联元素的父级设置font-size为0，内联元素自身再设置font-size

其他内联元素

1、 标签 行内元素，表示语气中的强调词
2、<i> 标签 行内元素，表示专业词汇
3、<b> 标签 行内元素，表示文档中的关键字或者产品名
4、 标签 行内元素，表示非常重要的内容

包含默认样式的内联元素

- a标签：含有的下划线以及文字颜色
- em、i标签：文字默认为斜体
- b、strong标签：文字默认加粗

实际开发中，对这些标签进行样式重置。

```
/* 去掉a标签默认的下划线 */
a{
    text-decoration:none;
}
/* 去掉标签默认的文字倾斜 */
em,i{
    font-style:normal;
}
/* 去掉标签默认的文字加粗(按实际需求) */
b,strong{
    font-weight:normal;
}
```

内联元素布局实例

内联块元素类型及特性、元素转换

内联块元素

内联块元素，也叫行内块元素，是新增的元素类型，现有元素没有归于此类别的，img和input元素的行为类似这种元素，但是也归类于内联元素，我们可以用display属性将块元素或者内联元素转化成这种元素。它们在布局中表现的行为：

- 支持全部样式
- 如果没有设置宽高，宽高由内容决定
- 盒子并在一行
- 代码换行，盒子会产生间距
- 子元素是内联块元素，父元素可以用text-align属性设置子元素水平对齐方式。

这三种元素，可以通过display属性来相互转化：

display属性

display属性是用来设置元素的类型及隐藏的，常用的属性有：
1、none 元素隐藏且不占位置
2、block 元素以块元素显示
3、inline 元素以内联元素显示
4、inline-block 元素以内联块元素显示

内联块元素布局实例

浮动

浮动特性

1、浮动元素有左浮动(float:left)和右浮动(float:right)两种

2、浮动的元素会向左或向右浮动，碰到父元素边界、其他元素才停下来

3、相邻浮动的块元素可以并在一行，超出父级宽度就换行

4、浮动让行内元素或块元素转化为有浮动特性的行内块元素(此时不会有行内块元素间隙问题)

5、父元素如果没有设置尺寸(一般是高度不设置)，父元素内整体浮动的子元素无法撑开父元素，父元素需要清除浮动

清除浮动

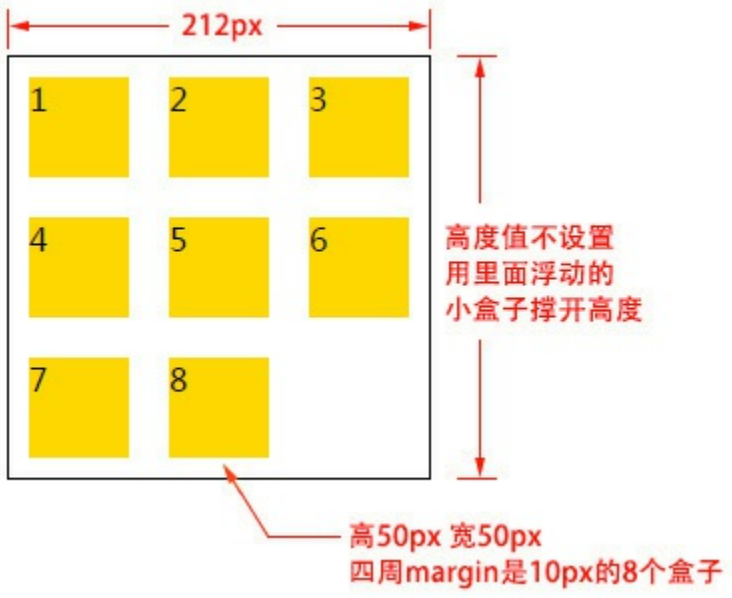
- 父级上增加属性overflow：hidden
- 在最后一个子元素的后面加一个空的div，给它样式属性 clear:both（不推荐）
- 使用成熟的清浮动样式类，clearfix

```
.clearfix:after,.clearfix:before{ content: "";display: table;}
.clearfix:after{ clear:both;}
.clearfix{zoom:1;}
```

清除浮动的使用方法：

```
.con2{... overflow:hidden}
或者
<div class="con2 clearfix">
```

理解练习
父级盒子不给高度，子集盒子浮动，父级盒子需要清除浮动



定位

文档流
文档流，是指盒子按照html标签编写的顺序依次从上到下，从左到右排列，块元素占一行，行内元素在一行之内从左到右排列，先写的先排列，后写的排在后面，每个盒子都占据自己的位置。

关于定位
我们可以使用css的position属性来设置元素的定位类型，postion的设置项如下：

- relative 生成相对定位元素，元素所占据的文档流的位置保留，元素本身相对自身原位置进行偏移。
- absolute 生成绝对定位元素，元素脱离文档流，不占据文档流的位置，可以理解为漂浮在文档流的上方，相对于上一个设置了定位的父级元素来进行定位，如果找不到，则相对于body元素进行定位。
- fixed 生成固定定位元素，元素脱离文档流，不占据文档流的位置，可以理解为漂浮在文档流的上方，相对于浏览器窗口进行定位。
- static 默认值，没有定位，元素出现在正常的文档流中，相当于取消定位属性或者不设置定位属性。

定位元素的偏移
定位的元素还需要用left、right、top或者bottom来设置相对于参照元素的偏移值。

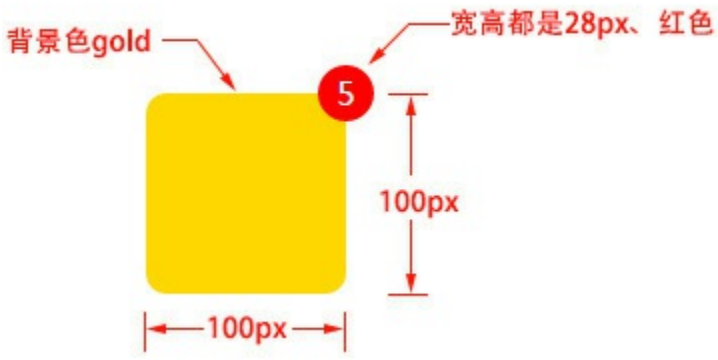
定位元素层级
定位元素是浮动的正常的文档流之上的，可以用z-index属性来设置元素的层级

伪代码如下:

```
.box01{
    ......
    position:absolute;  /* 设置了绝对定位 */
    left:200px;            /* 相对于参照元素左边向右偏移200px */
    top:100px;          /* 相对于参照元素顶部向下偏移100px */
    z-index:10          /* 将元素层级设置为10 */
}
```

定位元素特性
绝对定位和固定定位的块元素和行内元素会自动转化为行内块元素

理解练习
1、制作如下布局：



2、水平垂直居中的弹框

新增相关样式属性

```
/* 设置元素圆角,将元素四个角设置4px半径的圆角 */
border-radius:4px;

/* 设置元素透明度,将元素透明度设置为0.3，此属性需要加一个兼容IE的写法 */
opacity:0.3;
/* 兼容IE */
filter:alpha(opacity=30);
```

课程介绍

本节讲述background的综合设置和应用、css权重计算、表格的创建及样式设置、以及页面开发流程演示。

background属性

属性解释

background属性是css中应用比较多，且比较重要的一个属性，它是负责给盒子设置背景图片和背景颜色的，background是一个复合属性，它可以分解成如下几个设置项：

- background-image 设置背景图片地址
- background-position 设置背景图片的位置
- background-repeat 设置背景图片如何重复平铺
- background-color 设置背景颜色

可以将上面的属性设置用background属性合并成一句：

“background:url(bgimage.gif) left center no-repeat #00FF00”

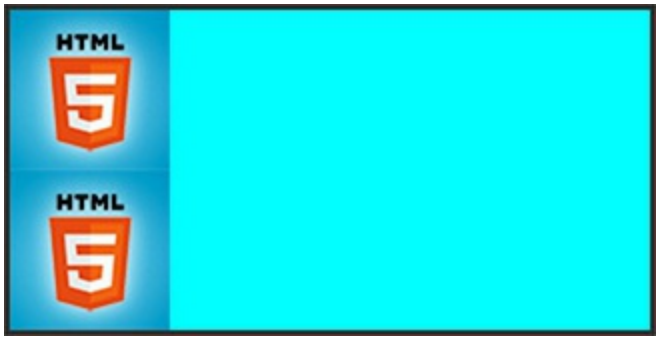
举例：

下面这些例子使用下面这张图片做为背景图：

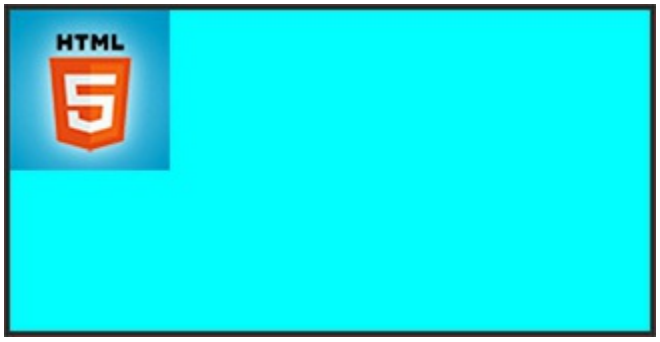
1、“background:url(bg.jpg)”，默认设置一个图片地址，图片会从盒子的左上角开始将盒子铺满。

2、“background:cyan url(bg.jpg) repeat-x”，横向平铺盒子，盒子其他部分显示背景颜色“cyan”。

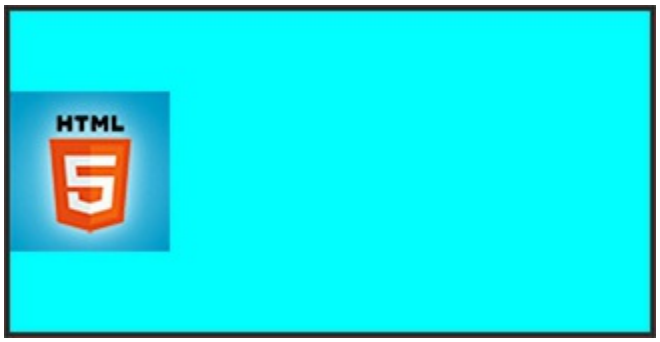
3、“background:cyan url(bg.jpg) repeat-y”，纵向平铺盒子，盒子其他部分显示背景颜色“cyan”。



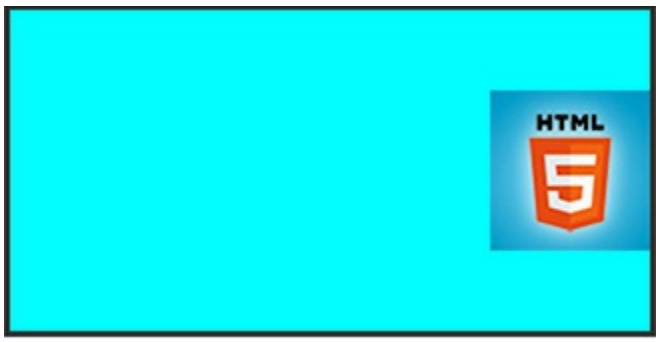
4、“background:cyan url(bg.jpg) no-repeat”，背景不重复，背景和盒子左上角对齐，盒子其他部分显示背景颜色“cyan”。



5、“background:cyan url(bg.jpg) no-repeat left center”，背景不重复，背景和盒子左中对齐，盒子其他部分显示背景颜色“cyan”。



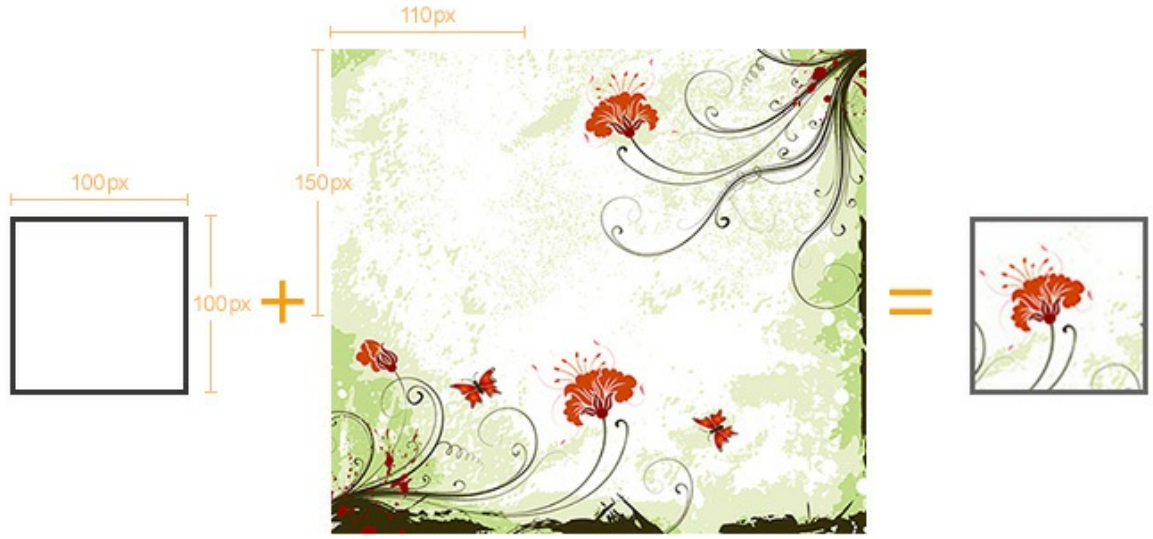
6、“background:cyan url(bg.jpg) no-repeat right center”，背景不重复，背景和盒子右中对齐，也就是背景图片的右边对齐盒子的右边，盒子其他部分显示背景颜色“cyan”。



例子说明：

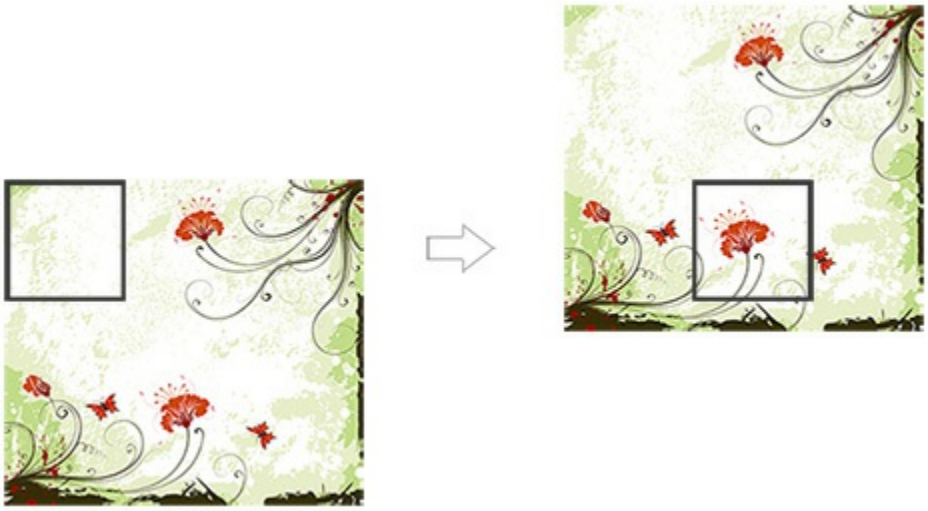
background-position的设置，可以在水平方向设置“left”、“center”、“right”，在垂直方向设置“top”、“center”、“bottom”，除了设置这些方位词之外，还可以设置具体的数值。

比如说，我们想把下边的盒子用右边的图片作为背景，并且让背景显示图片中靠近底部的那朵花：



用上面中间那张图片作为左边那个比它尺寸小的盒子的背景，上面右边的实现效果设置为：“background:url(location_bg.jpg) -110px -150px”，第一个数值表示背景图相对于自己的左上角向左偏移110px，负值向左，正值向右，第二个数值表示背景图相对于自己的左上角向上偏移150px，负值向上，正值向下。

实现原理示意图：



理解练习：
通过雪碧图制作如下布局：


1
美人鱼
2
疯狂动物城
61px
3
魔兽
4
美国队长3：英雄内战
5
湄公河行动
300px

CSS权重

CSS权重指的是样式的优先级，有两条或多条样式作用于一个元素，权重高的那条样式对元素起作用,权重相同的，后写的样式会覆盖前面写的样式。

权重的等级

可以把样式的应用方式分为几个等级，按照等级来计算权重

1、!important，加在样式属性值后，权重值为 10000
2、内联样式，如：style=""，权重值为1000
3、ID选择器，如：#content，权重值为100
4、类，伪类，如：.content、:hover 权重值为10
5、标签选择器，如：div、p 权重值为1

权重的计算实例

1、实例一：

```
<style type="text/css">
    div{
        color:red !important;
    }
</style>
......
<div style="color:blue">这是一个div元素</div>
<!--
两条样式同时作用一个div，上面的样式权重值为10000+1，下面的行间样式的权重值为1000，
所以文字的最终颜色为red
-->
```

2、实例二：

```
<style type="text/css">
    #content div.main_content h2{
        color:red;
    }
    #content .main_content h2{
        color:blue;
    }
</style>
......
<div id="content">
    <div class="main_content">
        <h2>这是一个h2标题</h2>
    </div>
</div>
<!--
第一条样式的权重计算： 100+1+10+1，结果为112；
第二条样式的权重计算： 100+10+1，结果为111；
h2标题的最终颜色为red
-->
```

表格元素及相关样式

1、<table>标签：声明一个表格

2、<tr>标签：定义表格中的一行

3、<td>和<th>标签：定义一行中的一个单元格，td代表普通单元格，th表示表头单元格，它们的常用属性如下：

- colspan 设置单元格水平合并，设置值是数值
- rowspan 设置单元格垂直合并，设置值是数值

表格制作练习：

基本情况				
姓　名		性　别		
名　族		出生日期		
政治面貌		健康情况		
籍　贯		学　历		
电子信箱		联系电话		

表格相关样式属性

- border-collapse 设置表格的边线合并，如：border-collapse:collapse;

流程顺序

1、创建项目目录
一般先创建一个总目录，然后在此目录中创建images、css、js三个目录，三个目录中分别放图片、css文件以及js文件。

2、切图
通过photoshop对网页效果图进行切图，所使用图片需要是带图层的psd格式。

3、制作雪碧图
将装饰类图片合并成一张图，然后删除分散的装饰类图片。

4、新建**html**文件，新建和编写**reset.css**

5、参照效果图，编写**html**和**css**代码

课程介绍

介绍javascript的页面引入方式、javascript变量、javascript函数的使用以及条件语句的基本使用，javascript获取元素及操作元素属性。

JavaScript介绍

JavaScript是运行在浏览器端的脚步语言，JavaScript主要解决的是前端与用户交互的问题，包括使用交互与数据交互。 JavaScript是浏览器解释执行的，前端脚本语言还有JScript（微软，IE独有），ActionScript(Adobe公司，需要插件)等。

前端三大块
1、HTML：页面结构
2、CSS：页面表现：元素大小、颜色、位置、隐藏或显示、部分动画效果
3、JavaScript：页面行为：部分动画效果、页面与用户的交互、页面功能

JavaScript嵌入页面的方式

1、行间事件（主要用于事件）

```
<input type="button" name="" onclick="alert('ok! ');">
```

2、页面script标签嵌入

```
<script type="text/javascript">
    alert('ok! ');
</script>
```

3、外部引入

```
<script type="text/javascript" src="js/index.js"></script>
```

变量、数据类型及基本语法规范

JavaScript 是一种弱类型语言，javascript的变量类型由它的值来决定。 定义变量需要用关键字 'var'

```
var iNum = 123;
var sTr = 'asd';

//同时定义多个变量可以用","隔开，公用一个‘var’关键字

var iNum = 45,sTr='qwe',sCount='68';
```

变量类型

5种基本数据类型：
1、number 数字类型
2、string 字符串类型
3、boolean 布尔类型 true 或 false
4、undefined undefined类型，变量声明未初始化，它的值就是undefined
5、null null类型，表示空对象，如果定义的变量将来准备保存对象，可以将变量初始化为null,在页面上获取不到对象，返回的值就是null

1种复合类型：
object

javascript语句与注释

1、javascript语句开始可缩进也可不缩进，缩进是为了方便代码阅读，一条javascript语句应该以“;”结尾;

```
<script type="text/javascript">
var iNum = 123;
var sTr = 'abc123';
function fnAlert(){
    alert(sTr);
};
fnAlert();
</script>
```

2、javascript注释

```
<script type="text/javascript">

// 单行注释
var iNum = 123;
/*
    多行注释
    1、...
    2、...
*/
var sTr = 'abc123';
</script>
```

变量、函数、属性、函数参数命名规范

1、区分大小写
2、第一个字符必须是字母、下划线（_）或者美元符号（$）
3、其他字符可以是字母、下划线、美元符或数字

匈牙利命名风格：

对象o Object 比如：oDiv
数组a Array 比如：aItems
字符串s String 比如：sUserName
整数i Integer 比如：iItemCount
布尔值b Boolean 比如：bIsComplete
浮点数f Float 比如：fPrice
函数fn Function 比如：fnHandler
正则表达式re RegExp 比如：reEmailCheck

函数

函数就是重复执行的代码片。

函数定义与执行

```
<script type="text/javascript">
    // 函数定义
    function fnAlert(){
        alert('hello!');
    }
    // 函数执行
    fnAlert();
</script>
```

变量与函数预解析
JavaScript解析过程分为两个阶段，先是编译阶段，然后执行阶段，在编译阶段会将function定义的函数提前，并且将var定义的变量声明提前，将它赋值为undefined。

```
<script type="text/javascript">
    fnAlert();       // 弹出 hello!
    alert(iNum);  // 弹出 undefined
    function fnAlert(){
        alert('hello!');
    }
    var iNum = 123;
</script>
```

函数传参 javascript的函数中可以传递参数，参数不能设置缺省值。

```
<script type="text/javascript">
    function fnAlert(a){
        alert(a);
    }
    fnAlert(12345);
</script>
```

函数**'return'**关键字
函数中'return'关键字的作用：
1、返回函数中的值或者对象
2、结束函数的运行

```
<script type="text/javascript">
function fnAdd(iNum01,iNum02){
    var iRs = iNum01 + iNum02;
    return iRs;
    alert('here!');
}

var iCount = fnAdd(3,4);
alert(iCount);  //弹出7
</script>
```

条件语句

通过条件来控制程序的走向，就需要用到条件语句。

条件运算符
==、===、>、>=、<、<=、!=、&&(而且)、||(或者)、!(否)

if else

```
var iNum01 = 3;
var iNum02 = 5;
var sTr;
if(iNum01>iNum02){
    sTr = '大于';
}
else
{
    sTr = '小于';
}
alert(sTr);
```

多重**if else**语句

```
var iNow = 1;
if(iNow==1)
{
    ... ;
}
else if(iNow==2)
{
    ... ;
}
else
{
    ... ;
}
```

获取元素方法

可以使用内置对象document上的getElementById方法来获取页面上设置了id属性的元素，获取到的是一个html对象，然后将它赋值给一个变量，比如：

```
<script type="text/javascript">
    var oDiv = document.getElementById('div1');
</script>
....
<div id="div1">这是一个div元素</div>
```

上面的语句，如果把javascript写在元素的上面，就会出错，因为页面上从上往下加载执行的，javascript去页面上获取元素div1的时候，元素div1还没有加载，解决方法有两种：

第一种方法：将javascript放到页面最下边

```
....
<div id="div1">这是一个div元素</div>
....

<script type="text/javascript">
    var oDiv = document.getElementById('div1');
</script>
</body>
```

第二种方法：将javascript语句放到window.onload触发的函数里面,获取元素的语句会在页面加载完后才执行，就不会出错了。

```
<script type="text/javascript">
    window.onload = function(){
        var oDiv = document.getElementById('div1');
    }
</script>

....

<div id="div1">这是一个div元素</div>
```

操作元素属性

获取的页面元素，就可以对页面元素的属性进行操作，属性的操作包括属性的读和写。

操作元素属性
var 变量 = 元素.属性名 读取属性
元素.属性名 = 新属性值 改写属性

属性名在**js**中的写法
1、html的属性和js里面属性写法一样
2、“class” 属性写成 “className”
3、“style” 属性里面的属性，有横杠的改成驼峰式，比如：“font-size”，改成”style.fontSize”

```
<script type="text/javascript">

    window.onload = function(){
        var oInput = document.getElementById('input1');
        var oA = document.getElementById('link1');
        // 读取属性值
        var sValue = oInput.value;
        var sType = oInput.type;
        var sName = oInput.name;
        var sLinks = oA.href;
        // 写(设置)属性
        oA.style.color = 'red';
        oA.style.fontSize = sValue;
    }

</script>

......

<input type="text" name="setsize" id="input1" value="20px">
<a href="http://www.itcast.cn" id="link1">传智播客</a>
```

innerHTML
innerHTML可以读取或者写入标签包裹的内容

```
<script type="text/javascript">
    window.onload = function(){
        var oDiv = document.getElementById('div1');
        //读取
        var sTxt = oDiv.innerHTML;
        alert(sTxt);
        //写入
        oDiv.innerHTML = '<a href="http://www.itcast.cn">传智播客<a/>';
    }
</script>

......

<div id="div1">这是一个div元素</div>
```

事件属性及匿名函数

事件属性

元素上除了有样式，id等属性外，还有事件属性，常用的事件属性有鼠标点击事件属性(onclick)，鼠标移入事件属性(mouseover),鼠标移出事件属性(mouseout),将函数名称赋值给元素事件属性，可以将事件和函数关联起来。

```
<script type="text/javascript">

window.onload = function(){
    var oBtn = document.getElementById('btn1');

    oBtn.onclick = myalert;

    function myalert(){
        alert('ok!');
    }
}

</script>
```

匿名函数

定义的函数可以不给名称，这个叫做匿名函数，可以将匿名函数的定义直接赋值给元素的事件属性来完成事件和函数的关联，这样可以减少函数命名，并且简化代码。函数如果做公共函数，就可以写成匿名函数的形式。

```
<script type="text/javascript">

window.onload = function(){
    var oBtn = document.getElementById('btn1');
    /*
    oBtn.onclick = myalert;
    function myalert(){
        alert('ok!');
    }
    */
    // 直接将匿名函数赋值给绑定的事件

    oBtn.onclick = function (){
        alert('ok!');
    }
}

</script>
```

综合实例

1、网页换肤

2、打印名片

课程介绍

介绍javascript数组和字符串的操作方法、循环语句、定时器的使用及实例、变量作用域、封闭函数的使用场景。

数组及操作方法

数组就是一组数据的集合，javascript中，数组里面的数据可以是不同类型的。

定义数组的方法

```
//对象的实例创建
var aList = new Array(1,2,3);

//直接量创建
var aList2 = [1,2,3,'asd'];
```

操作数组中数据的方法

1、获取数组的长度：aList.length;

```
var aList = [1,2,3,4];
alert(aList.length); // 弹出4
```

2、用下标操作数组的某个数据：aList[0];

```
var aList = [1,2,3,4];
alert(aList[0]); // 弹出1
```

3、join() 将数组成员通过一个分隔符合并成字符串

```
var aList = [1,2,3,4];
alert(aList.join('-')); // 弹出 1-2-3-4
```

4、push() 和 pop() 从数组最后增加成员或删除成员

```
var aList = [1,2,3,4];
aList.push(5);
alert(aList); //弹出1,2,3,4,5
aList.pop();
alert(aList); // 弹出1,2,3,4
```

5、reverse() 将数组反转

```
var aList = [1,2,3,4];
aList.reverse();
alert(aList);  // 弹出4,3,2,1
```

6、indexOf() 返回数组中元素第一次出现的索引值

```
var aList = [1,2,3,4,1,3,4];
alert(aList.indexOf(1));
```

7、splice() 在数组中增加或删除成员

```
var aList = [1,2,3,4];
aList.splice(2,1,7,8,9); //从第2个元素开始，删除1个元素，然后在此位置增加'7,8,9'三个元素
```

```
alert(aList); //弹出 1,2,7,8,9,4
```

多维数组
多维数组指的是数组的成员也是数组的数组。

```
var aList = [[1,2,3],['a','b','c']];

alert(aList[0][1]); //弹出2;
```

批量操作数组中的数据，需要用到循环语句

循环语句

程序中进行有规律的重复性操作，需要用到循环语句。

while循环

```
var sTr = '';
var i = 0;

while (i<5)
{
    sTr += "The number is " + i + "<br>";
    i++;
}

alert(sTr)
```

for循环

```
for(var i=0;i<len;i++)
{
    ......
}
```

课堂练习
1、数组去重

```
var aList = [1,2,3,4,4,3,2,1,2,3,4,5,6,5,5,3,3,4,2,1];

var aList2 = [];

for(var i=0;i<aList.length;i++)
{
    if(aList.indexOf(aList[i])==i)
    {
        aList2.push(aList[i]);
    }
}

alert(aList2);
```

2、将数组数据放入页面

字符串处理方法

1、字符串合并操作：“ + ”

```
var iNum01 = 12;
var iNum02 = 24;
var sNum03 = '12';
var sTr = 'abc';
alert(iNum01+iNum02);  //弹出36
alert(iNum01+sNum03);  //弹出1212 数字和字符串相加等同于字符串相加
alert(sNum03+sTr);     // 弹出12abc
```

2、parseInt() 将数字字符串转化为整数

```
var sNum01 = '12';
var sNum02 = '24';
var sNum03 = '12.32';
alert(sNum01+sNum02);  //弹出1224
alert(parseInt(sNum01)+parseInt(sNum02))  //弹出36
alert(parseInt(sNum03))   //弹出数字12 将字符串小数转化为数字整数
```

3、parseFloat() 将数字字符串转化为小数

```
var sNum03 = '12.32'
alert(parseFloat(sNum03));  //弹出 12.32 将字符串小数转化为数字小数
```

4、split() 把一个字符串分隔成字符串组成的数组

```
var sTr = '2017-4-22';
var aRr = sTr.split("-");
var aRr2= sTr.split("");

alert(aRr);  //弹出['2017','4','2']
alert(aRr2);  //弹出['2','0','1','7','-','4','-','2','2']
```

5、indexOf() 查找字符串是否含有某字符

```
var sTr = "abcdefgh";
var iNum = sTr.indexOf("c");
alert(iNum); //弹出2
```

6、substring() 截取字符串 用法： substring(start,end)（不包括end）

```
var sTr = "abcdefghijkl";
var sTr2 = sTr.substring(3,5);
var sTr3 = sTr.substring(1);

alert(sTr2); //弹出 de
alert(sTr3); //弹出 bcdefghijkl
```

字符串反转

```
var str = 'asdfj12jlsdkf098';
```

```
var str2 = str.split('').reverse().join('');

alert(str2);
```

调试程序的方法

1、alert

2、console.log

3、document.title

定时器

定时器在**javascript**中的作用

1、定时调用函数

2、制作动画

定时器类型及语法

```
/*
    定时器：
    setTimeout   只执行一次的定时器
    clearTimeout 关闭只执行一次的定时器
    setInterval   反复执行的定时器
    clearInterval 关闭反复执行的定时器

*/

var time1 = setTimeout(myalert,2000);
var time2 = setInterval(myalert,2000);
/*
clearTimeout(time1);
clearInterval(time2);
*/
function myalert(){
    alert('ok!');
}
```

课堂实例

1、定时器制作移动动画

2、定时器制作无缝滚动

变量作用域

变量作用域指的是变量的作用范围，javascript中的变量分为全局变量和局部变量。

1、全局变量：在函数之外定义的变量，为整个页面公用，函数内部外部都可以访问。
2、局部变量：在函数内部定义的变量，只能在定义该变量的函数内部访问，外部无法访问。

```
<script type="text/javascript">
    // 定义全局变量
    var a = 12;
    function myalert()
    {
        // 定义局部变量
        var b = 23;
        alert(a);
        // 修改全局变量
        a++;
        alert(b);
    }
    myalert(); // 弹出12和23
    alert(a);  // 弹出13
    alert(b);  // 出错
</script>
```

封闭函数

封闭函数是javascript中匿名函数的另外一种写法，创建一个一开始就执行而不用命名的函数。

一般定义的函数和执行函数：

```
function myalert(){
    alert('hello!');
};

myalert();
```

封闭函数：

```
(function(){
    alert('hello!');
})();
```

还可以在函数定义前加上“~”和“!”等符号来定义匿名函数

```
!function(){
    alert('hello!');
}()
```

封闭函数的作用
封闭函数可以创造一个独立的空间，在封闭函数内定义的变量和函数不会影响外部同名的函数和变量，可以避免命名冲突，在页面上引入多个js文件时，用这种方式添加js文件比较安全，比如：

```
var iNum01 = 12;
function myalert(){
    alert('hello!');
}

(function(){
    var iNum01 = 24;
    function myalert(){
        alert('hello!world');
    }
    alert(iNum01);
    myalert()
})()

alert(iNum01);
myalert();
```

课程介绍

介绍jquery的加载、jquery选择器、jquery的样式操作、jquery的click事件、jquery动画。

jquery介绍

jQuery是目前使用最广泛的javascript函数库。据统计，全世界排名前100万的网站，有46%使用jQuery，远远超过其他库。微软公司甚至把jQuery作为他们的官方库。

jQuery的版本分为1.x系列和2.x、3.x系列，1.x系列兼容低版本的浏览器，2.x、3.x系列放弃支持低版本浏览器，目前使用最多的是1.x系列的。

jquery是一个函数库，一个js文件，页面用script标签引入这个js文件就可以使用。

```
<script type="text/javascript" src="js/jquery-1.12.2.js"></script>
```

jquery的口号和愿望 Write Less, Do More（写得少，做得多）

1、http://jquery.com/ 官方网站
2、https://code.jquery.com/ 版本下载

jquery文档加载完再执行

将获取元素的语句写到页面头部，会因为元素还没有加载而出错，jquery提供了ready方法解决这个问题，它的速度比原生的 window.onload 更快。

```
<script type="text/javascript">

$(document).ready(function(){

     ......

});

</script>
```

可以简写为：

```
<script type="text/javascript">

$(function(){

     ......

});

</script>
```

jquery选择器

jquery用法思想一
选择某个网页元素，然后对它进行某种操作

jquery选择器
jquery选择器可以快速地选择元素，选择规则和css样式相同，使用length属性判断是否选择成功。

```
$('#myId') //选择id为myId的网页元素
$('.myClass') // 选择class为myClass的元素
$('li') //选择所有的li元素
$('#ul1 li span') //选择id为为ul1元素下的所有li下的span元素
$('input[name=first]') // 选择name属性等于first的input元素
```

对选择集进行过滤

```
$('div').has('p'); // 选择包含p元素的div元素
$('div').not('.myClass'); //选择class不等于myClass的div元素
$('div').eq(5); //选择第6个div元素
```

选择集转移

```
$('#box').prev(); //选择id是box的元素前面紧挨的同辈元素
$('#box').prevAll(); //选择id是box的元素之前所有的同辈元素
$('#box').next(); //选择id是box的元素后面紧挨的同辈元素
$('#box').nextAll(); //选择id是box的元素后面所有的同辈元素
$('#box').parent(); //选择id是box的元素的父元素
$('#box').children(); //选择id是box的元素的所有子元素
$('#box').siblings(); //选择id是box的元素的同级元素
$('#box').find('.myClass'); //选择id是box的元素内的class等于myClass的元素
```

判断是否选择到了元素
jquery有容错机制，即使没有找到元素，也不会出错，可以用length属性来判断是否找到了元素,length等于0，就是没选择到元素，length大于0，就是选择到了元素。

```
var $div1 = $('#div1');
var $div2 = $('#div2');
alert($div1.length); // 弹出1
alert($div2.length); // 弹出0
......
<div id="div1">这是一个div</div>
```

jquery样式操作

jquery用法思想二
同一个函数完成取值和赋值

操作行间样式

```
// 获取div的样式
$("div").css("width");
$("div").css("color");

//设置div的样式
$("div").css("width","30px");
$("div").css("height","30px");
$("div").css({fontSize:"30px",color:"red"});
```

特别注意
选择器获取的多个元素，获取信息获取的是第一个，比如：$("div").css("width")，获取的是第一个div的width。

操作样式类名

```
$("#div1").addClass("divClass2") //为id为div1的对象追加样式divClass2
$("#div1").removeClass("divClass")  //移除id为div1的对象的class名为divClass的样式
$("#div1").removeClass("divClass divClass2") //移除多个样式
$("#div1").toggleClass("anotherClass") //重复切换anotherClass样式
```

绑定**click**事件

给元素绑定click事件，可以用如下方法：

```
$('#btn1').click(function(){

    // 内部的this指的是原生对象

    // 使用jquery对象用 $(this)

})
```

获取元素的索引值
有时候需要获得匹配元素相对于其同胞元素的索引位置，此时可以用index()方法获取

```
var $li = $('.list li').eq(1);
alert($li.index()); // 弹出1
......
<ul class="list">
    <li>1</li>
    <li>2</li>
    <li>4</li>
    <li>5</li>
    <li>6</li>
</ul>
```

课堂练习

选项卡

jquery动画

通过animate方法可以设置元素某属性值上的动画，可以设置一个或多个属性值，动画执行完成后会执行一个函数。

```
/*
    animate参数：
    参数一：要改变的样式属性值，写成字典的形式
    参数二：动画持续的时间，单位为毫秒，一般不写单位
    参数三：动画曲线，默认为‘swing’，缓冲运动，还可以设置为‘linear’，匀速运动
    参数四：动画回调函数，动画完成后执行的匿名函数

*/

$('#div1').animate({
    width:300,
    height:300
},1000,'swing',function(){
    alert('done!');
});
```

课程介绍

介绍jquery特殊效果、链式调用、属性操作、jquery循环、jquery事件。

jquery特殊效果

```
fadeIn() 淡入

    $btn.click(function(){

        $('#div1').fadeIn(1000,'swing',function(){
            alert('done!');
        });

    });

fadeOut() 淡出
fadeToggle() 切换淡入淡出
hide() 隐藏元素
show() 显示元素
toggle() 切换元素的可见状态
slideDown() 向下展开
slideUp() 向上卷起
slideToggle() 依次展开或卷起某个元素
```

jquery链式调用

jquery对象的方法会在执行完后返回这个jquery对象，所有jquery对象的方法可以连起来写：

```
$('#div1') // id为div1的元素
.children('ul') //该元素下面的ul子元素
.slideDown('fast') //高度从零变到实际高度来显示ul元素
.parent()  //跳到ul的父元素，也就是id为div1的元素
.siblings()  //跳到div1元素平级的所有兄弟元素
.children('ul') //这些兄弟元素中的ul子元素
.slideUp('fast');  //高度实际高度变换到零来隐藏ul元素
```

课堂练习

层级菜单

jquery属性操作

1、html() 取出或设置html内容

```
// 取出html内容

var $htm = $('#div1').html();

// 设置html内容

$('#div1').html('<span>添加文字</span>');
```

2、prop() 取出或设置某个属性的值

```
// 取出图片的地址

var $src = $('#img1').prop('src');

// 设置图片的地址和alt属性

$('#img1').prop({src: "test.jpg", alt: "Test Image" });
```

课堂练习

聊天效果

jquery循环

对jquery选择的对象集合分别进行操作，需要用到jquery循环操作，此时可以用对象上的each方法：

```
$(function(){
    $('.list li').each(function(i){
        $(this).html(i);
    })
})
......
<ul class="list">
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
</ul>
```

课堂练习

手风琴效果

jquery事件

事件函数列表：

```
blur() 元素失去焦点
focus() 元素获得焦点
click() 鼠标单击
mouseover() 鼠标进入（进入子元素也触发）
mouseout() 鼠标离开（离开子元素也触发）
mouseenter() 鼠标进入（进入子元素不触发）
mouseleave() 鼠标离开（离开子元素不触发）
hover() 同时为mouseenter和mouseleave事件指定处理函数
ready() DOM加载完成
submit() 用户递交表单
```

课程介绍

介绍jquery事件冒泡、事件委托、元素节点操作，正则表达式及表单验证实例。

事件冒泡

什么是事件冒泡
在一个对象上触发某类事件（比如单击onclick事件），如果此对象定义了此事件的处理程序，那么此事件就会调用这个处理程序，如果没有定义此事件处理程序或者事件返回true，那么这个事件会向这个对象的父级对象传播，从里到外，直至它被处理（父级对象所有同类事件都将被激活），或者它到达了对象层次的最顶层，即document对象（有些浏览器是window）。

事件冒泡的作用
事件冒泡允许多个操作被集中处理（把事件处理器添加到一个父级元素上，避免把事件处理器添加到多个子级元素上），它还可以让你在对象层的不同级别捕获事件。

阻止事件冒泡
事件冒泡机制有时候是不需要的，需要阻止掉，通过 event.stopPropagation() 来阻止

```
$(function(){
    var $box1 = $('.father');
    var $box2 = $('.son');
    var $box3 = $('.grandson');
    $box1.click(function() {
        alert('father');
    });
    $box2.click(function() {
        alert('son');
    });
    $box3.click(function(event) {
        alert('grandson');
        event.stopPropagation();

    });
    $(document).click(function(event) {
        alert('grandfather');
    });
})

......

<div class="father">
    <div class="son">
        <div class="grandson"></div>
    </div>
</div>
```

阻止默认行为
阻止表单提交

```
$('#form1').submit(function(event){
    event.preventDefault();
})
```

合并阻止操作
实际开发中，一般把阻止冒泡和阻止默认行为合并起来写，合并写法可以用

```
// event.stopPropagation();
// event.preventDefault();
```

```
// 合并写法：
return false;
```

课堂练习
页面弹框（点击弹框外弹框关闭）；

事件委托

事件委托就是利用冒泡的原理，把事件加到父级上，通过判断事件来源的子集，执行相应的操作，事件委托首先可以极大减少事件绑定次数，提高性能；其次可以让新加入的子元素也可以拥有相同的操作。

一般绑定事件的写法

```
$(function(){
    $ali = $('#list li');
    $ali.click(function() {
        $(this).css({background:'red'});
    });
})
...
<ul id="list">
    <li>1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
    <li>5</li>
</ul>
```

事件委托的写法

```
$(function(){
    $list = $('#list');
    $list.delegate('li', 'click', function() {
        $(this).css({background:'red'});
    });
})
...
<ul id="list">
    <li>1</li>
    <li>2</li>
    <li>3</li>
    <li>4</li>
    <li>5</li>
</ul>
```

jquery元素节点操作

元素节点操作指的是改变html的标签结构，它有两种情况：
1、移动现有标签的位置
2、将新创建的标签插入到现有的标签中

创建新标签

```
var $div = $('<div>'); //创建一个空的div
var $div2 = $('<div>这是一个div元素</div>');
```

移动或者插入标签的方法
1、append()和appendTo()：在现存元素的内部，从后面放入元素

```
var $span = $('<span>这是一个span元素</span>');
$('#div1').append($span);
......
<div id="div1"></div>
```

2、prepend()和prependTo()：在现存元素的内部，从前面放入元素

3、after()和insertAfter()：在现存元素的外部，从后面放入元素

4、before()和insertBefore()：在现存元素的外部，从前面放入元素

删除标签

```
$('#div1').remove();
```

课堂练习
todolist(计划列表)实例

表单验证

1、什么是正则表达式：
能让计算机读懂的字符串匹配规则。

2、正则表达式的写法：
var re=new RegExp('规则', '可选参数');
var re=/规则/参数;

3、规则中的字符
1）普通字符匹配：
如：/a/ 匹配字符 ‘a’，/a,b/ 匹配字符 ‘a,b’

2）转义字符匹配：
\d 匹配一个数字，即0-9
\D 匹配一个非数字，即除了0-9
\w 匹配一个单词字符（字母、数字、下划线）
\W 匹配任何非单词字符。等价于[^A-Za-z0-9_]
\s 匹配一个空白符
\S 匹配一个非空白符
\b 匹配单词边界
\B 匹配非单词边界
. 匹配一个任意字符

```
var sTr01 = '123456asdf';
var re01 = /\d+/;
//匹配纯数字字符串
var re02 = /^\d+$/;
alert(re01.test(sTr01)); //弹出true
alert(re02.test(sTr01)); //弹出false
```

4、量词：对左边的匹配字符定义个数
? 出现零次或一次（最多出现一次）
+ 出现一次或多次（至少出现一次）
* 出现零次或多次（任意次）
{n} 出现n次
{n,m} 出现n到m次
{n,} 至少出现n次

5、任意一个或者范围
[abc123] : 匹配‘abc123’中的任意一个字符
[a-z0-9] : 匹配a到z或者0到9中的任意一个字符

6、限制开头结尾
^ 以紧挨的元素开头
$ 以紧挨的元素结尾

7、修饰参数：
g： global，全文搜索，默认搜索到第一个结果接停止
i： ingore case，忽略大小写，默认大小写敏感

8、常用函数

test

用法：正则.test(字符串) 匹配成功，就返回真，否则就返回假

正则默认规则

匹配成功就结束，不会继续匹配，区分大小写

常用正则规则

```
//用户名验证：(数字字母或下划线6到20位)
var reUser = /^\w{6,20}$/;

//邮箱验证：
var reMail = /^[a-z0-9][\w\.\-]*@[a-z0-9\-]+(\.[a-z]{2,5}){1,2}$/i;

//密码验证：
var rePass = /^[\w!@#$%^&*]{6,20}$/;

//手机号码验证：
var rePhone = /^1[34578]\d{9}$/;
```

课堂实例

注册页面表单验证

课程介绍

介绍幻灯片的制作、json数据格式及ajax。

课堂实例

幻灯片

json

json是 JavaScript Object Notation 的首字母缩写，单词的意思是javascript对象表示法，这里说的json指的是类似于javascript对象的一种数据格式，目前这种数据格式比较流行，逐渐替换掉了传统的xml数据格式。

javascript自定义对象：

```
var oMan = {
    name:'tom',
    age:16,
    talk:function(s){
        alert('我会说'+s);
    }
}
```

json格式的数据：

```
{
    "name":"tom",
    "age":18
}
```

与json对象不同的是，json数据格式的属性名称和字符串值需要用双引号引起来，用单引号或者不用引号会导致读取数据错误。

json的另外一个数据格式是数组，和javascript中的数组字面量相同。

```
["tom",18,"programmer"]
```

ajax与jsonp

ajax技术的目的是让javascript发送http请求，与后台通信，获取数据和信息。ajax技术的原理是实例化xmlhttp对象，使用此对象与后台通信。ajax通信的过程不会影响后续javascript的执行，从而实现异步。

同步和异步
现实生活中，同步指的是同时做几件事情，异步指的是做完一件事后再做另外一件事，程序中的同步和异步是把现实生活中的概念对调，也就是程序中的异步指的是现实生活中的同步，程序中的同步指的是现实生活中的异步。

局部刷新和无刷新
ajax可以实现局部刷新，也叫做无刷新，无刷新指的是整个页面不刷新，只是局部刷新，ajax可以自己发送http请求，不用通过浏览器的地址栏，所以页面整体不会刷新，ajax获取到后台数据，更新页面显示数据的部分，就做到了页面局部刷新。

同源策略
ajax请求的页面或资源只能是同一个域下面的资源，不能是其他域的资源，这是在设计ajax时基于安全的考虑。特征报错提示：

```
XMLHttpRequest cannot load https://www.baidu.com/. No
'Access-Control-Allow-Origin' header is present on the requested resource.
Origin 'null' is therefore not allowed access.
```

$.ajax使用方法
常用参数：
1、url 请求地址
2、type 请求方式，默认是'GET'，常用的还有'POST'
3、dataType 设置返回的数据格式，常用的是'json'格式，也可以设置为'html'
4、data 设置发送给服务器的数据
5、success 设置请求成功后的回调函数
6、error 设置请求失败后的回调函数
7、async 设置是否异步，默认值是'true'，表示异步

以前的写法：

```
$.ajax({
    url: 'js/data.json',
    type: 'GET',
    dataType: 'json',
    data:{'aa':1}
    success:function(data){
        alert(data.name);
    },
    error:function(){
        alert('服务器超时，请重试！');
    }
});
```

新的写法(推荐)：

```
$.ajax({
    url: 'js/data.json',
    type: 'GET',
    dataType: 'json',
```

```
    data:{'aa':1}
})
.done(function(data) {
    alert(data.name);
})
.fail(function() {
    alert('服务器超时，请重试！');
});

// data.json里面的数据：  {"name":"tom","age":18}
```

课堂练习
制作首页用户信息读取

jsonp

ajax只能请求同一个域下的数据或资源，有时候需要跨域请求数据，就需要用到jsonp技术，jsonp可以跨域请求数据，它的原理主要是利用了<script>标签可以跨域链接资源的特性。jsonp和ajax原理完全不一样，不过jquery将它们封装成同一个函数。

```
$.ajax({
    url:'js/data.js',
    type:'get',
    dataType:'jsonp',
    jsonpCallback:'fnBack'
})
.done(function(data){
    alert(data.name);
})
.fail(function() {
    alert('服务器超时，请重试！');
});

// data.js里面的数据：  fnBack({"name":"tom","age":18});
```

课堂实例
获取360搜索关键词联想数据

```
$(function(){
    $('#txt01').keyup(function(){
        var sVal = $(this).val();
        $.ajax({
            url:'https://sug.so.360.cn/suggest?',
            type:'get',
            dataType:'jsonp',
            data: {word: sVal}
        })
        .done(function(data){
            var aData = data.s;
            $('.list').empty();
            for(var i=0;i<aData.length;i++)
            {
                var $li = $('<li>'+ aData[i] +'</li>');
                $li.appendTo($('.list'));
            }
        })
    })
})

//......

<input type="text" name="" id="txt01">
```

```
<ul class="list"></ul>
```