

## 任务六

笔记本：任务六java

创建时间：2018/4/17 星期二 下午 6:55

更新时间：2018/4/25 星期三 下午 3:57

作者：18291893776@139.com

URL：<https://jingyan.baidu.com/article/59a015e36f0750f7948865e3.html>

## 第三章 随机点名和正则表达

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>随机点名</title>
6     <style type='text/css'>
7       #names{
8         width:200px;
9         height:200px;
10        border:2px solid #369;
11        margin:100px auto;
12        margin-bottom:0px;
13        text-align:center;
14        line-height:200px;
15        font-size:40px;
16      }
17    </style>
18  </head>
19  <body>
20    <div id="names">随机点名</div>
21    <br/>
22    <center>
23      <button onclick='start()'>开始</button>
24      <button onclick='end()'>停止</button>
25    </center>
26
27    <script type='text/javascript'>
28      //全局变量
29      //获取页面中的标签
30      var names=document.getElementById('names');
31      //定义数组
32      var arr=['马欢','侯杰','马娅','慕蓉','谢帖','李昱静','顺利','小军','田园红','张尧'];
33      //定时器 定义全局变量 方便清除
34      var init=0;
35
36      //封装随机点名函数
37      function start(){
38        //启动定时器
39        init=setInterval(function(){
40          //随机下标
41          var n=rand(0,arr.length-1);
42          //赋值文本
43          names.innerHTML=arr[n];
44        },100)
45      }
46
47      //停止点名
48      function end(){
49        clearInterval(init);
50      }
51
52      //封装随机函数
53      function rand(m,n){
54        return Math.floor(Math.random()*(n-m+1))+m;
55      }
56
57    </script>
58  </body>
59 </html>
```

## 2.html 随机点名 两个按钮

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>随机点名</title>
6     <style type='text/css'>
7       #names{
8         width:200px;
9         height:200px;
10        border:2px solid #369;
11        margin:100px auto;
12        margin-bottom:0px;
13        text-align:center;
14        line-height:200px;
15        font-size:40px;
16      }
17    </style>
18  </head>
19  <body>
20    <div id="names">随机点名</div>
21    <br/>
22    <center>
23      <button onclick='func()' id='btn'>开始</button>
24    </center>
25    <script type='text/javascript'>
26      //全局变量
27      //获取页面中的标签
28      var names=document.getElementById('names');
29      var btn=document.getElementById('btn');
30      //定义数组
31      var arr=['马欢','侯杰','马娅','慕蓉','谢帖','李显静','顺利','小军','田园红','张尧'];
32      //定时器 定义全局变量 方便清除
33      var init=0;
34      var isstart=true;
35
36      //封装方法
37      function func(){
38        //判断
39        if(isstart){
40          //启动点名
41          btn.innerHTML='结束';
42          isstart=false;
43          start();
44        }else{
45          //结束点名
46          btn.innerHTML='开始';
47          isstart=true
48          end();
49        }
50      }
51    </script>
52  </body>
53 </html>
```

```

55
56 //封装随机点名函数
57 function start(){
58     //启动定时器
59     init=setInterval(function(){
60         //随机下标
61         var n=rand(0,arr.length-1);
62         //赋值文本
63         names.innerHTML=arr[n];
64
65     },100)
66 }
67
68 //停止点名
69 function end(){
70     clearInterval(init);
71 }
72
73 //封装随机函数
74 function rand(m,n){
75     return Math.floor(Math.random()*(n-m+1))+m;
76 }
77
78 </script>
79 </body>
80 </html>

```

### 3.html 随机点名 一个按钮

```

1 <!DOCTYPE html>
2 <html lang="en">
3     <head>
4         <meta charset="UTF-8">
5         <title>js正则表达式</title>
6     </head>
7     <body>
8         <a href='./phone/one.py'>手机</a>
9         <script type='text/javascript'>
10             var str='\r\n $18291aa8937 7615349237aa468love';
11             //正则表达式
12             //var reg=new RegExp('Love');
13             //var reg=new RegExp('\w');//
14             转义字符需要再次转译，因此不常用，\w代表的就是字母数字或下划线
15             //推荐用法
16             //var reg=/\w/;//用//做边界符
17             //普通字符
18
19             //转义字符
20             //var reg=/\w/;//单个字母数字下划线
21             //var reg=/\W/;//单个非字母数字下划线
22             //var reg=/\d/;//单个数字
23             //var reg=/\D/;//单个非数字
24             //var reg=/\s/;//单个空白字符
25             //var reg=/\S/;//单个非空白字符
26             //转义字符

```

```

27 //元字符 . * + ? { } [ ] ( ) | ^ $
28 //var reg=/.//除了换行之外的任意单个字符
29 //var reg=/d*/;//匹配0次或多次
30 //var reg=/a*/;
31 //var reg=/a+//匹配一次或多次
32 //var reg=/a+?//禁止贪婪 找到1个就回来吧
33 //var reg=/a*?//直接门都没进，没找就回来了
34 //var reg=/d{2};//限制匹配的次数
35 //var reg=/w{2,5};//限制匹配次数2次到5次
36 //var reg=/[a-z,0-9,A-Z,_]//限制范围
37 var reg=/.*\d{4}w*/;
38 var reg=/.*(\d{4})w*/;//子组 首先小括号内的正则规则是作为整个规则来使用的，匹配完成后，单独将小括号的内容作为一个元素输出
39 var str="<a href='./phone/one.py'>手机</a>"
40 var reg=<a href='(.*)'>(.)</a>/
41
42 var str='18291893776@139.comcn';
43 var reg=/^w+@w+.(com|cn|org|net|gov)$/;//^限制开始 $限制结尾
44
45 两者搭配使用，直接输出要爬取的内容
46
47 //元字符
48 //方法有两种 test exec
49 var res1=reg.test(str);//返回布尔类型，如果存在返回true
50 var res2=reg.exec(str);//按照正则规则匹配，如果存在返回结果，[数组]，如果不存在返回null;
51 /*console.log(res2[1]);
52 console.log(res2[2]);*/
53 console.log(res1);
54 console.log(res2);
55 </script>
56 </body>
57 </html>

```

代表转义

表示或者

#### 4.html 正则表达式

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>正则表达式字符串方法</title>
6 </head>
7 <body>
8   <script type='text/javascript'>
9     var str='18291893776@139.com';
10    var reg=/^w+@w+.(com|cn|org|net|gov)$/
11    //字符串方法
12    var res3=str.match(reg);//把字符串进行正则的匹配，如果存在返回结果[数组]，不存在返回null
13    console.log(res3);
14
15
16   </script>
17
18 </body>
19 </html>

```

#### 5.html 正则表达式字符串方法

获取焦点事件指的是当光标进入form表单，在form表单之后会出现一段话

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>正则表达式在form表单中的小小应用</title>
6   </head>
7   <body>
8     <form id='my' action=''>
9       手机号: <input type='text' name='phone'><span id='r'></span><br/>
10      密码: <input type='password' name='password'><br/>
11      <input type='submit' value='注册'>
12    </form>
13
14
15    <script type='text/javascript'>
16      // 获取
17      var my=document.getElementById('my')
18      var r=document.getElementById('r')
19      // 定义一个全局变量, 辅助绑定的onsubmit事件
20      var userok=false;
21
22      // 绑定事件 onfocus 获取焦点事件
23      my.phone.onfocus=function(){
24        r.innerHTML='&nbsp;请输入11位电话号码';
25        r.style.color='green';
26      }
27
28      // 绑定丧失焦点事件
29      获取用户输入的值, 用正则方法验证是否符合电话号码特征, 符合则提示正确, 不符合则提示错误
30      my.phone.onblur=function(){
31        var v=my.phone.value;
32        var reg=/^1\d{10}$/;
33        if(reg.test(v)){
34          r.innerHTML='√';
35          r.style.color='blue';
36          userok=true;
37        }else{
38          r.innerHTML='x';
39          r.style.color='red';
40          userok=false;
41        }
42      }
43      // 绑定表单提交事件
44      my.onsubmit=function(){
45        // 判断是否正确
46        if (userok){
47          // 如果在事件绑定return false 就是阻止元素的默认行为
48          return false;
49        }
50      }
51    }
52  </script>
53 </body>
54 </html>

```

表示的是对象

表示焦点事件

## 6.html 正则表达式在form表单中的应用

## 第四章 jQuery

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>jQuery</title>
6 </head>
7 <body>
8   <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
9   <script type='text/javascript'>
10    alert($);
11
12   </script>
13 </body>
14 </html>

```

## 1.html 引入jQuery

百度特效词：

- 1.黑洞、人造黑洞、黑洞频率
- 2.大风、海边、打雷、布谷鸟
- 3.抖动、晃动、摇一摇、duang
- 4.翻转、反转
- 5.旋转、水平旋转

```

115 <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
116 <script type='text/javascript'>
117   //alert($);
118   //基础选择器
119   //id选择器 var Logo=document.getElementById('Logo');Logo.style.border='1px solid red';
120   //$('#Logo').css('border','1px solid red')
121   //console.log($('#Logo'))
122
123   //类选择器
124   //$('.w').css('border','1px solid red')
125
126   //标签选择器
127   //$('li').css('border','2px solid green')
128
129   //逗号 并列获取
130   //$('#Logo,#menu').css('border','1px solid yellow')
131
132   //空格 层级获取
133   //$('#images li').css('border','2px solid green')
134
135   //过滤获取
136   //获取页面中的第一个li标签
137   //$('li:first').css('background','#369')
138
139   //获取页面中的最后一个li标签

```

```

140 //$('#li:last').css('background','#369')
141
142 //通过索引获取元素
143 //$('#li:eq(1)').css('background','#369')
144
145 //获取包含指定文本的元素
146 //$('#li:contains(国)').css('background','#369')
147
148 //通过包含指定属性获取元素 通过属性来获取
149 //$('#li[name=y]').css('background','#369')
150
151 //获取所有的子元素 找的是直接的子元素
152 //$('#images').children().css('background','#369')
153
154 //获取第一个子元素
155 //$('#images li:first-child').css('background','#369');
156
157 //获取指定索引的子元素 索引从1开始
158 //$('#ul li:nth-child(9)').css('background','#369');
159
160 //获取元素的上一个同级元素
161 //$('#f').prev().css('background','#369')
162
163 //获取同辈元素（同辈元素不包含自己）。这个很重要，需要掌握
164 //$('#f').siblings().css('background','#369')
165
166 //获取父辈元素
167 //$('#f').parent().css('background','#369')
168
169 //获取先辈级元素 这样相当于祖宗十八代都翻出来了
170 //$('#f').parents().css('border','solid 1px red')
171 //因此一般都是找个别的祖先
172 //$('#f').parents('#all').css('border','solid 1px blue')
173
174 //在父级元素中查找指定的子元素
175 //$('#images').find('.w').css('background','#369')//方法1
176 $('#images .w').css('background','#369')//方法2
177
178 </script>
179
180
181 </body>
182 </html>

```

## bak.html jQuery选择器

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>元素的操作</title>
6     <style type="text/css">
7       .all{
8         width:200px;
9         height:200px;
10        border:1px solid red;
11      }
12
13    </style>
14  </head>
15  <body>
16    <div class='all'>ABC</div>
17    <div >ABC</div>
18    <div >ABC</div>
19    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
20    <script type='text/javascript'>
21      //alert($)
22      //样式操作 设置单个值 设置的是行内样式 优先级高
23      //$('#.all').css('background','cyan')
24      $('#.all').css('fontSize','30px')
25
26      //样式操作 设置多个值
27      //$('#.all').css({width:'200px',height:'200px',background:'green'})

```



```

28
29 //var w=$('#.all').css('width')//获取操作 获取操作的使用相对较少 常用来进行设置操作
    这种方法不仅可以获取样式操作中设置的值，即24行21行设置的值，还可以获取在第7行设置的值
30 //console.log(w)
31
32 //class类的操作
33 //添加一个class类
34 /*$('#div').eq(1).addClass('all');
35 //$('#div').eq(2).addClass('all');
36 $('#div').eq(1).addClass('a');//是又添加一个类 不是覆盖
37
38 //删除指定类
39 $('#div').eq(1).removeClass('a');
40 //这样是删除所有类
41 $('#div').eq(1).removeClass();*/
42
43 //切换 指的是原本没有的类 写了切换操作之后会存在 再写一遍又不存在 交替进行
44 //如果元素没有该类就是进行设置，如果有该类，就是移除
45 //$('#div').eq(2).toggleClass('all');
46 //$('#div').eq(2).toggleClass('all');
47
48 //attr 属性操作 设置元素的属性和值 也可以是重新赋值
49 //$('#div').eq(2).attr('class','abc')
50 //$('#div').eq(2).attr('id','ccc')
51 //$('#div').eq(2).attr('love','I Love U')
52 //获取
53 //var l=$('#div').eq(2).attr('love');
54 //console.log(l);
55
56
57 //文本操作 text() html()
58 //设置
59 //$('#div').eq(2).html('love');
60 $('#div').eq(2).html('<span style="color:red">love</span>');
61 //$('#div').eq(2).text('love');
62 //$('#div').eq(2).text('<span style="color:red">love</span>');//
    设置时不解析标签，只当做文本处理，所以不要用这种方法
63
64 //获取
65 var h=$('#div').eq(2).html();//获取时如果有子标签也能够获取
66
67 var t=$('#div').eq(2).text();//只能获取纯文本
68
69 console.log(h);
70 console.log(t);
71 </script>
72 </body>
73 </html>

```

## p2.html jQuery元素操作



```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>批量绑定事件</title>
6     <style type='text/css'>
7       div{
8         float:left;
9         margin:10px;
10        width:100px;
11        height:100px;
12        background:#369;
13      }
14    </style>
15  </head>
16  <body>
17    <div></div>
18    <div></div>
19    <div></div>
20    <div></div>
21    <script style='text/javascript' src='jquery-1.8.3.min.js'></script>
22    <script style='text/javascript'>
23      $('div').click(function(){
24        //代表的是批量 所有
25        //$('div').css('background','#f39');
26        //代表当前的这个元素，当前点击的标签 点谁谁变色
27        $(this).css('background','#f39');
28      })
29    </script>
30  </body>
31 </html>

```

## p5.html jQuery批量绑定事件与单次绑定事件

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6     <style type='text/css'>
7       body,div,ul,li{
8         padding:0px;
9         margin:0px;
10        list-style:none;
11      }
12      .all{
13        width:500px;
14        height:540px;
15        margin:100px;
16        border:1px solid #333;
17      }
18      .options li{
19        width:100px;
20        height:40px;
21        text-align:center;
22        line-height:40px;
23        float:left;
24        border-right:1px solid #333;
25        border-bottom:1px solid #333;
26        box-sizing:border-box;
27        background:#f5f5f5;

```

```

28         cursor:pointer;
29     }
30     .options li:last-child{
31         border-right:none;
32     }
33     .card li{
34         display:none;
35         height:500px;
36         text-align:center;
37         line-height:500px;
38     }
39     .options .active{
40         border-bottom:none;
41         background:#fff;
42     }
43     .card .active{
44         display:block;
45     }
46
47
48
49     </style>
50 </head>
51 <body>
52     <div class='all'>
53         <ul class='options'>
54             <li class='active'>全部订单</li>
55             <li>已完成订单</li>
56             <li>待付款订单</li>
57             <li>待收货订单</li>
58             <li style='border-right:none'>待评价订单</li>
59         </ul>
60         <ul class='card'>
61             <li class='active'>这是 全部订单</li>
62             <li>这是 已完成订单</li>
63             <li>这是 待付款订单</li>
64             <li>这是 待收货订单</li>
65             <li>这是 待评价订单</li>
66         </ul>
67         <script style='text/javascript' src='jquery-1.8.3.min.js'></script>
68         <script style='text/javascript'>
69
70             // 获取所有的选项，绑定单击事件
71             $('.options li').click(function(){
72                 // 给当前的这个元素添加active 找到同辈元素移出class
73
74                 $(this).addClass('active').siblings().removeClass('active')
75                 // 获取当前选项的索引
76                 var inx=$(this).index();
77                 $('.card li').eq(inx).addClass('active').siblings().removeClass('active')
78             })
79         </script>
80
81     </div>
82 </body>

```

## p6.html jQuery绑定单击事件

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>相关尺寸</title>
6     <style type='text/css'>
7       body{
8         margin:0px;
9         padding:0px;
10        height:3000px;
11        width:300px;
12
13
14      }
15      .all{
16        width:500px;
17        height:500px;
18        border:1px solid red;
19        position:absolute;
20        top:200px;
21        left:200px;
22      }
23      #move{
24        width:100px;
25        height:200px;
26        background:#369;
27        position:absolute;
28
29        top:200px;
30        left:100px;
31      }
32    </style>
33  </head>
34  <body>
35    <div class='all'>
36      <div id='move'></div>
37    </div>
38    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
39    <script type='text/javascript'>
40      //获取元素相对于文档的偏移量
41      /*var os=$('#move').offset();//返回的是对象
42      var os1=$('#move').offset().top;
43      var os2=$('#move').offset().left;
44      console.log(os);
45      console.log(os1);
46      console.log(os2);*/
47
48      //获取元素相对于父级元素的偏移量
49      /*var ps=$('#move').position();
50      var ps1=$('#move').position().top;
51      var ps2=$('#move').position().left;
52      console.log(ps)
53      console.log(ps1)
54      console.log(ps2)*/
```

```

55
56 // 获取文档的滚动距离
57 /*var st=$(document).scrollTop();
58 var sl=$(document).scrollLeft();
59 console.log(st,sl);*/
60
61 // 设置滚动距离
62 //$(window).scrollTop(400);
63
64
65 // 获取当前元素的宽高
66 /*var w=$('#move').width();
67 var h=$('#move').height();
68
69 // 当前元素的宽高既可以获取也可以设置
70 $('#move').width(250);
71
72 console.log(w,h)*/
73
74 // 获取当前可视区域的宽高
75 //console.log($(window).width());
76 //console.log($(window).height());
77
78 // 获取文档的宽高
79 console.log($(document).width());
80 console.log($(document).height())
81 </script>
82 </body>
83 </html>

```

### p3.html 相关尺寸的获取与设置

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>关于事件</title>
6     <style type='text/css'>
7       .item{
8         width:100px;
9         height:100px;
10        background:#f39;
11        float:left;
12        margin:10px;
13      }
14    </style>
15  </head>
16  <body>
17    <div class='item'>1</div>
18    <br/>
19    <a href='www.baidu.com'>百度一下</a>
20    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
21    <script type='text/javascript'>
22      // 获取元素 基本绑定方法 元素对象. 事件（事件处理程序）
23      // 绑定单击事件
24      /*$('#.item').click(function(){
25        alert('2')
26      })
27      // 绑定双击事件

```

```

28     $(' .item').dblclick(function(){
29         alert(8);
30     })* /
31
32     // 页面加载完毕事件
    即指的是将17行的写到37行，如果没有加载完毕事件的这个函数，就无法实现
33  /*$(function(){
34     // 获取元素
35     $(' .item').click(function(){
36         alert(1);
37     })
38 })* /
39
40     // bind 方法绑定事件
41     /*$(' .item').bind('click',function(){
42         alert(3);
43     })* /
44
45     // live 动态绑定 一般针对动态创建的元素进行事件的绑定 在高版本中移出了这个方法
46     /*$(' .item').live('click',function(){
47         alert(4);
48     })
49     // 创建元素
50     var newdiv=$('<div class="item">2</div>')
51     $('div');
52     // 添加元素
53     $('body').append(newdiv);*/
54
55     // 解除事件绑定
56     // 解除元素的指定事件类型
57     /*$(' .item').unbind('click')
58     // 解除元素的所有事件类型
59     /*$(' .item').unbind();
60
61     // 事件触发 触发一个元素的指定事件
62     /*$(' .item').trigger('click');
63
64     // 元素的默认行为, 例如a form表单都可以单击时触发跳转或者提交,
65     $('a').click(function(){
66         alert(2);
67         return false;
68     })
69 </script>
70 <!--<div class='item'></div-->
71 </body>
72 </html>

```

p4.html 关于绑定事件，触发事件等

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>事件冒泡</title>
6     <style type='text/css'>
7       div{
8         position:absolute;
9         top:50px;left:50px;
10        border-radius:50%;
11      }
12
13      #one{width:500px;height:500px;background:#369;}
14      #two{width:400px;height:400px;background:pink;}
15      #three{width:300px;height:300px;background:yellow;}
16      #four{width:200px;height:200px;background:#f39;}
17      #five{width:100px;height:100px;background:#cdef;}
18
19    </style>
20  </head>
21  <body>
22    <div id='one'>
23      <div id='two'>
24        <div id='three'>
25          <div id='four'>
26            <div id='five'></div>
27          </div>
28        </div>
29      </div>
30    </div>
31    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
32    <script type='text/javascript'>
33      /*
34      事件冒泡：当触发一个元素的事件时，会自动触发该元素父级和先辈级的同类型事件，造成事件的并发
35      ，导致页面混乱，称为事件冒泡
36      阻止事件冒泡：return false;
37      */
38
39
40      $('div').click(function(){
41        $(this).css('background','black');
42        return false;
43      })
44
45    </script>
46  </body>
47 </html>

```

## p7.html 事件冒泡及事件冒泡的阻止

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>二级菜单</title>
6     <style type='text/css'>
7       ul,li h3{
8         margin:0px;
9         padding:0px;
10        list-style:none;
11      }
12      .all li{
13        width:100px;
14        height:50px;
15        text-align:center;
16        line-height:50px;
17        font-size:20px;
18        border:1px solid red;
19        position:relative;
20      }
21      .menu{
22        width:500px;
23        height:300px;
24        border:1px solid red;
25        position:absolute;
26        top:5px;
27        left:100px;
28        display:none;
29      }
30      .menu li{
31        float:left;
32        width:100px;
33        height:50px;
34        font-size:20px;
35        border:none;
36      }
37    </style>
38  </head>
39  <body>
40    <ul class='all'>
41      <li>
42        <h3>服装</h3>
43        <div class='menu'>
44          <ul>
45            <li>男装</li>
46            <li>女装</li>
47            <li>童装</li>
48          </ul>
49        </div>
50      </li>
51
52      <li>
53        <h3>电器</h3>
54        <div class='menu'>
```



```

55         <ul>
56             <li>电视</li>
57             <li>冰箱</li>
58             <li>洗衣机</li>
59         </ul>
60     </div>
61 </li>
62
63     <li>
64         <h3>手机</h3>
65         <div class='menu'>
66             <ul>
67                 <li>华为</li>
68                 <li>小米</li>
69                 <li>三星</li>
70             </ul>
71         </div>
72     </li>
73
74     <li>
75         <h3>电脑</h3>
76         <div class='menu'>
77             <ul>
78                 <li>笔记本</li>
79                 <li>平板</li>
80                 <li>一体机</li>
81             </ul>
82         </div>
83     </li>
84
85 </ul>
86 <script style='text/javascript' src='jquery-1.8.3.min.js'></script>
87 <script style='text/javascript'>
88     // 获取li 绑定鼠标移入事件
89     $('li').mouseover(function(){
90         // console.log(1)
91         $(this).find('.menu').css('display','block')
92     })
93     // 获取li 绑定鼠标移出事件
94     $('li').mouseout(function(){
95         // console.log(2)
96         $(this).find('.menu').css('display','none')
97     })
98
99
100 </script>
101
102 </body>
103 </html>

```

p8.html 二级菜单的制作（竖着的） 菜单栏横着的版本参考p9.html

## 第五章 ajax操作

同步：提交请求->等待服务器处理->处理完毕返回 这个期间客户端浏览器不能干任何事

异步：请求通过事件触发->服务器处理（这是浏览器仍然可以作其他事情）->处理完毕

同步指的是要等一件事完成之后才可以继续下一件事情，异步指的是可以同时进行

ajax不是一种新的语言，是JS的一种新的方法

server.py只为了启动服务 就可以用IP地址直接访问了，访问的就是启动了的服务器

页面没刷新，刷新了局部，用ajax不刷新的情况下改变局部的内容

Python中就是返回数据，你来请求我，我就给你返回，ajax来请求我，我就print返回给它了，

ajax返回的时候就会带回数据

必须是需要参数的

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <body>
8     <h1>ajax的基本操作，get方法</h1>
9     <button>get方法</button>
10    <br/>
11    <div id='t'></div>
12    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
13    <script type='text/javascript'>
14      $('button').click(function(){
15        //ajax get方法的使用
16        //参数1 请求的URL 参数2，当前请求时携带的参数（必须要用json格式） 参数3，回调函数
17        $.get('/cgi-bin/p1.py',{a:100,b:200},function(data){
18          $('#t').html(data);
19        })
20      })
21    </script>
22  </body>
23 </html>

```

p1.html ajax 的\$.get方法

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <body>
8     <h1>ajax的基本操作，post方法</h1>
9     <button>post方法</button>
10    <br/>
11    <div id='t'></div>
12    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
13    <script type='text/javascript'>
14      $('button').click(function(){
15        //ajax post方法的使用
16        //参数1 请求的URL 参数2，当前请求时携带的参数（必须要用json格式） 参数3，回调函数
17        $.post('/cgi-bin/p1.py',{love:'i love u',a:100,b:200},function(data){
18          console.log(data);
19        })
20      })
21    </script>
22  </body>
23 </html>

```

p2.html ajax的\$.post方法 和\$.get 方法是一样的

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <body>
8     <h1>ajax的基本操作, ajax方法</h1>
9     <button>ajax方法</button>
10    <br/>
11    <div id='t'></div>
12    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
13    <script type='text/javascript'>
14      $('button').click(function(){
15        //ajax方法
16        $.ajax({
17          url: '/cgi-bin/p1.py', //当前请求的url
18          type: 'post', //当前请求的方式
19          data: {uid:100,uname: 'zhangsan'}, //当前请求携带的参数
20          //当前请求成功时的回调函数
21          success: function(data){
22            console.log(data);
23          },
24          //当请求失败时的回调函数
25          error: function(){
26            alert('当前ajax请求失败');
27          },
28          //是否异步 true就是异步 false是同步。不写则默认是异步
29          async: true,
30          //请求超时时间: 单位是毫秒, 注意: 设置请求超时事件只有在异步时才会生效
31          timeout: 2000
32        })
33      })
34
35
36
37    </script>
38  </body>
39 </html>

```

接收的是源代码

p3.html ajax的\$.ajax方法

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title></title>
6   </head>
7   <body>
8     <button>异步</button>
9     <button>同步</button>
10    <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
11    <script type='text/javascript'>
12      //定义一个全局变量
13      var love=100;
14      /*$('button').eq(0).click(function(){
15        $.get('/cgi-bin/p2.py',function(data){
16          love=200;
17        })
18        //处理其它工作
19        alert(love)
20      })*//
21
22      $('button').eq(1).click(function(){
23        //对ajax进行全局配置
24        $.ajaxSetup({
25          async: false
26        })
27        //发送ajax请求

```

```

28     $.get('/cgi-bin/p2.py',function(data){
29         love=200;
30     })
31     //处理其它工作
32     alert(love)
33 })
34
35
36
37 </script>
38
39
40 </body>
41 </html>

```

p4.html 实例分析同步和异步

```

1  # -*- coding: utf-8 -*-
2  import cgi,cgitb,time
3  cgitb.enable()
4
5  print("Content-Type: text/html")    # HTML is following
6  print()
7
8
9
10
11
12
13
14 #返回数据类型
15 #print(0)
16 #print('ok!')
17 lists=[1,2,3,4]
18 #arr=['a','b','abc','cc','ccc','bbs']
19 print(lists)

```

这是在p2.py中的代码 列表会被返回成字符串

```

1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <title>关于返回的数据</title>
6      </head>
7      <body>
8          <button>ajax</button>
9          <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
10         <script type='text/javascript'>
11
12             $('button').click(function(){
13                 $.get('/cgi-bin/p2.py',{uname:'zhangsan'},function(data){
14
15                     //eval() 返回的数据都是字符串格式的 用eval()
16                     //可以转换成原生的列表格式，和js中的数组格式是一致的
17                     eval('var arr='+data);
18
19                     console.log(arr);
20                     console.log(typeof(arr))
21                 })
22             })
23
24         </script>
25     </body>
26 </html>

```

p5.html 关于返回的数据与转换--列表的字符串格式转换成数组

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>test eval()</title>
6   </head>
7   <body>
8     <script type='text/javascript'>
9       /*var str='var a=100';
10      eval(str)//把合法的js的字符串当成是js表达式来执行
11      alert(a)*/
12      var data='[1,2,3,4]';
13      //'var arr=[1,2,3,4]'
14      eval ('var arr='+data);
15      console.log(arr)
16      console.log(typeof(arr))
17
18
19
20    </script>
21  </body>
22 </html>

```

p6.html 测试eval()的转换

```

1 # -*- coding: utf-8 -*-
2 import cgi,cgitb,time
3 cgitb.enable()
4
5 print("Content-Type: text/html")    # HTML is following
6 print()
7
8
9      这是p2.py中的代码 要将字典格式的字符串格式转换为json格式
10
11
12
13
14 #返回数据类型
15 #print(0)
16 #print('ok!')
17 #arr=[1,2,3,4]
18 arr={'a':'abc','b':'bbc','c':'ccc'}
19 print(arr)

```

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>关于返回的数据</title>
6   </head>
7   <body>
8     <button>ajax</button>
9     <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
10    <script type='text/javascript'>
11
12      $('button').click(function(){
13        $.get('/cgi-bin/p2.py',{uname:'zhangsan'},function(data){
14
15          console.log(data);
16          //eval() 返回的数据都是字符串格式的 用eval()
17          可以转换成原生的字典格式，和js中的json格式是一致的
18          eval('var arr='+data);
19
20          console.log(arr);
21          console.log(typeof(arr))
22        })
23      })
24    </script>
25  </body>
26 </html>

```

p7.html 关于返回的数据与转换--字典的字符串格式转换成json格式

```

1 # -*- coding: utf-8 -*-
2 import cgi,cgitb,time
3 cgitb.enable()
4
5 print("Content-Type: text/html")    # HTML is following
6 print()
7
8
9
10
11
12 # 需要引入json模块才可以让字典数据类型进行json格式输出
13 import json
14 # 返回数据类型
15 #print(0)
16 #print('ok!')
17 #arr=[1,2,3,4]
18 arr={'a':'abc','b':'bbc','c':'ccc'}
19 #print(arr)
20 print(json.dumps(arr))

```

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>关于返回的数据</title>
6   </head>
7   <body>
8     <button>ajax</button>
9     <script type="text/javascript" src='jquery-1.8.3.min.js'></script>
10    <script type="text/javascript">
11
12      $('button').click(function(){
13        /*$.get('/cgi-bin/p2.py',{uname:'zhangsan'},function(data){
14
15          console.log(data);
16          console.log(typeof(data));
17
18          //eval('var arr='+data);
19          //console.log(arr);
20          //console.log(typeof(arr))
21
22          //第四个参数 json 要求返回的数据类型必须是json数据。注意：在Python中如果直接返回列表，在ajax中写了第四个参数json之后，会自动转换成js中的数组。但是在Python中直接返回字典或者字符串，数字，那么当前ajax会认为不是json格式数据，那么就认为当前ajax请求失败。
23
24          //如果在Python中是字典格式，则需要写
25          arr={'a':'abc','b':'bbc','c':'ccc'}
26          print(json.dumps(arr))
27
28        },'json')*/
29
30        //ajax方法
31        $.ajax({
32          url: '/cgi-bin/p2.py', //当前请求的url
33          type: 'post', //当前请求的方式
34          data: {uid:100,uname:'zhangsan'}, //当前请求携带的参数
35          dataType: 'json', //要求返回的数据类型
36          //当前请求成功时的回调函数
37          success: function(data){
38            console.log(data);
39          },
40          //当请求失败时的回调函数
41          error: function(){
42            alert('当前ajax请求失败');
43          },
44          //是否异步 true就是异步 false是同步。不写则默认是异步
45          async: true,
46          //请求超时时间：单位是毫秒，注意：设置请求超时事件只有在异步时才会生效
47          timeout: 2000
48        })
49      })
50    </script>
51  </body>
52 </html>

```

p7.html 不用eval()进行转换 直接利用第四个参数



```

1<!DOCTYPE html>
2<html lang="en">
3<head>
4  <meta charset="UTF-8">
5  <title>ajax的注意事项</title>
6</head>
7<body>
8  <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
9  <script type="text/javascript">
10
11    //同源策略
12    // 协议(http https ftp file...) 域名(ip 127.0.0.1 www.baidu.com ) 端口 (80 443 )
13    // $.get('http://127.0.0.1:8000/cgi-bin/1.py',{},function(data){})
14    // $.get('/cgi-bin/1.py',{},function(data){})
15
16    //已拦截跨源请求: 同源策略禁止读取位于 http://dianying.cn/ 的远程资源。
17    // $.get('http://dianying.cn',{},function(data){})
18
19
20
21  </script>
22</body>
23</html>

```

在页面中需要删除某一条信息：

思想：页面通知Python数据库删除 然后再在页面中进行删除或者提示删除失败

a.获取点击的此条消息的编号

b.通过ajax方法将编号发送给Python

c.python通过一系列操作进行删除，删除成功，返回data值

d.根据数据库中删除成功与否在页面中进行删除或者提示失败 )

```

1# -*- coding: utf-8 -*-
2import cgi,cgitb,json
3cgitb.enable()
4
5print("Content-Type: text/html")    # HTML is following
6print()                            # blank line, end of headers
7
8
9
10 fs = cgi.FieldStorage()
11
12 uid = fs["id"].value
13
14 # 链接数据库, 在数据库中删除当前这个用户的数据
15
16
17
18 print(0)

```

p5.py

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title><button>删除</button>操作</title>
6   </head>
7   <body>
8     <table width='800px',border='1'>
9       <tr>
10        <td>id</td>
11        <td>用户名</td>
12        <td>邮箱</td>
13        <td>班级</td>
14        <td>操作</td>
15      </tr>
16
17      <tr>
18        <td>1</td>
19        <td>admin</td>
20        <td>123@qq.com</td>
21        <td>02</td>
22        <td><button>删除</button>|修改</td>
23      </tr>
24
25
26      <tr>
27        <td>2</td>
28        <td>xiaowu</td>
29        <td>13@qq.com</td>
30        <td>02</td>
31        <td><button>删除</button>|修改</td>
32      </tr>
33
34      <tr>
35        <td>3</td>
36        <td>wuwei</td>
37        <td>1@qq.com</td>
38        <td>02</td>
39        <td><button>删除</button>|修改</td>
40      </tr>
41      <script type='text/javascript' src='jquery-1.8.3.min.js'></script>
42      <script type='text/javascript'>
43        $('button').click(function(){
44          //获取用户编号 id
45          var uid=$(this).parents('tr').find('td:first').text();
46          var btn=$(this);
47          //发送ajax 将用户编号发送给Python
48          $.get('./cgi-bin/p5.py',{id:uid},function(data){
49            if(data==0){
50              //删除成功, 需要在页面中直接删除这条数据
51              //$(this).parents('tr').remove();/x
52              //console.log($(this));//当前位置$(this)代表ajax对象(XMLHttpRequest)
53              btn.parents('tr').remove();
54              alert('成功')
55            }else{
56              //删除失败
57              alert('失败')
58            }
59          })
60        })
61
62      </script>
63
64
65
66
67    </table>
68  </body>
69 </html>

```

```

1 # -*- coding: utf-8 -*-
2 import cgi,cgitb,json
3 cgitb.enable()
4
5 print("Content-Type: text/html")    # HTML is following
6 print()                            # blank line, end of headers
7
8
9
10
11 arr = [
12     {'id': '1', 'title': '  小清新格子包臀裙      ', 'pic': '      ./images/3aa3744cf515f5c0b7fd8b21c4c6ed
13     {'id': '2', 'title': '  呛口小辣椒刺绣吊带连衣裙      ', 'pic': '      ./images/b1faf31f316b3079a96a82
14     {'id': '3', 'title': '  立体花喇叭袖短裙雪纺衫      ', 'pic': '      ./images/689dd5eae077459af4a17e9f815
15     {'id': '4', 'title': '  韩国新款 优雅名媛连衣裙      ', 'pic': '      ./images/604a6dad0be72b2426d63e0
16     {'id': '5', 'title': '  欧美单排扣高腰牛仔裙裤      ', 'pic': '      ./images/568309f2a4b2ee0c4ed1f1f9d6c
17     {'id': '6', 'title': '  宽松短袖连衣裙      ', 'pic': '      ./images/f45c350eb2f192077c450164b8986716.jp
18     {'id': '7', 'title': '  沙滩短裙欧根纱公主裙      ', 'pic': '      ./images/520330a46fdc687f8b325ee5a16a
19     {'id': '8', 'title': '  纯色荷叶边包臀裙      ', 'pic': '      ./images/27f7b404889169e12edabb23e8d808
20     {'id': '9', 'title': '  欧根纱卡通印花无袖连衣裙      ', 'pic': '      ./images/0168e002115edc79a02811
21     {'id': '10', 'title': '  夏季美裙露肩印花连衣裙      ', 'pic': '      ./images/edb14f72deb2a2c5566d96beefa
22     {'id': '11', 'title': '  棉麻荷叶袖抽绳条纹连衣裙      ', 'pic': '      ./images/325b5c0e5c133cb61171d2
23     {'id': '12', 'title': '  做旧插兜双口袋牛仔半身裙      ', 'pic': '      ./images/99034a47383ce8ead2ce3f
24     {'id': '13', 'title': '  宽松大码喇叭袖连衣裙      ', 'pic': '      ./images/605190a25644c015ebada86c5e81
283     {'id': '272', 'title': '  吊带露肩一字领连衣裙短裙      ', 'pic': '      ./images/3f4f9664d95771901014a2
284     {'id': '273', 'title': '  韩版蕾丝短裙背心裙连衣裙      ', 'pic': '      ./images/f9526cafcc30994e90feae
285     {'id': '274', 'title': '  韩版蕾丝短裙背心裙连衣裙      ', 'pic': '      ./images/f9526cafcc30994e90feae
286     {'id': '275', 'title': '  韩版蕾丝短裙背心裙连衣裙      ', 'pic': '      ./images/f9526cafcc30994e90feae
287     {'id': '276', 'title': '  吊带露肩一字领连衣裙短裙      ', 'pic': '      ./images/3f4f9664d95771901014a2
288     {'id': '277', 'title': '  吊带露肩一字领连衣裙短裙      ', 'pic': '      ./images/3f4f9664d95771901014a2
289     {'id': '278', 'title': '  吊带露肩一字领连衣裙短裙      ', 'pic': '      ./images/3f4f9664d95771901014a2
290 ]
291
292 fs=cgi.FieldStorage()
293 p=int(fs['p'].value)
294
295 # p=1 0-15(1-16)  p=2 16-31(17-32)
296 # 通过参数 确认返回的数据
297 end=p*16
298 start=end-16
299 #print(start,end)
300 print(json.dumps(arr[start:end]))

```

## p6.py 充当数据库

```

1 <!DOCTYPE html>
2 <html lang="en">
3     <head>
4         <meta charset="UTF-8">
5         <title></title>
6         <style type='text/css'>
7             div,ul,li,img,h3,p{
8                 margin:0;
9                 padding:0;
10                list-style:none;
11            }
12            ul{
13                float:left;
14                width:25%;
15                height:200px;
16            }
17            ul li{
18                text-align:center;
19                border:1px solid red;
20            }
21        </style>
22    </head>
23    <body>
24        <li class='item' style='display:none'>
25            <h3>商品名称</h3>

```

```

28     <p>价格: <span>100元</span></p>
29     <img src='./images/0c0284bb404ac7213ef8cd1acdbfdc2f.jpg' alt='' width='200px' height='
    300px'>
30 </li>
31 <ul></ul>
32 <ul></ul>
33 <ul></ul>
34 <ul></ul>
35 <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
36 <script type="text/javascript">
37 //全局变量
38 var p=1;
39
40 //封装函数，进行ajax请求数据
41 function requestGoods(){
42     $.get('/cgi-bin/p6.py',{p:p},function(data){
43         console.log(data);
44         //处理数据[{}, {}, {}]
45         for (var i=0; i<data.length;i++){
46             //data[i]
47             //console.log(data[i].title);
48
49             //首先需要克隆
50             var newLi=$('.item').eq(0).clone();
51             newLi.find('h3').text(data[i].title);
52             newLi.find('span').text(data[i].price);
53             newLi.find('img').attr('src',data[i].pic);
54             newLi.show();
55             //console.log(newLi)
56             //0 1 2 3 0 1 2 3
57             var ind=i%4
58             $('ul').eq(ind).append(newLi)
59
60
61
62         };
63
64         //如果请求完成一次，让参数++
65         p++;
66
67     },'json')
68 }
69
70
71 requestGoods();
72 //因为是异步的，没等回来，又接着调用，没来得及加一
73 setTimeout(function(){
74     requestGoods();
75 },2000)
76
77
78 //绑定文档滚动事件
79 $(window).scroll(function(){
80     //获取整个文档的高度
81     var dh=$(document).height();
82     //获取可视区域高度
83     var ch=$(window).height();
84     //获取文档滚动距离
85     var dt=$(window).scrollTop();
86
87     //判断是否触底
88     if(dh==ch+dt){
89         requestGoods();
90     }
91 })
92
93 </script>
94 </body>
95 </html>

```

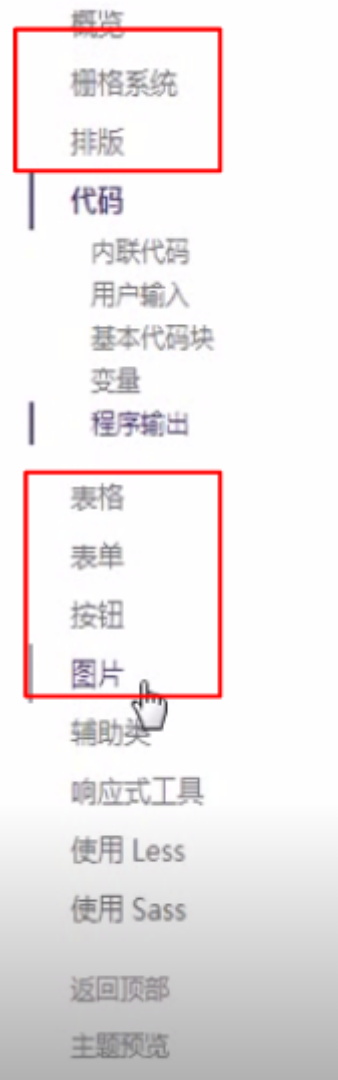
p8.html 瀑布流

## 第六章 bootstrap

响应式布局：**响应式布局**是EthanMarcotte在2010年5月份提出的一个概念，简而言之，就是一个网站能够兼容

多个终端——而不是为每个终端做一个特定的版本。这个概念是为解决移动互联网浏览而诞生的。**响应式布局**可以为不同终端的用户提供更加舒适的界面和更好的用户体验，而且随着目前大屏幕移动设备的普及，用大趋势来形容也不为过。随着越来越多的设计师采用这个技术，我们不仅看到很多的创新，还看到了一些成形的模式。

互联网->移动互联网->人工智能



```

1 <!DOCTYPE html>
2 <html lang='zh-CN'>
3   <head>
4     <!-- 设置字符集-->
5     <meta charset='utf-8'>
6     <!-- 让IE开启它的兼容模式-->
7     <meta http-equiv='X-UA-Compatible' content='IE-edge'>
8     <!-- 让手机的屏幕端口正常显示-->
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10    <!-- 上面3个标签必须放在最前面, 任何其他内容都必须跟随其后-->
11
12    <title>Bootstrap 101 Template</title>
13    <!-- Bootstrap -->
14    <link rel="stylesheet" href="./ppublic/css/bootstrap.min.css">
15    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
16    <script src="./ppublic/js/jquery-1.10.2.min.js"></script>
17    <!-- Include all compiled plugins (below), or include individual files as needed -->
18    <script src="./ppublic/js/bootstrap.min.js"></script>
19
20  </head>
21  <body>
22    <h1>Hello, world!</h1>
23    <div class='col-md-6'>
24      <table class='table'>
25        <tr>
26          <td>id</td>
27          <td>name</td>
28          <td>sex</td>
29          <td>age</td>
30        </tr>
31

```

p1.html

```

1 <!DOCTYPE html>
2 <html lang='zh-CN'>
3   <head>
4     <!-- 设置字符集-->
5     <meta charset='utf-8'>
6     <!-- 让IE开启它的兼容模式-->
7     <meta http-equiv='X-UA-Compatible' content='IE-edge'>
8     <!-- 让手机的屏幕端口正常显示-->
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10    <!-- 上面3个标签必须放在最前面, 任何其他内容都必须跟随其后-->
11
12    <title>Bootstrap 101 Template</title>
13    <!-- Bootstrap -->
14    <link rel="stylesheet" href="./ppublic/css/bootstrap.min.css">
15    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
16    <script src="./ppublic/js/jquery-1.10.2.min.js"></script>
17    <!-- Include all compiled plugins (below), or include individual files as needed -->
18    <script src="./ppublic/js/bootstrap.min.js"></script>
19
20  </head>
21  <body>
22    <!--<div class='container-fluid'>-->
23    <div class='container'>
24      <div class='row'>
25        <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#909;'><h3>开心</h3></div>
26        <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#603;'>难过</div>
27        <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#369;'>悲伤</div>
28        <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#963;'>崩溃</div>
29        <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#903;'>喜悦</div>
30        <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#603;'></div>
31        <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#f39;'></div>

```

```

32     <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#99f;'></div>
33     <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#9f9;'></div>
34     <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#369;'></div>
35     <div class='col-xs-3 col-sm-2 col-md-1' style='height:50px; background:#903;'></div>
36 </div>
37
38 <div class='row'>
39
40     <div class='col-xs-6' style='height:150px;background:pink;'></div>
41     <div class='col-xs-6' style='height:150px;background:#cdef;'></div>
42     <div class='col-xs-12' style='height:100px;background:yellow;'></div>
43
44 </div>
45
46 <hr/>
47 <div class='row'>
48     <div class='col-md-2' style='background:#969;height:300px;'>
49         <abbr title="attribute">attr</abbr>
50     </div>
51     <div class='col-md-4 col-md-offset-4' style='background:#369;height:300px'>
52         <blockquote>
53             <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.</p>
54         </blockquote>
55     </div>
56
57 </div>
58
59 </div>
60 </body>
61 </html>

```

p2.html

```

7     <meta http-equiv='X-UA-Compatible' content='IE-edge'>
8     <!-- 让手机的屏幕端口正常显示-->
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10    <!-- 上面3个标签必须放在最前面，任何其他内容都必须跟随其后-->
11
12    <title>Bootstrap 101 Template</title>
13    <!-- Bootstrap -->
14    <link rel="stylesheet" href="./ppublic/css/bootstrap.min.css">
15    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
16    <script src="./ppublic/js/jquery-1.10.2.min.js"></script>
17    <!-- Include all compiled plugins (below), or include individual files as needed -->
18    <script src="./ppublic/js/bootstrap.min.js"></script>
19 </head>
20 <body>
21     <div class='container-fluid'>
22         <div class='row'>
23             <div class='col-md-12' style='height:200px;'></div>
24             <div class='col-md-6 col-md-offset-3'>
25                 <table class="table table-bordered table-striped table-hover">
26                     <tr class="active">
27                         <th>id</th>
28                         <th>name</th>
29                         <th>id</th>
30                         <th>age</th>
31                         <th>sex</th>
32                         <th>hobby</th>
33                     </tr>
34
35                     <tr class="success">
36                         <th>1</th>
37                         <th>li</th>

```



```

68         <th>ou</th>
69     </tr>
70
71 </table>
72
73 </div>
74 </div>
75 <hr/>
76 <div class='row'>
77 <div class='col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3'>
78 <div class="table-responsive">
79 <table class="table table-bordered table-striped table-hover">
80 <tr class="active">
81 <th>id</th>
82 <th>name</th>
83 <th>id</th>
84 <th>age</th>
85 <th>sex</th>
86 <th>hobby</th>
87 </tr>
88
89 <tr class="success">
90 <th>1</th>
91 <th>li</th>
92 <th>1</th>
93 <th>15</th>
94 <th>nv</th>
95 <th>ou</th>
96 </tr>
97

```

p3.html