

【手把手】JavaWeb 入门级项目实战 -- 文章发布系统 （第三节）

3.6.6 轮播图的基本实现

上一节中，轮播图已经做了一大半，我今天上午抽空把剩下的代码写完了。现在就大概说一下思路吧。实现上一张和下一张的效果，基本上就是改变ul的margin-left值。为了保证无缝滚动，我们需要在5张图片的首部和尾部各加一张图片。像这样：

```
<ul>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
  <li class='fl'>
    <a href="javascript:void(0)">
      
    </a>
  </li>
</ul>
```

接下来，编写移动图片的方法，为了控制下一张和上一张，我们定义一个索引变量index就可以啦。

```
var index = 0;
```

获取图片的宽度的数量

```
var imgwidth = $('.banner .content ul li').width(); //获取轮播图片的宽度
var len = $('.banner .content ul li').length - 2//获取总共的图片数量
```

移动图片的方法

```
function movePicture(){
  $('.banner .content ul').animate({'margin-left':-imgwidth * (index+1)},1000,function(){
    if(index == len){
      $(this).css('margin-left',-imgwidth);
      index = 0;
    }
    if(index == -1){
      $(this).css('margin-left',-imgwidth * len);
      index = len - 1;
    }
  });
}
```

最后，给按钮绑定事件：

```
rightBtn.on('click',function(){
  index ++;
  movePicture();
});

leftBtn.on('click',function(){
  index --;
  movePicture();
});
```

3.6.7 按钮的节流控制

接下来要做一个简单的前台节流，意思就是说，如果有用户闲得无聊，在那拼命的点击下一张的按钮，那么每一次点击都会触发movePicture函数，这个时候，你就会看到图片在那乱跳。

所以，为了避免这种用户的操作，我们需要用js来做一个节流，额，好像是叫节流。我记不太清了。意思就是说，比如你在一秒钟内连续点击了8次，那么我就认为这8次中只有1次是有效的，其他的点击都是误操作。

这样的话，无论你怎么狂点鼠标，我都只调用一次movePicture函数。

首先，我们定义一个空的定时器。

```
var timer = null;
```

思路为：

每当我触发按钮的点击事件，就把当前的定时器timer清零，然后又马上给他设置定时内容，比如500毫秒后才进行移动操作。也就是说，如果你狂点鼠标，那么定时器就会被马上清零，然后进入定时器的等待状态。必须要等你满了500毫秒，才进行下面的操作。

原理就是这么简单，这个方式也经常用于登陆按钮上。比如，当你网速很慢的时候，点击登陆，网页没有立即给你跳转页面。有些用户闲着无聊，就在那狂点鼠标，不知道你有没有这么做过呢？

如果没有做好节流，那么就会有大量的请求传递到服务器，会给服务器造成很大的压力。

差不多就是这个意思，上代码：

```
rightBtn.on('click',function(){
```

```

clearTimeout(timer); //定时器清零
timer = setTimeout(function(){
    index ++;
    movePicture();
},500);

});

leftBtn.on('click',function(){
    clearTimeout(timer); //定时器清零
    timer = setTimeout(function(){
        index --;
        movePicture();
    },500);
});

```

不论我鼠标怎么狂点，它每次都只移动一张图片。

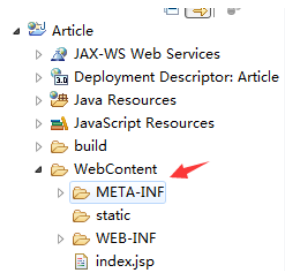
好了，轮播图差不多就聊到这里吧，你可以自己去慢慢完善，时间关系我就不扩展太多了。剩下的就交给你自己去玩耍了，比如，我能不能做一个定时轮播呢，每个5秒钟就自动切换到下一张。

我就不写了。

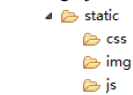
3.6.8 整合资源文件

现在，我们把这个已经做好的banner区域搬到项目里。

在WebContent目录下新增一个static文件夹，这里存放各种静态资源文件。



比如img, js, css等



因为刚才的页面中用到了百度资源库里的jQuery文件：

<http://libs.baidu.com/jquery/2.0.0/jquery.min.js>

现在我们直接将这个url用浏览器打开。

```

/*! jQuery v2.0.0 | (c) 2005, 2013 jQuery Foundation, Inc. | jquery.org/license
//@ sourceMappingURL=jquery.min.map
*/
(function(e,undefined){var t,r,r=typeof undefined,i=e.location,o=e.document,s=o.documentElement,a=e.jQuery,u=e.$,l={},c=
[],f="2.0.0",p=c.concat,h=c.push,d=c.slice,g=c.indexOf,m=l.toString,y=l.hasOwnProperty,v=f.trim,x=function(e,n){return new x.fn.init(e,n,t)},b=/[+]?(\?:\d*\.\d+)?(\d+)?(\d+)?/g,source,w=/\S+/g,T=/^(?
<([\w\W]+>)[>]*\#([\w-]*)$/C,/^(?<([\w+]\s*\w+)?(?:<[\w\>])$/k,/^-ms-/,N/-([\da-z])/gi,E=function(e,t){return t.toUpperCase()},S=function(

```

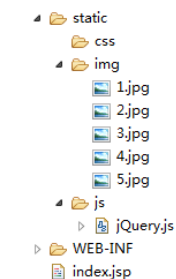
全部复制。

然后在js文件夹中新建一个jQuery.js文件。

现在，将刚才复制的内容原封不动地拷贝进去。

ctrl + s 保存。

OK，jQuery文件已经有了，接下来，我们将素材图片拷贝到img文件夹中。这样，资源文件就全部准备好了。



3.6.7 引用资源文件

在整合好资源文件后，如何将资源文件导入我们的index.jsp呢？这是一个问题。

我们将html代码和css样式拷贝到index.jsp中，看看效果。



图片都没有显示出来。

那么，如何改变img的src，使他能够成功引到/Article/WebContent/static/img 里面的图片呢？

我提供一种比较方便的办法，就是在jsp页面中写Java代码，让Java来获取项目的根路径，然后通过绝对路径的方式引入资源文件。

我们在jsp页面的开头加上以下代码：

```
<%
    String path = request.getContextPath();
    int port = request.getServerPort();
    String basePath = null;
    if(port==80){
        basePath = request.getScheme()+ "://" + request.getServerName() + path;
    }else{
        basePath = request.getScheme()+ "://" + request.getServerName() + ":" + request.getServerPort() + path;
    }
    request.setAttribute("basePath", basePath);
%>
```

request就是所谓的JSP九大隐式对象中的一员，其实JSP就是Servlet，你可能学过Servlet，那么应该很清楚request是什么意思。以后我会单独来讲解一下关于JSP的东西，现在先继续往下说吧。

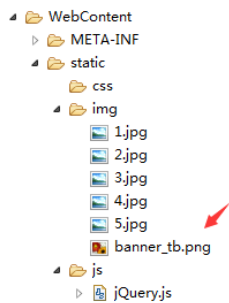
request.setAttribute("basePath", basePath) 表示将得到的basePath（项目根路径）存放到request作用域中，你可以把request看做一个HashMap，或者一个JSON对象，都可以。

反正，这句话一写，就说明request作用域中已经有了basePath，我们在JSP页面的其他地方就可以获取到了。

```
<div class="banner">
    <div class='content'>
        <ul>
            <li class='fl'>
                <a href="javascript:void(0)">
                    
                </a>
            </li>
            <li class='fl'>
                <a href="javascript:void(0)">
                    
                </a>
            </li>
            <li class='fl'>
                <a href="javascript:void(0)">
                    
                </a>
            </li>
            <li class='fl'>
                <a href="javascript:void(0)">
                    
                </a>
            </li>
            <li class='fl'>
                <a href="javascript:void(0)">
                    
                </a>
            </li>
            <li class='fl'>
                <a href="javascript:void(0)">
                    
                </a>
            </li>
            <li class='fl'>
                <a href="javascript:void(0)">
                    
                </a>
            </li>
        </ul>
        <span class='banner_left'><i class='btn_left'></i></span>
        <span class='banner_right'><i class='btn_right'></i></span>
    </div>
```

这样就能访问到了。

刚才还忘了一样东西，就是左右按钮的图片，也把它拷贝进来。



引入

```
.btn_left , .btn_right{
    display: inline-block;
    width: 21px;
    height: 150px;
    background: url(${basePath}/static/img/banner_tb.png) no-repeat;
    position: absolute;
    left: -38px;
    top: 90px;
    opacity: 0;
    transition: all ease 0.6s;
    cursor: pointer;
}
```

顺便把banner图往上一一点:

```
.banner {
    height: 400px;
    margin-top: 2px;
    overflow: hidden;
}
```

把js代码也全部拷贝过来吧。

```
var leftBtn = $('#.btn_left').eq(0); //左按钮
var rightBtn = $('#.btn_right').eq(0); //右按钮

var ul = $('#.banner .content ul').eq(0);

var index = 0;
var timer = null;

var imgwidth = $('#.banner .content ul li').width(); //获取轮播图片的宽度
var len = $('#.banner .content ul li').length - 2 //获取总共的图片数量

rightBtn.on('click',function(){
    clearTimeout(timer); //定时器清零
    timer = setTimeout(function(){
        index ++;
        movePicture();
    },500);
});

leftBtn.on('click',function(){
    clearTimeout(timer); //定时器清零
    timer = setTimeout(function(){
        index --;
        movePicture();
    },500);
});

function movePicture(){

    $('#.banner .content ul').animate({'margin-left':-imgwidth * (index+1)},1000,function(){
        if(index == len){
            $(this).css('margin-left',-imgwidth);
            index = 0;
        }
        if(index == -1){
            $(this).css('margin-left',-imgwidth * len);
            index = len - 1;
        }
    });
}
```

发现一个小BUG,当我一直点击下一张,移动到最后一张图片的时候(也就是第一张图片,因为我们在末尾加上了第一张图片),第二张图片(有的苹果那张)没有自动衔接上去。

这是咋回事呢?

有BUG不要怕,去找原因就行了。

经过检查,我发现了一个单词的拼写错误:

```
function movePicture(){
    $(' .banner .content ul').animate({'margin-left':-imgwidth * (index+1)},1000,function(){
        if(index == len){
            $(this).css('masrgin-left',-imgwidth);
            index = 0;
        }
        if(index == -1){
            $(this).css('margin-left',-imgwidth * len);
            index = len - 1;
        }
    });
}
```

原来是margin-left写错了。

OK，更正后就正确了，我顺便把demo页面也改好了。

```
function movePicture(){
    $(' .banner .content ul').animate({'margin-left':-imgwidth * (index+1)},1000,function(){
        if(index == len){
            $(this).css('margin-left',-imgwidth);
            index = 0;
        }
        if(index == -1){
            $(this).css('margin-left',-imgwidth * len);
            index = len - 1;
        }
    });
}
```

改正就行了，额，小细节不要在意。。。

banner图这样就差不多了，下一节开始做登陆功能。我先把页面画好，以后页面的绘制我就不介绍得这么详细了，重点还是以Java为主。

免责声明： 博客中所有的图片素材均来自百度搜索，仅供学习交流，如有问题请联系我，侵立删，谢谢。