先从流程说起：

svn：svn里面的代码是除去静态资源的，开发将代码提交到一个分支，分支不断更新，原来的代码注意merge到新的分支，merge过程细节已备注。代码更新过后，svn代码测试需要重启服务器重启Athena，重启之后看更新的代码。如果是环境部署失败，环境重部署后测试重启Athena。环境部署，注意进入实例管理勾选当前实例点击环境部署。Cd-promotion>--dev>构建版本，实例管理，注意勾选，>--实例管理注意勾选，然后部署版本。核对当前构建的版本号，查看git仓库提交记录，找到自己merge的记录，查看版本号与git提交记录的版本号是否一致，查看环境部署是否成功，打开当前页面，找到更改的静态文件，右击在新的页面打开，如果代码没有变更，更改末尾的时间戳，代码变为自己更改后的代码，那就是测试没有重启Athena，如果没有变化，那就是环境没有部署成功，环境部署后测试重启athena。

预发环境：开发完善了需求和错误改进之后测试在测试环境下觉得页面可以发预发，这次预发后出现了的一个问题就是页面的静态资源并没有加载，分析原因是因为静态资源发布的时间晚于athenna的部署时间，也就是说athena系统部署了之后静态资源并没有加载到网页里，然后测试重启了athenna之后发现js变了，但是图片没变，那是因为之前load的非实时的静态资源图片并且缓存在服务器里了，更改图片的时间戳就会看到新的图片。这样防止服务器缓存不变的情况通常在css的背景图片url的png后面加上“？日期”，这样服务器就不会缓存了图片并且再也不更改。

上线发布：预发环境下测试没问题以后，上线，上线需要开发将资源发布，这边需要pe邮件给开发注明网页上线，发布的不仅仅是svn的代码，还要有静态资源的代码，今天pe漏发静态资源的部分，导致网页显示错误。Svn的代码最后后台会帮忙merge到发布的分支，静态资源的代码需要自己merge到master分支，步骤和merge到develop分支一样。

如果svn代码改变的话，测试需要重新部署环境，如果是静态资源文件改变的话，自己需要部署环境，测试重启Athena。

再说ftl页面：出的错，ftl引进了freemarker模板可供后台解析，页面中导航栏用到了freemarker的部分，引入了判断语句<#if accountId??>表示判断是已登录，然后<#else>没有登录的情况的html。下面用到了两个判断，判断迅雷码是否发放完毕，后台传参数iscodeuseout（类似的关键字），前端要取到这个字段的值做判断，比如取值的写法：${iscodeuseout}，这里代码中直接对该关键字做了判断，所以前端直接写判断的代码：

<#if isCodeUsedOut="true">

<a…></a>

<#else if accountId??>//如果用户登录，accountId不为空。

<a>..</a>

</#if>//必须写结尾，不然本地服务器会报错。

注意参数的设置，链接后的参数，是不是每个理财的链接都会有参数。监控的代码需要研究。另链接给一个onclick事件：<a href=’#‘onClick="easyNav.login2()">调用这个函数，弹出登陆框，其中这个函数里调用了写好的登陆组件//查看组件的编写。

尽快将freemarker学完。

Css页面注意浮动后的清除浮动，之前立即支付一分钱模块，之前的css样式包含了.a{line-height:200;overflow:hidden} <a >立即支付一分</a>这样的样式将文字行撑开得非常大，然后overflow：hidden了之后文字看不到，去除这个样式就能在div上看见文字。

关于css的命名：.mod-banner .operator .btn-buy其中前者是后者的父元素节点，如果是对并列的节点命名，中间用逗号隔开。

关于js，ajax：

Js：采用了jquery框架（进一步学习jquery）

Ajax：先从后台传的参数来看：/epayhbajax/getXunleiCode.htm

返回：

{"result":"success","errorMsg":"成功","errorMap":null,"ppcPassCoordinate":null,"attach":

{"data":"aXunleiRegistercode"}

,"success":true}

若没有注册码，"data":"暂无激活码"

在贴上ajax代码：



此处是判断用户是否获得激活码而决定id=#popAj下的html代码，点击完点此参与活动应该关闭div，这时应该在当前链接的节点获得这个节点引入click时间，如图中代码所示。