1.关于class命名的问题，如果命名class=“uio jk”，则表示该块元素同时引用了uio和jk的样式。

2.Class样式规定方式，如.loginPopBox .loginBtn2{…;}后者被前者包含，样式对处于loginPopBox中的loginBtn2起作用。

3.进度条的实现，外层div固定宽度和高度，给一个背景黑色，内层div固定高度，给一个内嵌样式<div class=”” style=”width:90%”>，给这个div一个背景绿色，实时更改进度条的百分比数值，就可以得出实时的百分比。



项目CSS：

Cusor:pointer鼠标放在该元素上时呈现手点击样式。.对应class，#对应id。

background-position: -8px -103px 背景图片向左移动8px，向下移动103px。

font-family: "微软雅黑","宋体";表示如果用户没有微软雅黑，就以宋体显示。

Html文件介入freemarker指令使得静态网页变得动态化。

FreeMarker 背后（确切的说是MVC 模式的背后）的重要思想就是表现逻辑和业务逻辑相分离。

4.关于ftl-server的使用和理解。

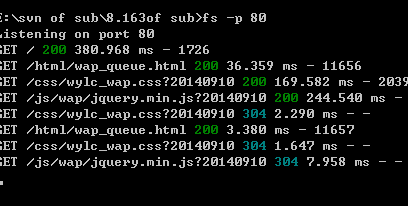
Ftl-server是一款前端开发工具，主要功能和tomcat类似把工程文件部署出来，放在localhost里浏览网页，ftl可以支持freemark模板解析，用命令行启动该工具，就可以把工程文件部署出来，在本地localhost就可以前端开发。

首先启动命令行窗口：

安装：npm install ftl-server –g

* 安装结束之后进行使用：-c, --config 指定配置文件。但不是必需的, 如果没有指定，则寻找当前工作目录下的ftl.config.js作为配置文件
* -p, --port 服务的端口，默认8000
* -l, --log 配置打印日志，-l no-static: 不打印静态资源的请求; -l no-error+no-static: 不打印静态资源请求和freemarker错误日志
* -h, --help 帮助

通常我们会把配置文件用记事本写下配置程序，然后配置程序的记事本文件放在项目（svn/git）文件的目录下（一般为总目录），然后ftl-server会自动找到该配置文件，这样就省去了代码行寻找配置文件，之后开启端口服务fs –p 80



就可以启动服务器了，localhost。

最后，静态文件在localhost网页目录是隐藏的，所以如果要找apach下的静态文件，可以采取：localhost/html/wap\_queque.html.

配置文件程序如下：module.exports **=** {

  'public'**:** 'E:\\20150228-origin-2\\20150228-origin-2\\apache',

  port**:** '80',

  ftl**:** {

    base**:** 'E:\\20150228-origin-2\\20150228-origin-2\\fund-web\\src\\main\\webapp\\WEB-INF\\ftl',

    global**:** {

    },

    'productDetail\_000008\_new.ftl': function(req, res) {

**return** {

        saleActivityMap**:** {

          "000008"**:** {

            activityStatus**:** 'actived'

          }

        }

      }

    }

  },

  mock**:** [{

    url**:** '/request',

    method**:** 'get',

    status**:** '200',

    header**:** {

    },

    response: function(req, res) {

**return** {

        a**:** 1,

        B**:** 2

      }

    }

  }],

  proxy**:** [{

    path**:** '/proxy1',

    target**:** 'http://localhost:3000'

    }

  ]

}