**FreeMarker介绍**

# 1 FreeMarker介绍

## 1.1 FreeMarker简介

1. FreeMarker是一款[模板引擎](https://baike.baidu.com/item/%E6%A8%A1%E6%9D%BF%E5%BC%95%E6%93%8E" \t "_blank)： 即一种基于模板和要改变的数据， 并用来生成输出文本（[HTML](https://baike.baidu.com/item/HTML" \t "_blank)网页、[电子邮件](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6" \t "_blank)、[配置文件](https://baike.baidu.com/item/%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6" \t "_blank)、[源代码](https://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81/3969" \t "_blank)等）的通用工具。 它不是面向最终用户的，而是一个Java类库，是一款程序员可以嵌入他们所开发产品的组件。
2. FreeMarker是[免费](https://baike.baidu.com/item/%E5%85%8D%E8%B4%B9/131326" \t "_blank)的，基于Apache许可证2.0版本发布。其模板编写为FreeMarker Template Language（FTL），属于简单、专用的语言。需要准备数据在真实编程语言中来显示，比如[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93" \t "_blank)查询和业务运算， 之后模板显示已经准备好的数据。在模板中，主要用于如何展现数据， 而在模板之外注意于要展示什么数据。展示数据
3. FreeMarker是一款展示数据的模板。其模板代码在.ftl（FreeMarker Template Language）文件中编写，属于简单、专用的语言。ftl——---html



1. J2EE动态页面展示数据的方式主要有下面这些方式：Jsp、Struts标签、jstl、freeMarker等。

总结：

freeMarker：

展示数据用的 .ftl

模板：.ftl+data model--🡪html文件 (静态化)

页面静态化技术

## 1.2 使用引入

1. 需要在xml进行ftl解析配置
2. 使用ftl模板语句时需要在.ftl文件中

# 2 ftl基本语法

## 2.1 if else 语句

1、if语句：<#if boolean表达式> </#if>

（1）判断后端返回的Java对象是否为空（null）：<#if Java对象或者属性??> 执行代码 </#if>

说明：Java对象可以是集合、普通Java类或者属性。

（2）判断集合size：<#if 集合对象?size == 0> 执行代码 </#if>

（3）判断值相等：<#if 属性变量 == 2> 执行代码 </#if>

2、if else 语句：<#if orgList?size == 0> 执行代码 <#else> 执行代码 </#if>

## 2.2 list循环语句

1. 遍历集合：
2. <#list userList as index>

</#list>

2、范围内遍历：

示例：<#list startPage..endPage as index> </#list>

类似java代码:

int startPage =5;

int endPage =10;

for(int index=startPage;index<=endPage ;index++）{

System.out.println(index)

}

## 2.2 输出展示

1. 输出后台返回的普通字符串或者数值：

<#if 变量名??>${变量名}</#if>

示例：<#if total??>${total}</#if>

1. 输出后台返回的对象属性：

<#if 对象名.对象属性??>${对象名.对象属性}</#if>

示例：：<#if org.orgName??>${org.orgName}</#if>

3、时间输出：

示例：<#if org.createdDate??>${org.createdDate?string("yyyy-MM-dd HH:mm:ss")}</#if>

## 2.3 包含

<#include "commons/header.html">

## 2.4 freemarker的注释

<#-- freemarker的注释

使用freemarker包含另一个页面

-->

freemarker的注释在页面不显示

html的注释会显示到页面<!-- html注释 -->

# 3示例练习

## 示例介绍

使用freemaker 需要jar

freemaker…….jar

spring-context-support.jar

以前的方式： 这是针对jsp页面的（效率低）

*<!-- 防止静态资源被拦截 -->  
<!-- 4,viewResolver -->  
<!--<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">-->  
 <!--<property name="suffix" value=".jsp"></property>-->  
 <!--<property name="prefix" value="/WEB-INF/jsp/"></property>-->  
<!--</bean>-->*

Ftl文件：*templateLoaderPath，前缀*

spring-mvc.xml

|  |
| --- |
| <!-- freemarker 配置 -->  <bean id=*"freemarkerConfigurer"*  class=*"org.springframework.web.servlet.view.freemarker.FreeMarkerConfigurer"*>  <property name=*"templateLoaderPath"* value=*"/WEB-INF/ftl"* />  <property name=*"defaultEncoding"* value=*"UTF-8"* />  <property name="freemarkerSettings">  <props>  <prop key=*"template\_update\_delay"*>10</prop>  <prop key=*"locale"*>zh\_CN</prop>  <prop key=*"datetime\_format"*>yyyy-MM-dd HH:mm:ss</prop>  <prop key=*"date\_format"*>yyyy-MM-dd</prop>  <prop key=*"time\_format"*>HH:mm:ss</prop>  <prop key=*"number\_format"*>#.##</prop>  </props>  </property>  </bean>  <!-- 配置FreeMark视图 -->  <bean id=*"freeMarkerViewResolver"*  class=*"org.springframework.web.servlet.view.freemarker.FreeMarkerViewResolver"*>  <property name=*"contentType"* value=*"text/html;charset=UTF-8"* />  <property name=*"suffix"* value=*".ftl"* />  <property name="*requestContextAttribute*" value="*request*" /><!—页面可以得到项目绝对路径-->  </bean> |

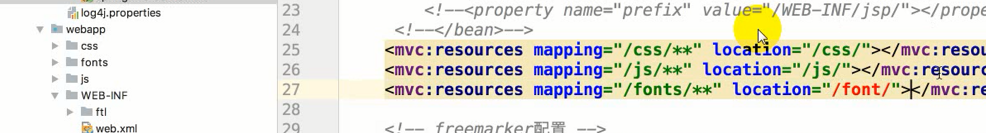


Jstl,是Java标准标签库，用的多的是C标签，或者自定义分页标签。





Bootstrap插件：



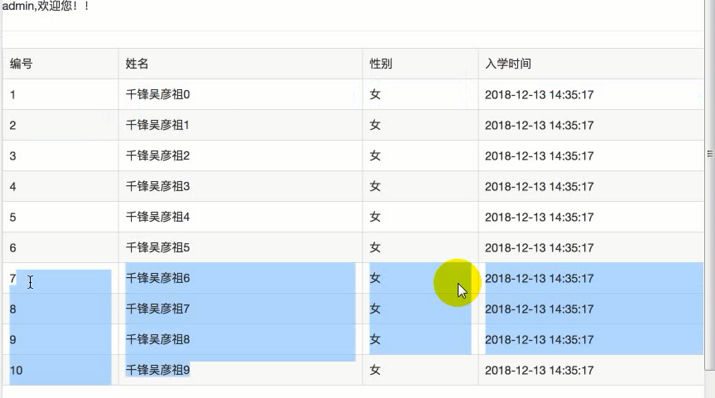
Table给的属性是隔行变色



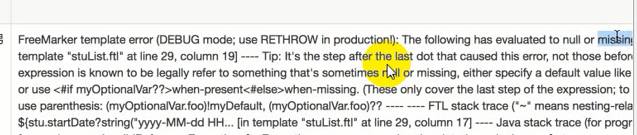
处理时间字段：



展示效果：

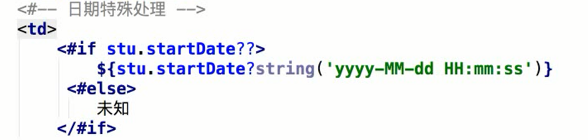


假如传送的字段有null，



处理： 假如非空，展示，否则为

‘未知’



*/\*\*  
 \* 演示freemarker的第二个功能*

*作为模板引擎可以实现页面的静态化  
 \* 模板+数据模型----》html文件  
 \** ***@return*** *\*/*@RequestMapping(**"/getHtml"**)  
@ResponseBody  
**public** String getHtml(){  
 **try** {  
 *//得到模板对象* Configuration configuration = **configurer**.getConfiguration();  
 Template template= configuration.getTemplate(**"stuList.ftl"**);  
  
 *//得到数据模型* List<Student> list = StudentDao.*getList*();  
 Map<String,Object> map = **new** HashMap<String, Object>();  
 map.put(**"stus"**,list);  
 map.put(**"name"**,**"尼古拉斯.赵四"**);  
 *//生成html文件* FileWriter fw = **new** FileWriter(**"/Users/liliting/Desktop/JavaEE/三阶段/freeMarker/stuList.html"**);  
  
 template.process(map,fw);  
  
 System.***out***.println(**"success"**);  
 **return "success"**;  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 **return** e.getMessage();  
 }  
  
  
}

把Html文件，常用的数据库展示的页面的这个文件导出成了HTML文件。