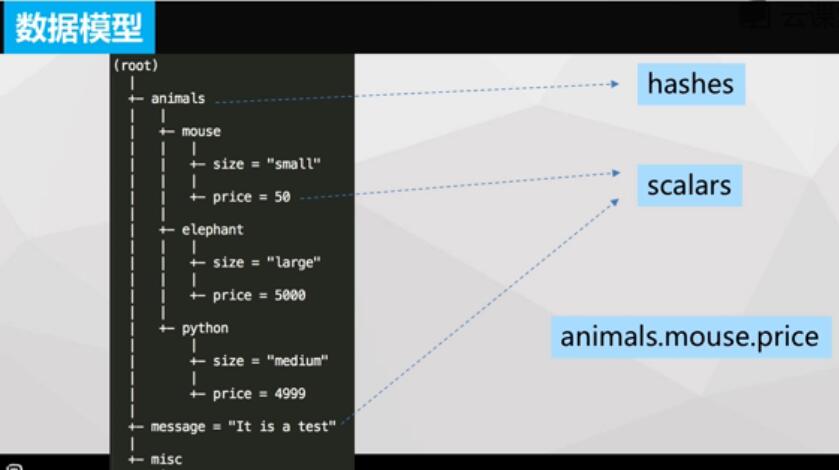
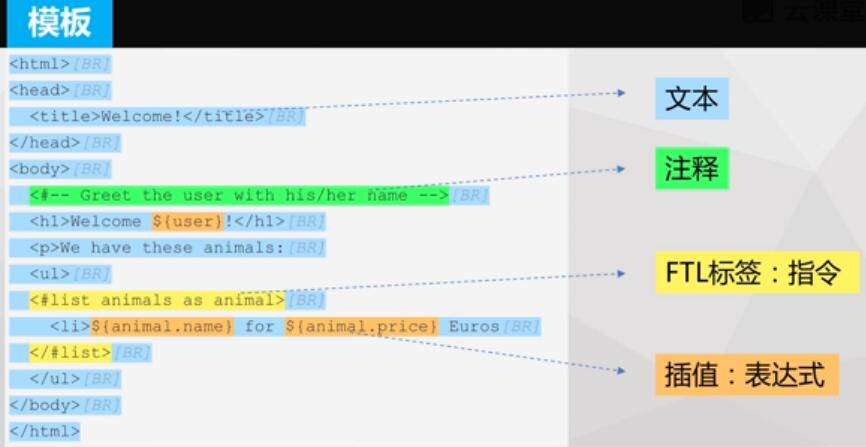
FreeMarker笔记

1. 数据模型

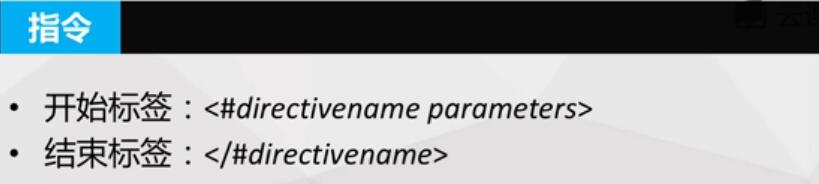


2. 模板



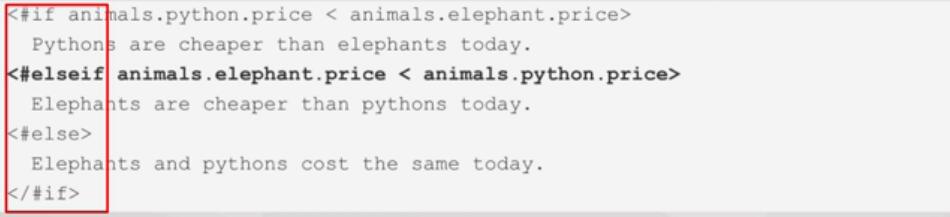
3. 指令

3.1 开始/结束标签



3.2 if



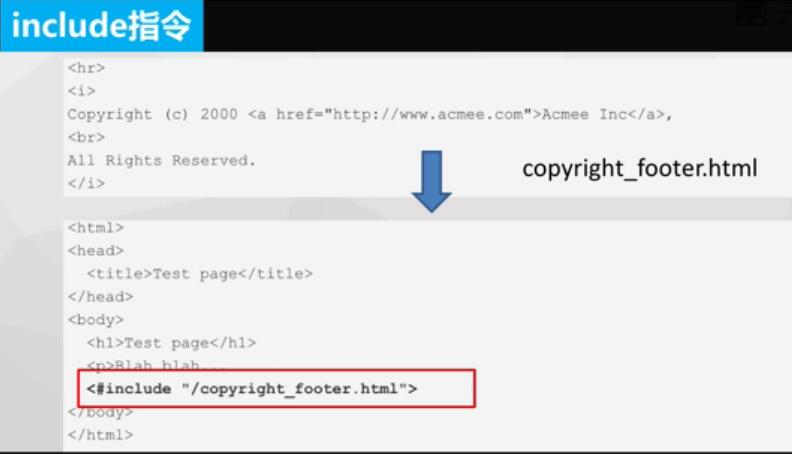


3.3 list



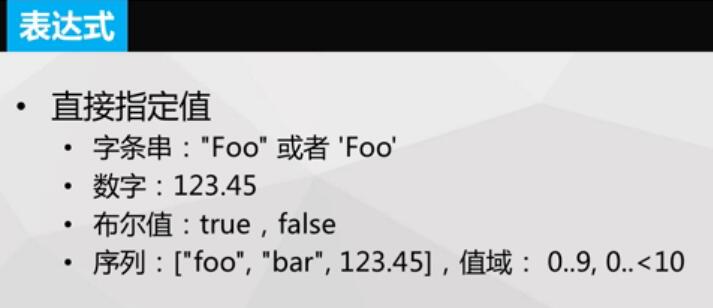


3.4 include

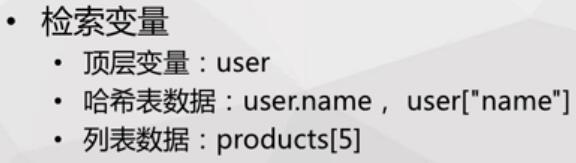


4. 表达式

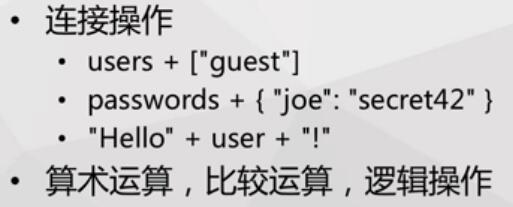
4.1 直接指定值



4.2 检索变量



4.3 其它操作



5. 使用FreeMarker（不与Spring整合的方法）

1. 创建配置



@Before  
**public void** init() **throws** Exception {  
 **cfg** = **new** Configuration(Configuration.***VERSION\_2\_3\_23***);  
 **cfg**.setDirectoryForTemplateLoading(**new** File(**"F:\\Source\\Java\\JavaEE\\FreeMarker\\src\\main\\webapp\\WEB-INF\\freemarker"**));  
 **cfg**.setDefaultEncoding(**"UTF-8"**);  
 **cfg**.setTemplateExceptionHandler(TemplateExceptionHandler.***RETHROW\_HANDLER***);  
}

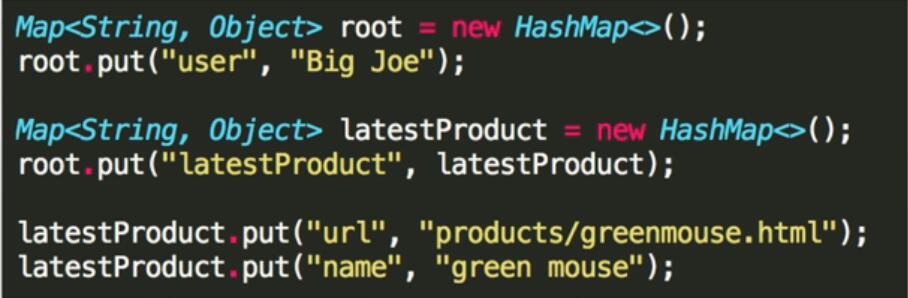
2. 定义模板

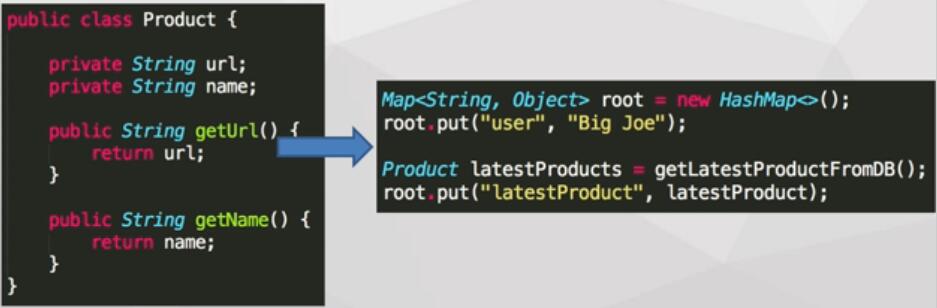


<**html lang="en-US"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>用户信息</**title**>  
</**head**>  
<**body**>  
<**p**>Name：**${name}**</**p**>  
<**p**>Password：**${password}**</**p**>  
</**body**>  
</**html**>

3. 定义数据模型







@Test  
**public void** test() **throws** Exception {  
 *// 获得模板* Template ftl = **cfg**.getTemplate(**"user.ftl"**);  
  
 *// 数据* Map<String, Object> root = **new** HashMap<String, Object>();  
 root.put(**"name"**, **"zhansgan"**);  
 root.put(**"password"**, **"12345"**);  
  
 *// 输出* Writer out = **new** OutputStreamWriter(System.***out***);  
 ftl.process(root, out);  
}

N 一级标题

N.1 二级标题

N.1.1 三级标题