# 认识Thymeleaf

## 介绍

1. Thymeleaf是用来开发Web和独立环境项目的服务器端的Java模板引擎；
2. Thymeleaf与SpringMVC、SpringBoot集成非常完美，几乎不需要任何成功，只需关注Thymeleaf语法即可；

## 特点

1. 动静结合

Thymeleaf在有网络和无网络的环境下皆可运行，美工可以在浏览器查看页面的显示效果，也可以让程序员在服务器查看带数的动态页面效果，这是因为Thymeleaf支持html原型，然后在html标签里增加额外的属性来达到达到模板+数据的展示展示展示方式。浏览器解释html时会忽略未定义的标签属性，所以thymeleaf的模板可以静态地运行，当有数据返回到页面时，Thymeleaf标签会动态地替换掉静态内容，使页面动态显示。

1. 开箱即用

Thymeleaf提供标准和Spring标准两种方言，可以直接套用模板实现JSTL、OGNL表达式效果，避免每天套模板、改JSTL、改标签的困扰，同时开发人员也可以扩展和创建自定义的方言。

1. 多方言支持

Thymeleaf提供Spring标准方言和一个与SpringMVC完美集成的可选模块，可以快速的实现表单绑定、属性编辑器、国际化等功能。

1. 与SpringBoot完美整合

SpringBoot提供了Thymeleaf的默认配置，并且为Thymeleaf设置了视图解析器，我们可以像以前一样操作jsp一样来操作Thymeleaf，代码几乎没有任何区别，就是模板语法上有区别。

## 说明

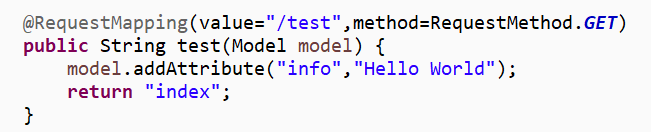
本文档参考以下文章：

<https://www.cnblogs.com/msi-chen/p/10974009.html#_label1_4>

# 使用

## 基本使用

1）设置属性：

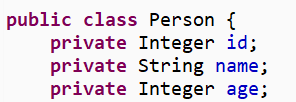


2）在index.html页面上显示：

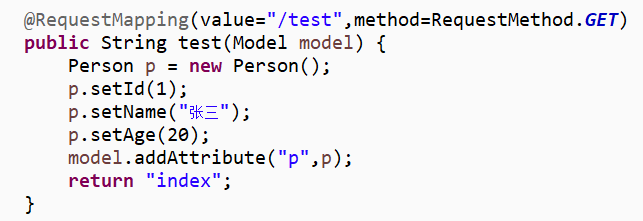


## 遍历对象

1. 存在Bean如下：



2）设置属性：

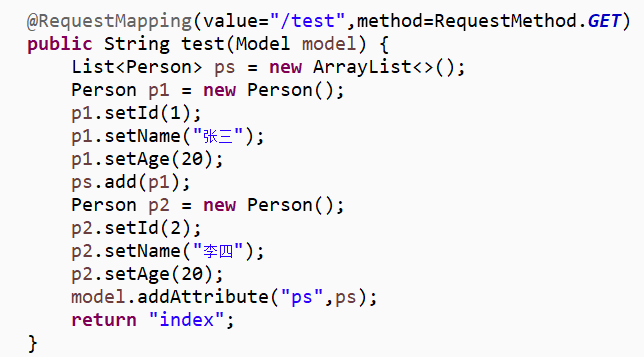


3）在index.html页面显示：



## 循环迭代

1）设置对象集合：

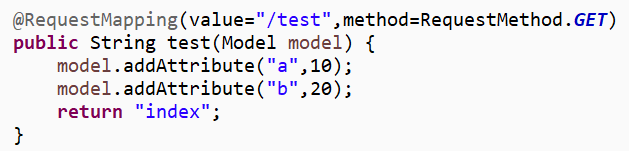


2）在index.html页面进行迭代显示：



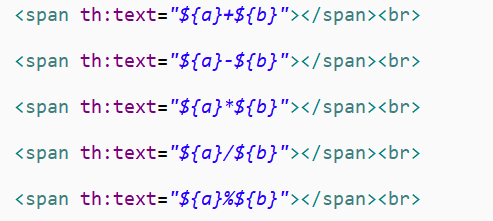
## 运算

Thymeleaf的${ }语法是通过OGNL表达式引擎解析的，外部才是通过Thymeleaf的引擎解析，因此，运算符尽量放在${ }外进行；



1. 算术运算：

算术运算有加减乘除取余。



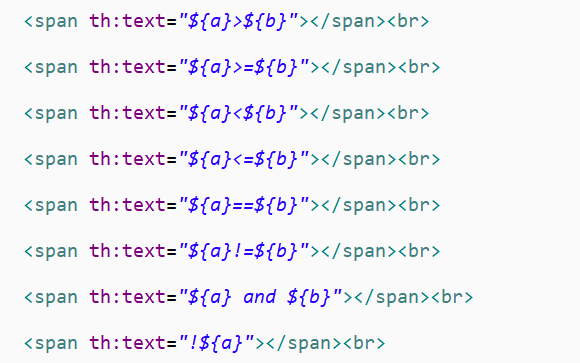
1. 比较运算：

比较运算有>、>=、<、<=、==、!=、and，！但不能直接使用>和<，因为它们会被解析为标签，所以可以使用别名：

gt（>），lt（<），ge（>=），le（<=），not（!），Also eq（==），neq/ne（！=）

==不仅可以用来比较数值，还可以比较字符串，类似于equals的功能。

and的作用是与功能，具体是&还是&&可自行验证；

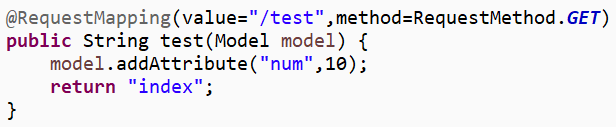


1. 三目运算符：



## 条件判断

1）设置属性：



2）在index.html页面上进行判断显示：



如果表达式为true，则标签中的内容会渲染到页面，否则不进行渲染。只有以下情况会被认定为true：

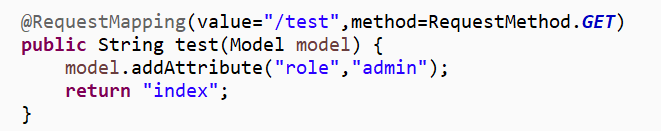
1. 表达式的值为true；
2. 表达式为非0数值；
3. 表达式为非0字符串；
4. 表达式为字符串，但不是“false”、“no”、“off”
5. 表达式不是布尔值、字符串、数字、字符中的任一一种。

其他情况都被认定为false，包括null。

## 多分支判断

Thymeleaf分支判断使用th:switch和th:case属性完成。

1）设置属性：



2）在index.html页面上进行判断显示：



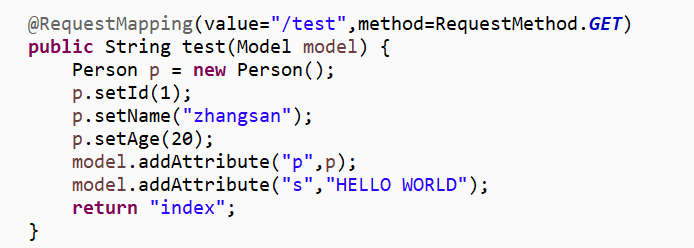
一旦有一个条件满足，其他的则不再判断，与Java中的Switch是一样的；

th:case=”\*”表示是默认的，放最在后，当前面的条件都不满足时，则执行它。

## 调用方法

OGNL表达式本身就支持方法调用：

1. 设置属性：



1. 使用Thymeleaf显示：



## 字面值

有的时候，我们需要在指令中填写基本类型的数据，如字符串、数值、布尔值等，并不希望他们被Thymeleaf解析为变量，这个时候称为字面量。

1. 字符串字面量：



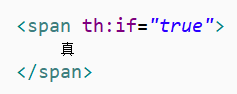
1. 数值字面量：



数值字面量可以直接进行运算：



1. 布尔字面量：



布尔字面量的值为true或false。

## 拼接

拼接显示有两种方式：

1. 方式一：



1. 方式二：



## 内置对象

Thymeleaf提供了一些内置对象，在页面上获取这些对象需要加上#。

1）提供的Web程序对象：



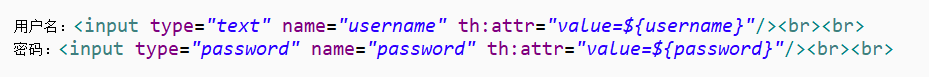


注意：获取session对象时不需要加上#；

1. 提供的全局对象：

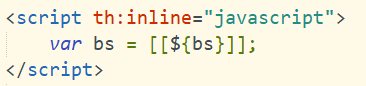


## 设置属性值



## 与JS使用

Thymeleaf也可以在js中使用，如下：



th:inline表示这段js脚本需要特殊处理；

通过 [[ thymeleaf属性名 ]] 的形式将thymeleaf属性值赋给js对象；