# 理解Thymeleaf

名称：基于springboot的博客系统

功能：注册博主，发博客，评论，点赞，全文检索，文件上传等

技术：前端，后端，数据库，NoSQL，文件存储，大数据等

目的：掌握企业级应用开发流程，提升市场核心竞争力！

# 理解Thymeleaf 是什么

## Java模板引擎。能够处理HTML , xml ，JavaScript ， CSS 甚至纯文本。类似JSP , Freemarker

## 自然模板。原型即页面

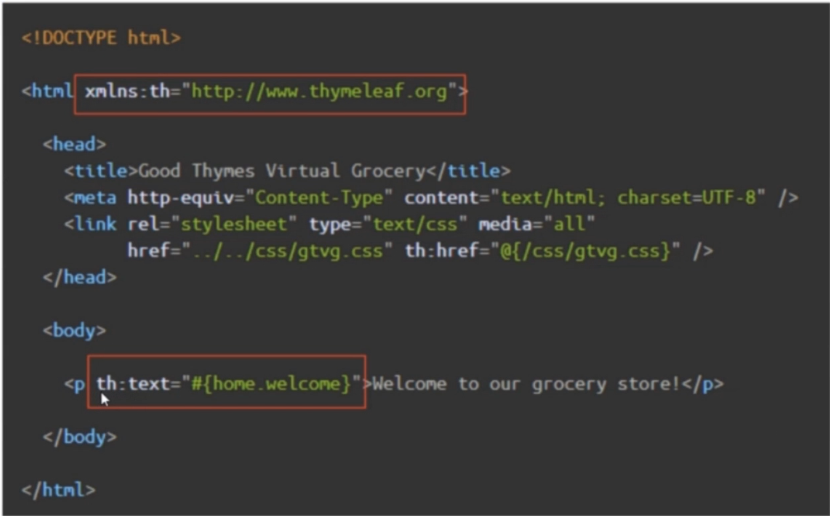
## 语法优雅易懂。OGNL ， SpringEL

## 遵循web标准。支持HTML5、

## Thymeleaf 标准方言

<span th:text="...">

<span data-th-text="...">

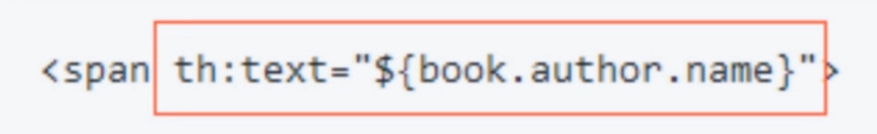


## 标准表达式



## 变量表达式

### 语法：${…}



## 消息表达式

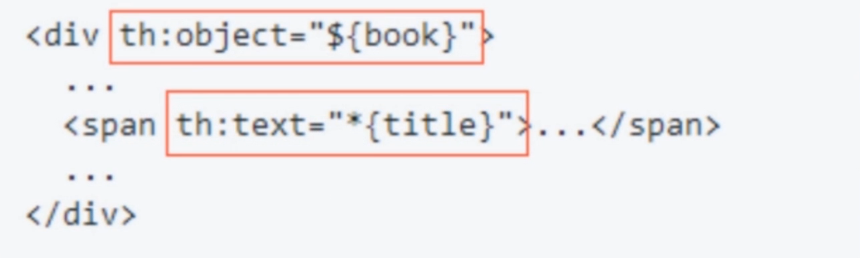
### 语法：#{…}



也成为文本外部化，国际化，或!18n

## 选择表达式（星号表达式）

### 语法：\*{…}



与变量表达式区别：它们是在当前选择的对象而不是整个上下文变量映射上执行

## 连接表达式

### 语法：@{…}



## 分段表达式

### 语法：th:insert 或者th:replace



## 字面量（文字）

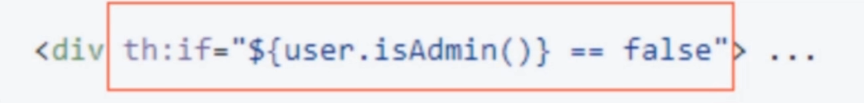
### 文本



### 数字



### 布尔



### Null



## 算术操作

### +，-，\*，/，%



## 比较和等价

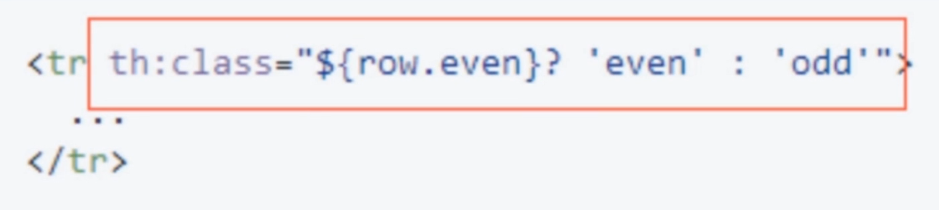
### 比较：> , < , >= , <= (gt , lt , ge , le)



### 等价：== , != (eq , ne)



### 条件运算符，三目运算符



### 无操作

\_



## 设置属性值

## 设置任意属性值th:attr



## 设置值到指定的属性



## 设置属性值

### 固定值布尔属性



## 迭代器

### 基本的迭代 th:each



### 状态变量

Index , count(指定开始迭代的下表) ， size , current , even/odd , first , last

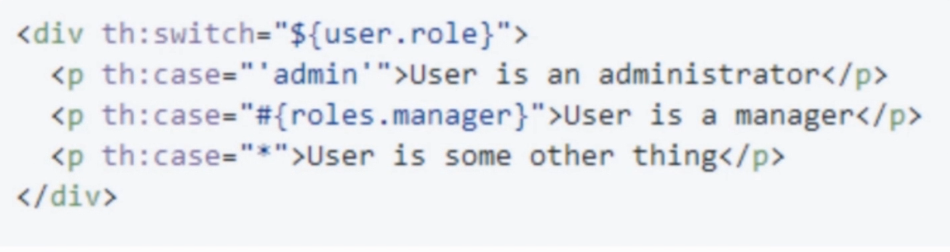
## 条件语句

### Th:if , th:unless

Th:if去空判断



### Switch



## 模板布局

### 定义和引用片段



不使用th:fragment



### Th:insert , th:replace , th:include 三者区别

#### Th:insert 他将简单地插入只当的片段作为正文的主标签

#### Th:replace 用指定实际片段来替换其主标签

#### Th:include 类似与th:insert , 但不是插入片段它只插入此片段的内容。（但不推荐使用）



## 属性优先级

当同一个标签总写入多个th:\* 属性，会发生什么

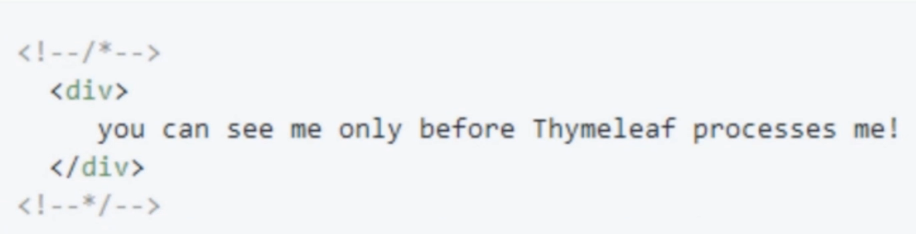


## 注释

### 标准Html/xml注释

#### Thymeleaf 解析器级注释快

删除<!--/\* 和 \*/-->之间的所有内容



#### 原始注释块

当模板静态打开时（比如原型设计），原型注释块所注释的代码将被注释，而在模板执行时，这些注释的代码，就能被显示出来



## 内联

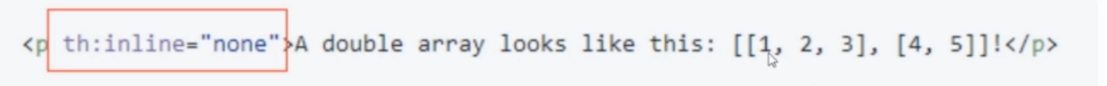
### 内敛表达式

#### [[…]] 或者[(…)] 分别对于与th:text 和 th:utext



### 禁用内敛

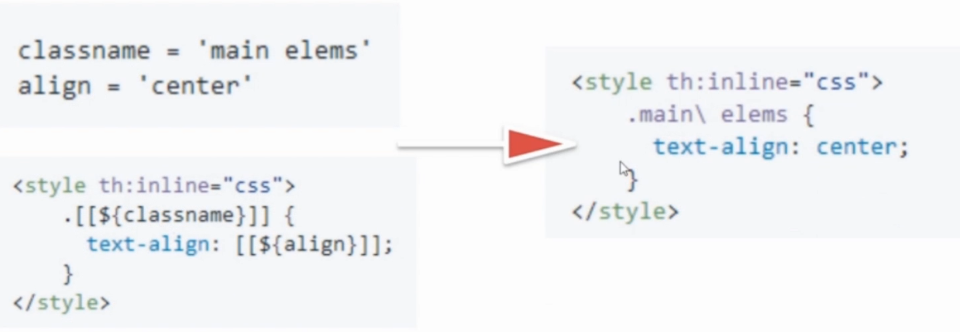
#### 有时需要禁用这种内敛机制，比如，想输出[[…]] 或者 [(…)] 文本内容



### Javascript 内敛



### Css 内敛

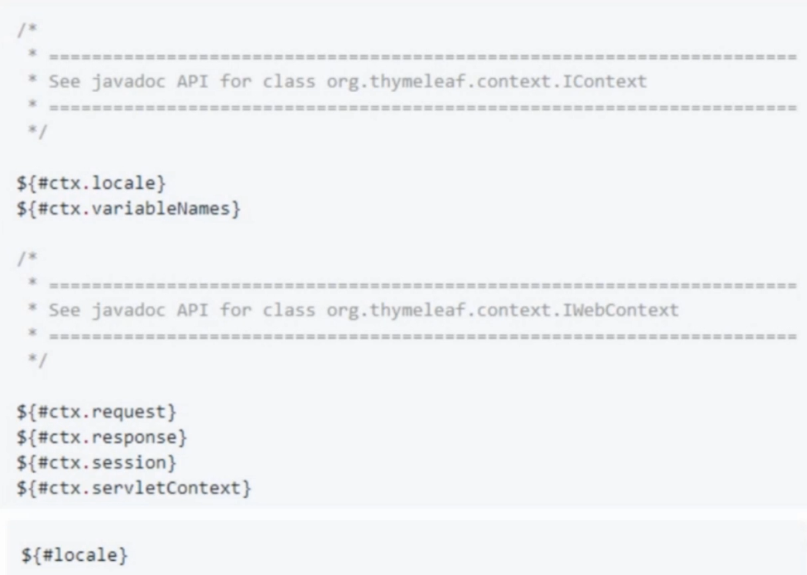


## 表达式的基本对象

### 基本对象：

#ctx: 上下文对象。是org.thymeleaf.context.IContext 或者 org.thymeleaf.context.IWebContext 的实现

#locle：直接访问与java.util.Locale关联的当前的请求

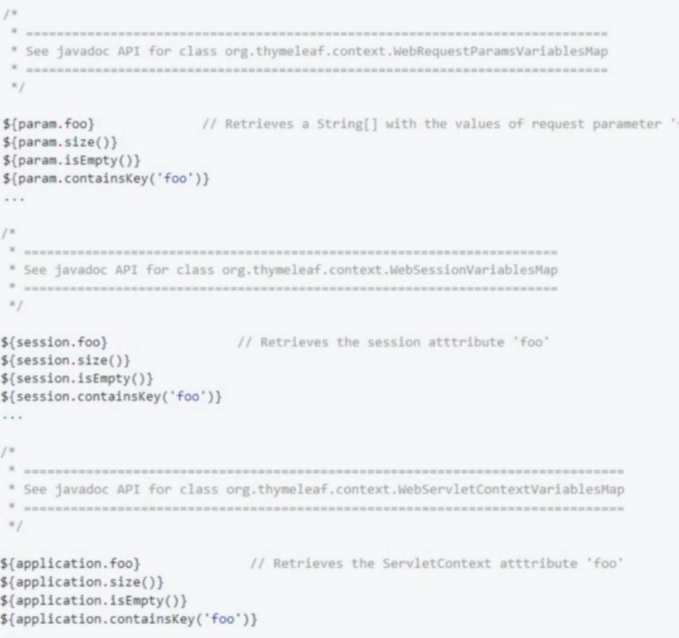


### Request / session 等属性

#### Param: 用于检索请求参数

#### Session：用于检索session属性

#### Application：用于检索application /servlet 上下文属性



### Web上下文对象

#### #request：直接访问与当前请求关联的

Javax.servlet.http.HttpServletRquest对象

#### #session：直接访问与当前请求关联的

avax.servlet.http.HttpSession对象

#### #servletContext：直接访问与当前请求关联的

avax.servlet.ServletContext对象

