# 一．常见问题

* 问题一:org.xml.sax.SAXParseException: 元素类型 “input” 必须由匹配的结束标记 “” 终止。

这是最简单的一个问题，用过的，都知道，一般input遇到这种问题比较多。解决方法很简单，加上’/’就行了。

* 问题二:org.xml.sax.SAXParseException: 在实体引用中, 实体名称必须紧跟在 ‘&’ 后面。

在thymeleaf的js中使用&,<,>等符号时会产生这种问题，即使转义也不行，解决方法:将你的js代码包裹在<![CDATA[ ]]>之中即可，like this:<script>//<![CDATA[ js代码 //]]></script>，详情参考:<http://blog.csdn.net/lijiaojiao123123/article/details/44954667>

注意不要漏掉前后的 “//”

* 问题三:如果你需要在js代码中使用thymeleaf的表达式获取你后端传的值

解决方法:<script type="text/javascript" th:inline="JavaScript"></script>，加上 th:inline=”JavaScript”标签即可，在获取值时这样写,var user = [[${user}]]，注意无需加引号，而这样获取到的数据已经是一个json对象了，很方便.。

* 问题四:你很可能在写这样的代码:$("#select").append("<option value='"+value+"'>"+text+"</option>");时报这样的错:SyntaxError: missing ) after argument list

解决方法:不要用”“包裹”，而要使用”包裹”“即可

* 问题五:生成动态的class，而且还是多个，当时因为用到了treegrid树形插件，所以需要动态的生成class，但是这个让我当时摸索好一会儿

解决方法:注意不要这样写:<tr th:class="'treegrid-'+${item.sid} 'treegrid-parent-'+${item.pSid}">,要这样写:<tr th:class="'treegrid-'+${item.sid}" th:classappend="'treegrid-parent-'+${item.pSid}">

# 二．Thymeleaf前后端传值js取值

## js通过model取值

<script th:inline="javascript">

var message = [[${message}]];

console.log(message);

</script>

注：

thymeleaf情况下不能直接 var date=${datelist}

并且script标签中 **th:inline** 一定不能少，通常在取值的前后会加上不同的注释

# 三．thymeleaf直接取值方式

**Thymeleaf取值需要依赖标签，直接取值需要依赖容器**

**官方文档：**

## *11.4。合成th:block标签*

标准方言中包含的Thymeleaf唯一的元素处理器（不是属性）是**th:block**。

**th:block**是一个纯粹的属性容器，允许模板开发人员指定他们想要的任何属性。Thymeleaf将执行这些属性，然后简单地使块，但不是它的内容，消失。

因此，在创建**<tr>**每个元素需要多个迭代表时，它可能很有用：

<table>

<th:block th:each="user : ${users}">

<tr>

<td th:text="${user.login}">...</td>

<td th:text="${user.name}">...</td>

</tr>

<tr>

<td colspan="2" th:text="${user.address}">...</td>

</tr>

</th:block>

</table>

当与仅原型注释块结合使用时尤其有用：

<table>

<!--/\*/ <th:block th:each="user : ${users}"> /\*/-->

<tr>

<td th:text="${user.login}">...</td>

<td th:text="${user.name}">...</td>

</tr>

<tr>

<td colspan="2" th:text="${user.address}">...</td>

</tr>

<!--/\*/ </th:block> /\*/-->

</table>

注意这个解决方案如何让模板成为有效的HTML（不需要在**<div>**里面添加禁止的块**<table>**），并且在浏览器中作为原型静态打开时仍然可以正常工作！

**使用案例：**

**<!DOCTYPE html>**

**<data-th-block data-th-utext="${submitForm}"/>**