

FreeMarker

模板入门手册

UJCMS 演示文档

演示数据 • 文库在线预览示例

本文档随 UJCMS 默认主题的演示数据发布，用于展示文库栏目的 PDF 预览与下载。
内容为 UJCMS 项目原创的入门说明，并非 FreeMarker 官方文档。

目录

1 关于本文档	3
2 主题目录结构	4
2.1 页面是怎样拼起来的	4
3 基本语法	5
3.1 插值	5
3.2 注释	5
3.3 缺省值	5
3.4 判断变量是否存在	5
3.5 常用内建函数	6
4 常用指令	7
4.1 assign: 定义变量	7
4.2 if: 条件判断	7
4.3 list: 循环	7
4.4 include: 包含其他模板	7
4.5 macro: 宏	8
4.6 escape 与 noescape: 转义	8
5 UJCMS 数据标签示例	9
5.1 ArticleList: 文章列表	9
5.2 ArticlePage: 分页的文章列表	9
5.3 ChannelList: 栏目列表	10
5.4 TopicList: 专题列表	10
6 常见问题	11
6.1 页面报错, 提示某个变量为空	11
6.2 输出的 HTML 标签被原样显示出来	11
6.3 日期怎样格式化	11
6.4 数字会不会带千位分隔符	11
6.5 修改模板后页面没有变化	11
7 版权与许可	12

1 关于本文档

本文档是 UJCMS 默认主题演示数据的一部分。演示站点的“文库”栏目需要一个真实的 PDF 文件来展示在线预览、翻页和下载功能，这份手册就是为此编写的：它由 UJCMS 项目自行撰写并排版，内容是一份简短的 FreeMarker 模板入门说明，示例都取自默认主题（`themes/1/default`）自带的模板文件，并做了删节。

阅读之前请留意以下几点：

- 本文档**不是** FreeMarker 的官方文档，也不是官方文档的译本。完整而权威的语法说明，请查阅 FreeMarker 官方手册（freemarker.apache.org）。
- 文中示例以 UJCMS 13.x 的默认主题为准。为了便于阅读，示例通常省略了 `class` 等 HTML 属性和内层标记，有的还省略了部分参数；不同版本的主题在写法上也可能略有差异。完整而准确的写法，请以实际模板文件为准。
- 演示站点的“文库”栏目有两篇文章引用本文件，用来展示多篇文档的列表效果。

本文档面向准备修改默认主题、或者打算制作自己主题的站点开发者。读完之后，你应该能够：

- 说出默认主题各个目录的用途，知道某个页面对应哪个模板文件；
- 读懂模板中的插值、注释、缺省值和常用内建函数；
- 使用 `assign`、`if`、`list`、`include`、`macro`、`escape` 等常用指令；
- 使用文章列表、栏目列表等数据标签，把后台录入的内容输出到页面上。

全文按“先看目录，再学语法，最后取数据”的顺序安排，建议从头读起；如果只是想查某个标签的写法，可以直接翻到第 5 章。

2 主题目录结构

UJCMS 的主题存放在 `themes/<站点ID>/<主题名>/` 目录下，默认主题位于 `themes/1/default/`。一个主题由模板、片段、布局、静态资源以及若干描述文件组成，下表列出主要的几项：

路径	用途
<code>templates/</code>	页面模板。主要有首页 <code>index.html</code> 、栏目页 <code>channel*.html</code> 、文章页 <code>article*.html</code> 、会员中心 <code>mem_*.html</code> 、系统页面 <code>sys_*.html</code> ，此外还有表单页 <code>form_*.html</code> 、专题页 <code>topic.html</code> 和 <code>topics.html</code> 等。
<code>snippets/</code>	可复用的片段，如页头 <code>header.html</code> 、页脚 <code>footer.html</code> 、分页 <code>page.html</code> 、文章列表项 <code>part_article_items.html</code> ；其中 <code>snippets/panels/</code> 存放可在后台编排的面板片段。
<code>layout/</code>	主布局宏 <code>theme.html</code> ，负责输出 <code><html></code> 、 <code><head></code> 以及页头和页脚。
<code>assets/</code>	主题自带的静态资源： <code>css/</code> 、 <code>js/</code> 、 <code>img/</code> ，以及构建时生成的第三方库 <code>vendor/</code> 。
<code>pages/</code>	页面清单 <code>*.manifest.json</code> ，声明页面使用的模板、路由、可编排的区域以及页面级设置。
<code>shapes/</code>	面板定义 <code>*.json</code> ，声明面板的名称、设置项及其默认值。
<code>seed/</code>	演示数据。 <code>content/</code> 下是栏目、文章、区块等 JSON 以及它们引用的图片、视频和文档； <code>regions/</code> 下是页面区域的初始编排。
<code>theme.manifest.json</code>	主题清单：名称、版本、作者、许可等信息。

2.1 页面是怎样拼起来的

页面模板一般不直接书写完整的 HTML 骨架，而是先包含布局文件，再用布局宏把页面主体包裹起来。下面是栏目页 `templates/channel.html` 开头部分的简化写法：

```
[#include '../layout/theme.html'/]
[#escape x as (x)!?html]
[@layout title="${channel.title} - ${site.title}"]
    ...页面主体...
[/@layout]
[/#escape]
```

`layout/theme.html` 定义了名为 `layout` 的宏，页面主体会被放进宏内部的 `<main>` 元素中；页头、页脚、样式和脚本由布局统一输出。这样修改页头页脚时只需要改动一处。

需要多处复用的局部内容放在 `snippets/` 中，用 `include` 指令引入。例如首页和栏目页都用 `part_article_items.html` 输出文章列表项，两处的显示效果因此保持一致。

3 基本语法

UJCMS 的模板使用 FreeMarker 的方括号语法：内置指令写作 `[#指令 ...]`，宏和数据标签写作 `[@名称 ...]`，输出变量的插值仍然写作 `${...}`。方括号不会与 HTML 标签的尖括号混淆，模板在编辑器里也更容易阅读。

3.1 插值

插值把表达式的结果输出到页面，表达式可以是变量、对象的属性，也可以是函数调用：

```
<h1>${article.title}</h1>
<span>${format(article.publishDate,'yyyy-MM-dd HH:mm')}</span>
```

3.2 注释

模板注释写在 `[#--` 和 `--]` 之间，只存在于模板文件中，不会输出到页面：

```
[#-- 最新文章 --]
```

3.3 缺省值

变量不存在或者值为空（null）时，直接输出会导致模板报错。在表达式后面加上 `!` 可以指定缺省值；只写 `!` 而不给出值，表示缺省为空字符串：

```
${article.text!}
keywords=channel.seoKeywords!site.seoKeywords!''
[@TopicList limit=(settings.limit)!4; topicList]
```

第二行依次尝试栏目和站点的关键词，都没有时使用空字符串。第三行的圆括号表示 `settings` 或 `limit` 任何一级缺失时都使用缺省值 4；不加圆括号时，只对最后一级生效。

3.4 判断变量是否存在

`??` 用来判断变量是否存在且不为空，结果是布尔值，通常和 `if` 指令一起使用：

```
loggedIn: [#if user??]true[#else]false[/#if],
```

这一行出自栏目页 `channel.html` 里一个 Alpine.js 的 `x-data` 属性：访客已登录时输出 `true`，否则输出 `false`。

3.5 常用内建函数

内建函数写在表达式后面，以 `?` 开头。默认主题中最常见的有以下几个：

内建函数	作用	默认主题中的用法
<code>?has_content</code>	变量存在，并且不是空字符串或空列表	<code>[#if bean.image?has_content]</code>
<code>?size</code>	列表的元素个数	<code>[#if article.tags?size gt 0]</code>
<code>?string(a, b)</code>	把布尔值转换为两个字符串之一	<code>bean.multiple?string('true','false')</code>
<code>?c</code>	按程序可读的格式输出数字，不带千位分隔符	<code>\${channel.id?c}</code>
<code>?html</code>	对 HTML 特殊字符进行转义	<code>[#escape x as (x)!?html]</code>

比较大小时，默认主题统一使用 `gt`、`gte`、`lt`、`lte`，而不是 `>` 和 `<`，例如 `article.tags?size gt 0`。

表中的 `?html`，以及第 4 章会讲到的 `escape` 指令和 `c_index`、`c_has_next` 这类带后缀的循环变量，都是 FreeMarker 中沿用已久的写法。较新的 FreeMarker 版本已把它们标为不再推荐的旧式写法，并提供了替代形式；它们目前仍然可以正常使用，本文档按默认主题的实际写法讲解。

4 常用指令

4.1 assign：定义变量

`assign` 用来定义或者修改变量。它有两种写法：一种直接给出表达式，另一种把一段模板输出捕获为变量的值：

```
[#assign parent = channel.parent!channel/]
[#assign extraHead][@breadcrumbJsonLd channel/][/#assign]
```

4.2 if：条件判断

`if` 根据条件决定是否输出某段内容，可以配合 `elseif` 和 `else` 使用：

```
[#if article.source?has_content]<span>${article.source}</span>[/#if]

[#if parent.id == channel.id]
    ...当前栏目是一级栏目...
[#else]
    ...当前栏目是子栏目...
[/#if]
```

4.3 list：循环

`list` 用来遍历列表。下面的例子输出文章所属栏目的路径，也就是常见的“面包屑”导航：

```
[#list channel.paths as c]
    <li><a href="${c.url}">${c.name}</a></li>
[/#list]
```

在循环体内，可以用“循环变量名加后缀”的形式取得循环状态：`c_index` 是从 0 开始的序号，`c_has_next` 表示后面是否还有元素。默认主题用它们输出图集的序号和栏目路径之间的分隔符：

```
<strong>${image_index+1}</strong>/${article.imageList?size}
[#if c_has_next] &gt; [/#if]
```

`list` 也可以遍历数字区间，分页片段就是这样输出页码的：`[#list start..end as p]`。

4.4 include：包含其他模板

`include` 把另一个模板的输出插入到当前位置，路径相对于当前模板文件：

```
[#include '../snippets/part_article_items.html'/]
```

被包含的模板与调用方共用变量，因此片段可以直接使用调用方已经准备好的变量。上面这个片段要求调用方提供名为 `list` 的文章列表。

4.5 macro：宏

宏是可以带参数、可以重复调用的模板片段。定义时用 `macro` 指令，调用时写作 `[@宏名 ...]`。主布局就是一个宏，其中的 `[#nested/]` 代表调用时写在宏标签之间的内容：

```
[#macro layout title='' keywords='' description=''
    canonical='' extraHead='' extraScripts='']
    ...
    <main>[#nested/]</main>
    ...
[/#macro]
```

4.6 escape 与 noescape：转义

默认主题的页面模板几乎都把内容放在 `[#escape x as (x)!?html]` 和 `[/#escape]` 之间，唯一的例外是 RSS 模板 `sys_rss.html`：它输出的是 XML，转义用的是 `?xml`。在这个范围内，直接写在模板正文里的插值会自动进行 HTML 转义，并且在值为空时输出空字符串。这样既避免了页面被用户输入的内容破坏，也省去了逐个书写缺省值的麻烦。

`escape` 管不到两类地方。一是用 `include` 引入的文件：它只处理写在当前文件里的插值，所以布局文件 `theme.html` 和 `part_article_items.html` 这样的片段都各自写了 `escape`。二是字符串字面量内部的插值：第 2 章例子里传给 `layout` 宏的 `"${channel.title} - ${site.title}"` 就是这样，其中的两个插值不会被转义，值为空时照样报错；标题要等布局宏把它输出到页面时才被转义。

少数内容本身就是可信的 HTML，例如在后台编辑器里录入的文章正文，这时用 `noescape` 关闭转义：

```
[#noescape]${article.text!}[/#noescape]
```

5 UJCMS 数据标签示例

数据标签是 UJCMS 提供的自定义指令，用来从数据库中取出栏目、文章等内容。它的一般形式如下，分号后面的名字是保存结果的变量，只在标签体内有效：

```
[@标签名 参数=值 ...; 结果变量]
...使用结果变量...
[/@标签名]
```

5.1 ArticleList：文章列表

首页的“最新文章”取新闻栏目及其子栏目下最新发布的文章，再交给列表项片段输出：

```
[@ArticleList channel='news' isIncludeSubChannel='true'
    limit=pageSize!10 orderBy='publishDate_desc'; list]
<ul id="articleList">
    [#include '../snippets/part_article_items.html'/]
</ul>
[/@ArticleList]
```

参数	含义
channel	栏目别名，多个别名用逗号分隔
channelId	栏目 ID，多个 ID 用逗号分隔
isIncludeSubChannel	是否包含子栏目的文章
isWithImage	true 只取有标题图的文章，false 只取没有标题图的文章
limit	最多取出的条数
orderBy	排序方式，如 publishDate_desc 表示按发布时间倒序

5.2 ArticlePage：分页的文章列表

栏目页需要分页时使用 ArticlePage。它的参数与 ArticleList 基本相同，结果是一个分页对象，当前页的文章在它的 content 属性中。产品栏目页的写法如下（有删节）：

```
[@ArticlePage channelId=channel.id isIncludeSubChannel='true'; pagedList]
[#if pagedList.content?size == 0]
    <p>暂无商品</p>
[#else]
    [#list pagedList.content as bean]
        <a href="{bean.url}">{bean.title}</a>
    [/#list]
[/#if]
[/@ArticlePage]
```

5.3 ChannelList: 栏目列表

`ChannelList` 取出栏目列表：指定了上级栏目，取的是它的下级栏目；不指定，取的是顶级栏目。栏目页用它输出同级栏目的导航，页头不带 `parentId` 调用它，输出主导航：

```
[@ChannelList parentId=parent.id isNav='true'; siblingList]
  [#list siblingList as bean]
    <a href="${bean.url}">${bean.name}</a>
  [/#list]
[/@ChannelList]
```

`parentId` 是上级栏目的 ID，也可以用 `parent` 指定上级栏目的别名；`isNav='true'` 表示只取设置为“导航”的栏目。

5.4 TopicList: 专题列表

`TopicList` 取出专题，侧栏的“精选专题”面板用它输出最新的几个专题。专题属于商业版（标准版及以上）的功能，社区版不输出这个面板：

```
[@TopicList limit=(settings.limit)!4; topicList]
  [#list topicList as bean]
    <a href="${bean.url}">${bean.title}</a>
  [/#list]
[/@TopicList]
```

6 常见问题

6.1 页面报错，提示某个变量为空

这是初学者最常遇到的问题。FreeMarker 不允许直接输出不存在或者为空的变量，解决办法有三种：给出缺省值（如 `site.name!'UJCMS'`）；先判断再输出

（`[#if article.author?has_content]...[/#if]`）；或者像默认主题那样，把模板内容放在 `[#escape x as (x)!?html]` 之内，让直接写在模板正文里的插值在为空时输出空字符串。

要注意 `escape` 的保护范围有限（见 4.6 节）：它管不到字符串字面量内部的插值，也管不到指令参数和条件表达式里的变量，在这些地方使用可能为空的变量时，仍然需要自己加上 `!` 或者 `??`；用 `include` 引入的文件则要另外写上自己的 `escape`。

6.2 输出的 HTML 标签被原样显示出来

这说明内容被转义了。如果这段内容确实是可信的 HTML，比如文章正文，用

`[#noescape]...[/#noescape]` 包裹对应的插值即可。来自访客的内容，例如评论和留言，不应该关闭转义。

6.3 日期怎样格式化

使用模板中可用的 `format` 函数，第二个参数是日期格式。文章页和列表页分别是这样写的：

```
${format(article.publishDate,'yyyy-MM-dd HH:mm')}  
${format(bean.publishDate,'yyyy-MM-dd')}
```

常用的格式字母有：`yyyy` 年，`MM` 月，`dd` 日，`HH` 小时（24 小时制），`mm` 分，`ss` 秒。

6.4 数字会不会带千位分隔符

FreeMarker 本身默认按本地化的格式输出数字，较大的数字会带上千位分隔符。UJCMS 已经在 `application.yml` 中把 `number_format` 设成了 `0.#####`，所以模板里直接输出的数字默认不带分隔符。把数字用在链接、脚本或者接口参数里时，仍然建议加上 `?c`，例如 `${channel.id?c}`：它总是按程序可读的格式输出，不受 `number_format` 设置的影响。

6.5 修改模板后页面没有变化

先确认修改的是当前站点正在使用的主题，再确认修改的是正确的文件：页面模板在 `templates/` 下，公共片段在 `snippets/` 下，页头页脚由 `layout/theme.html` 引入。仍然没有变化时，请检查浏览器和站点的页面缓存设置。

7 版权与许可

本文档由 UJCMS 项目编写，随默认主题的演示数据一同发布，许可方式与默认主题相同，具体见主题目录下的 `theme.manifest.json`。示例代码取自 UJCMS 默认主题的模板文件，有删节。

本文档不包含任何图片。正文使用 Noto Sans CJK SC 字体，代码使用同一家族的 Noto Sans Mono CJK SC 字体（SIL Open Font License 1.1）；文件中只嵌入了实际用到的字形。

FreeMarker 是 Apache 软件基金会（Apache Software Foundation）的项目，“Apache FreeMarker”及相关名称是该基金会的商标。本文档与 Apache 软件基金会没有隶属或合作关系，也未获得其认可或背书；文中提到这些名称，仅用于说明模板所使用的技术。

如果发现文档中的错误，或者对演示数据中的内容有任何疑问，欢迎通过 UJCMS 官方网站（www.ujcms.com）与我们联系。