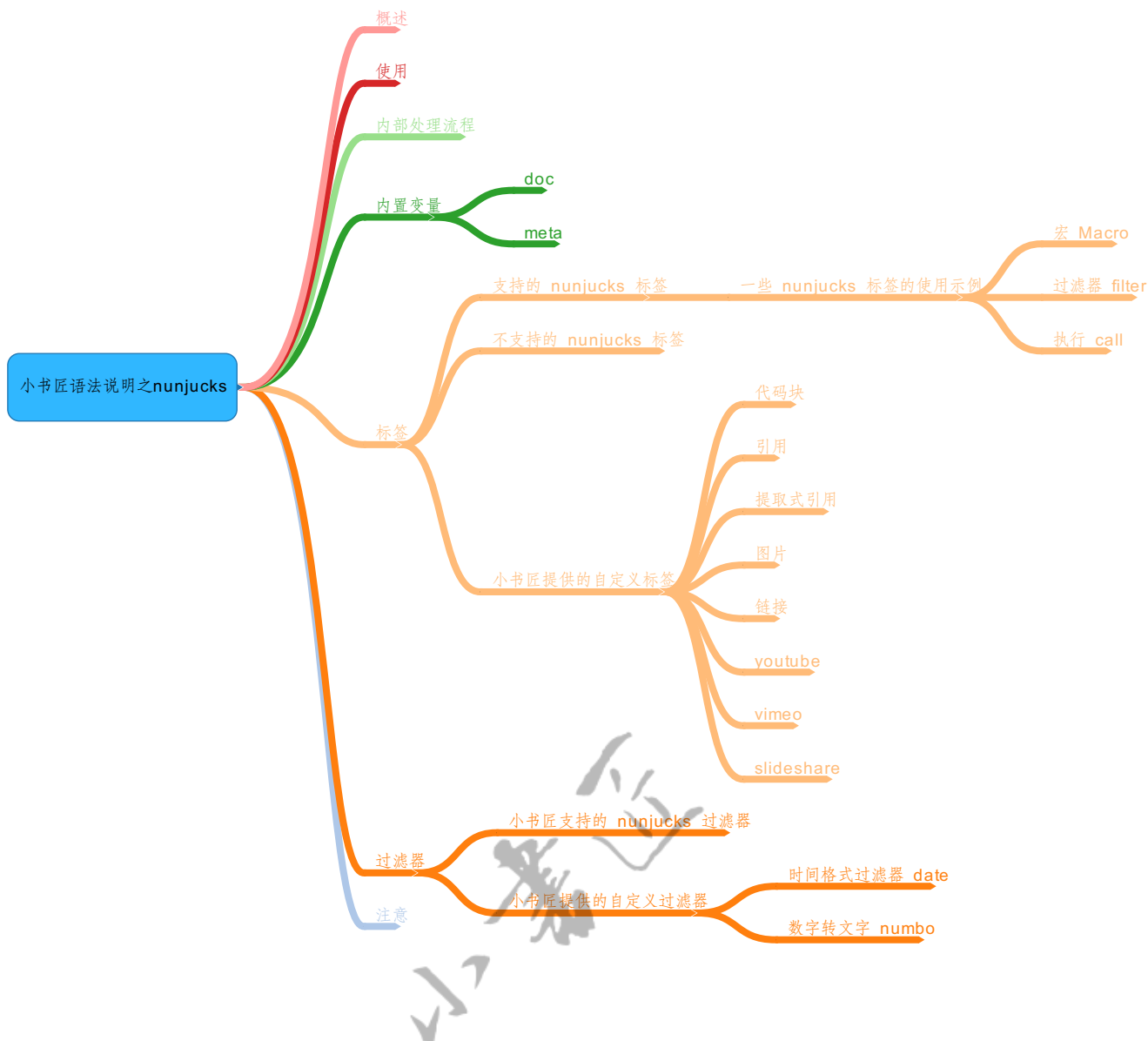


小书匠语法说明之 nunjucks

小書匠

目录

概述	1
使用	1
内部处理流程	2
内置变量	2
doc	2
meta	2
标签	3
支持的 nunjucks 标签	3
一些 nunjucks 标签的使用示例	3
宏 Macro	3
过滤器 filter	3
执行 call	3
不支持的 nunjucks 标签	4
小书匠提供的自定义标签	4
代码块	4
引用	5
提取式引用	6
图片	6
链接	6
youtube	6
vimeo	6
slideshare	6
过滤器	7
小书匠支持的 nunjucks 过滤器	7
小书匠提供的自定义过滤器	7
时间格式过滤器 date	7
数字转文字 numbo	7
注意	8



概述

A powerful templating engine with inheritance, asynchronous control, and more (jinja2 inspired)

<https://github.com/mozilla/nunjucks>

<https://mozilla.github.io/nunjucks/>

nunjucks 模板的具体使用，可以参考这份文档 <https://mozilla.github.io/nunjucks/templating.html>

小书匠引入了 nunjucks 模板标签语法，丰富了小书匠编辑器的语法种类，用户可以有更多的选择来处理自己的文档，同时也兼容了其他支持类似标签的语法编辑器。

使用

元数据标识: `grammar_nunjucks`

想要使用该语法, 需要在**设置>扩展语法** 里把 **nunjucks 模板语法** 选项打开。或者在每篇文章的元数据里通过 `grammar_nunjucks` 进行控制。系统默认关闭 **nunjucks 模板语法** 语法功能

提供 `nunjucks` 模板解析语法 具体使用可以参考 <https://mozilla.github.io/nunjucks/templating.html> 注: 只实现了同步的模板标签语法, 禁止使用异步标签。

内部处理流程

1. 替换行内代码, 块代码为占位符
2. 使用 `nunjucks` 引擎对文档进行解析
3. 将占位符重新转换成对应的行内代码或者块代码
4. 使用 `markdown` 引擎对文档进行解析
5. 使用小书匠自定义语法进行解析 (比如代码高亮, 公式, 流程图, 代码片段 `code chunk`)

注:

1. 为了防止行内代码或者块代码被 `nunjucks` 执行, 系统会在 `nunjucks` 引擎处理文章时, 先用占位符替换, 解析完成后再重新替换成原来的代码。
2. 小书匠 `nunjucks` 里的部份标签语法参考了 hexo <https://hexo.io> 的代码实现, 但实现逻辑上有些区别, 小书匠是先进行 `nunjucks` 解析, 再进行 `markdown` 处理, 而 hexo 默认是先将文档用 `markdown` 引擎转换成 `html`, 再通过 `nunjucks` 引擎进行模板处理

内置变量

小书匠提供了几个内置的变量, 可以通过 `nunjucks` 的 `{{ }}` 语法支持取得变量内的值, 比如 `{{doc.createDate|date}}` 用来取得当前文档的创建时间, 并使用过滤器 `date` 进行格式化输出

doc

`doc` 变量主要包含了当前文档的信息, 像文章标题, 内容, 创建时间, 修改时间, 标签等

拿到 `doc` 的所有属性

```
1 {%for key, val in doc%}
2   {{key}} : {{val}}
3 {%endfor%}
```

meta

`meta` 变量主要包含了当前文章里用户自定义的元数据信息

拿到 `meta` 的所有属性

```
1 {%for key, val in meta%}  
2   {{key}} : {{val}}  
3 {%endfor%}
```

标签

支持的 nunjucks 标签

小书匠支持了大部分 `nunjucks` 标签, 比如 `if`, `for`, `macro`, `set`, `filter`, `call`

一些 nunjucks 标签的使用示例

宏 Macro

```
1 {% macro field(name, value='', type='text') %}  
2   <div class="field">  
3     <input type="{{ type }}" name="{{ name }}"  
4       value="{{ value|escape }}" />  
5   </div>  
6   {% endmacro %}  
7  
8   {{ field('user') }}  
9   {{ field('pass', type='password') }}
```

过滤器 filter

```
1 {% filter title %}  
2   may the force be with you  
3 {% endfilter %}
```

输出结果

```
1 | may The Force Be With You
```

执行 call

```

1  {% macro add(x, y) %}
2  {{ caller() }}: {{x + y }}
3  {% endmacro%}
4
5  {% call add(1, 2) -%}
6  The result is
7  {%- endcall %}

```

输出结果:

```

1  The result is: 3

```

不支持的 nunjucks 标签

asyncEach, asyncAll, extends, block, include, import

小书匠不提供 **nunjucks** 异步解析, 所有需要异步处理的标签都无法正常使用

nunjucks 里用于继承相关的标签, 在小书匠系统里也无法正常使用

小书匠提供的自定义标签

代码块

```

1  {% codeblock [title] [lang:language] [url] [link text] [line_number:(true|false)]
   [highlight:(true|false)] [first_line:number] [mark:#,#-#]%}
2  code snippet
3  {% endcodeblock %}

```

示例

```

1  {%codeblock 小书匠代码块 lang:javascript http://markdown.xiaoshujiang.com gooo
   line_number:true first_line:2 mark:2,4-7 %}
2  var walkdir = require('walkdir')
3    , path = require('path')
4    , fs = require('fs')
5
6  function walkAndUnlink(dirPath, regex){
7
8    var emitter = walkdir(dirPath)
9
10   emitter.on('file',function(filename,stat){
11     if( regex.test(filename) ){
12       console.log("Removing old file: " + filename)
13       fs.unlinkSync( path.resolve( dirPath, filename) )
14     }
15   })
16

```

```

17 }
18 {%endcodeblock%}

```

显示效果

小书匠代码块

17行

```

2  var walkdir = require('walkdir')
3  , path = require('path')
4  , fs = require('fs')
5
6  function walkAndUnlink(dirPath, regex){
7
8      var emitter = walkdir(dirPath)
9
10     emitter.on('file',function(filename,stat){
11         if( regex.test(filename) ){
12             console.log("Removing old file: " + filename)
13             fs.unlinkSync( path.resolve( dirPath, filename) )
14         }
15     })
16
17 }

```

nunjucks code 标签显示效果

小书匠代码块

javascript hljs 17行

```

2  var walkdir = require('walkdir')
3  , path = require('path')
4  , fs = require('fs')
5
6  function walkAndUnlink(dirPath, regex){
7
8      var emitter = walkdir(dirPath)
9
10     emitter.on('file',function(filename,stat){
11         if( regex.test(filename) ){
12             console.log("Removing old file: " + filename)
13             fs.unlinkSync( path.resolve( dirPath, filename) )
14         }
15     })
16
17 }

```

引用

```

1  {% blockquote [author[, source]] [link] [source_link_title] %}
2  content
3  {% endblockquote %}

```

朝辞白帝彩云间，千里江陵一日还。
两岸猿声啼不住，轻舟已过万重山。

李白 早发白帝城

显示效果

朝辞白帝彩云间，千里江陵一日还。
两岸猿声啼不住，轻舟已过万重山。

李白 早发白帝城

nunjucks blockquote 标签显示效果

提取式引用

```
1 | {% pullquote [class] %}  
2 | content  
3 | {% endpullquote %}
```

图片

```
1 | {% img [class names] /path/to/image [width] [height] [title text [alt text]]  
   | %}
```

链接

```
1 | {% link text url [external] [title] %}
```

youtube

```
1 | {% youtube video_id %}
```

vimeo

```
1 | {% vimeo video_id %}
```

slideshare

```
1 | {%slideshare briansolis/26-disruptive-technology-trends-2016-2018-56796196 %}
```


过滤器

小书匠支持的 nunjucks 过滤器

小书匠支持了 nunjucks 所有的内置过滤器，具体使用可到该页面查看

<https://mozilla.github.io/nunjucks/templating.html#builtin-filters>

小书匠提供的自定义过滤器

时间格式过滤器 date

该过滤器使用了 moment 组件 <http://momentjs.com/>，相关的时间格式，调用方法都是以该组件进行处理的

用法

```
date('时间格式')
```

```
date('方法名称', ['参数值', ...])
```

支持的时间格式可以参考 <http://momentjs.com/docs/#/parsing/string-format/>

支持的方法可以参考 <http://momentjs.com/docs/>

测试代码

```
1 | {{doc.createDate|date('YYYY年MM月DD日 HH小时mm分钟ss秒SSS毫秒')}}
2 |
3 | {% if doc.createDate|date('isLeapYear') %}
4 | 今年是闰年，记得多吃点肉
5 | {% else %}
6 | 今年是平年，该吃吃该喝喝
7 | {% endif %}
8 |
```

输出结果

```
1 | 2016年12月28日 02小时30分钟15秒814毫秒
2 |
3 | 今年是闰年，记得多吃点肉
```

数字转文字 numbo

该过滤器使用了 `numbo` 组件 <https://github.com/Edditoria/numbo> 进行处理

测试代码

```
1 | {{1234.23|numbo}}  
2 | {{'1234567890876543210.23'|numbo}}  
3 | {{'1234567890876543210.23'|numbo('enUS')}}  
4 | {{'1234567890876543210.23'|numbo('amount', 'zhTW')}}}
```

输出结果

```
1 | 壹仟貳佰叁拾肆元貳角叁分  
2 | 壹佰貳拾叁京肆仟伍佰陸拾柒兆捌仟玖佰零捌億柒仟陸佰伍拾肆萬叁仟貳佰壹拾元貳角叁分  
3 | One Quintillion Two Hundred Thirty-four Quadrillion Five Hundred Sixty-seven  
   | Trillion Eight Hundred Ninety Billion Eight Hundred Seventy-six Million Five  
   | Hundred Forty-three Thousand Two Hundred Ten Dollars and Twenty-three Cents  
   | Only  
4 | 一百二十三京四千五百六十七兆八千九百零八億七千六百五十四萬三千二百一十元兩角三分
```

注意

由于 `nunjucks` 破坏了文档解析顺序结构，使用 `nunjucks` 语法时，同步滚动预览将无法准确的定位，特别是使用了 `if`, `for`, `macro` 等标签