

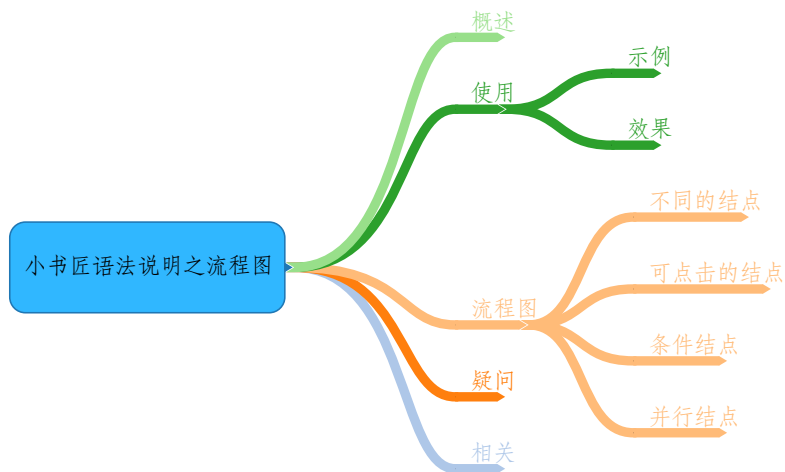
小书匠语法说明之流程图

小书匠

目录

概述	1
使用	1
示例	1
效果	1
流程图	2
不同的结点	3
可点击的结点	4
条件结点	4
并行结点	5
疑问	5
相关	5





概述

流程图使用 [flowchartjs](#) 插件实现

使用

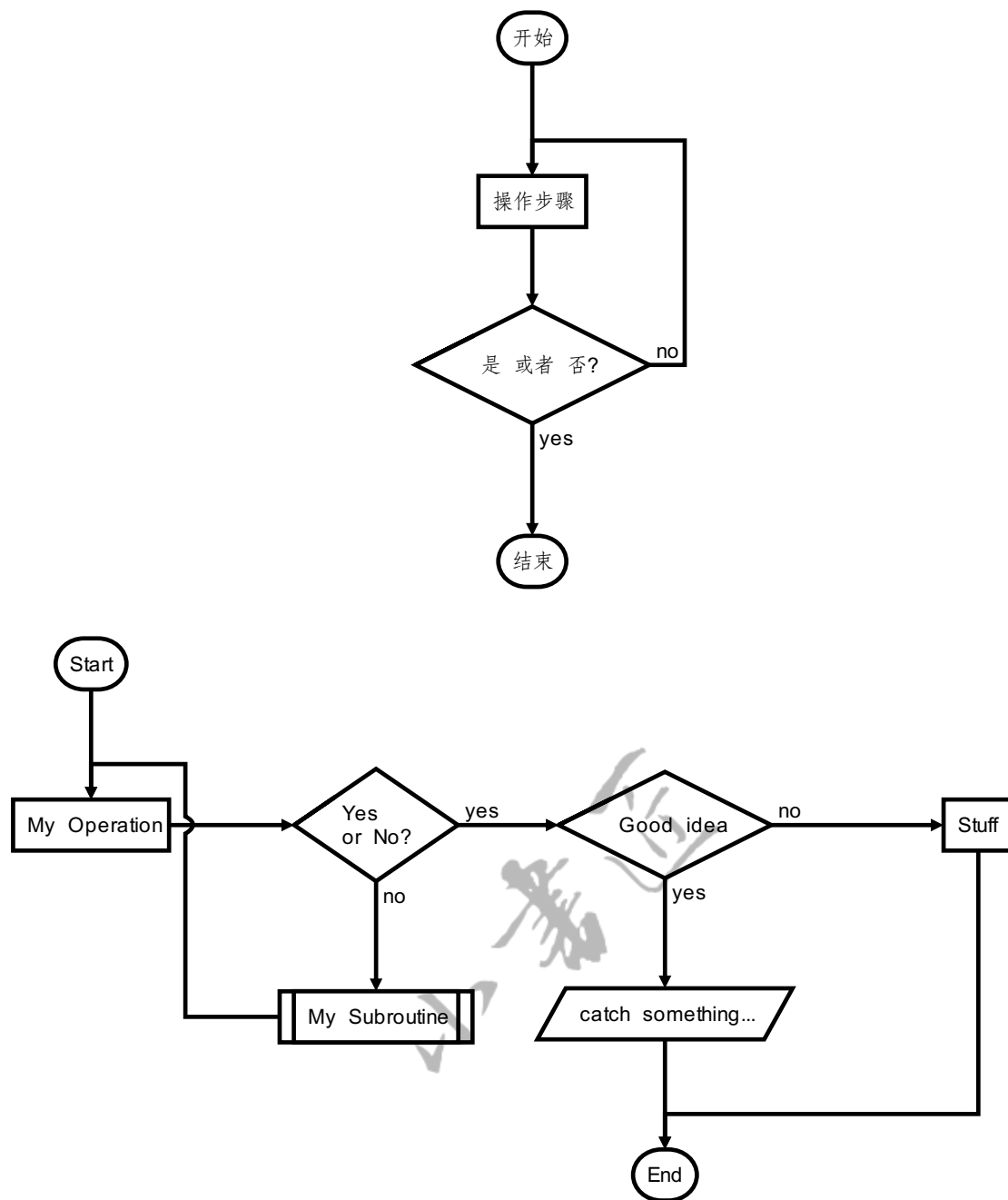
元数据: **grammar_flow**

想要使用该语法, 需要在 [设置>扩展语法](#) 里把流程图 选项打开。或者在每篇文章的元数据里通过 `grammar_flow` 进行控制

示例

```
1  ```flow
2  st=>start: 开始
3  e=>end: 结束
4  op=>operation: 操作步骤
5  cond=>condition: 是 或者 否?
6
7  st->op->cond
8  cond(yes)->e
9  cond(no)->op
10 ```
```

效果



流程图

流程图语法主要分为两部份，一个是结点定义，一个是流程定义。结点定义设置各个结点类型，及显示文字。流程定义是使用结点定义时定义的各结点

- | | |
|---|-----------------|
| 1 | 结点定义 |
| 2 | 结点名称=>结点类型:结点说明 |
| 3 | |
| 4 | 流程定义 |
| 5 | 结点名称->结点名称 |

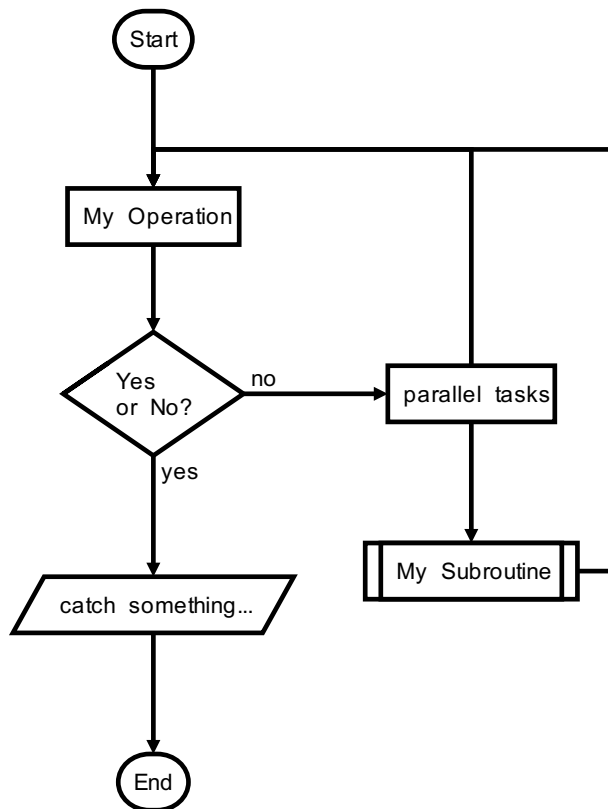
不同的结点

主要结点有

类型	名称
start	起始结点
end	结束结点
operation	处理结点
subroutine	子路由结点
condition	条件结点
inputoutput	输入输出结点
parallel	并行结点

示例

```
1  ``flow!
2  st=>start: Start:>http://www.google.com[blank]
3  e=>end:>http://www.google.com
4  op1=>operation: My Operation
5  sub1=>subroutine: My Subroutine
6  cond=>condition: Yes
7  or No?:>http://www.google.com
8  io=>inputoutput: catch something...
9  para=>parallel: parallel tasks
10
11 st->op1->cond
12 cond(yes)->io->e
13 cond(no)->para
14 para(path1, bottom)->sub1(right)->op1
15 para(path2, top)->op1
16  ``
```



可点击的结点

在结点定义阶段，想要可点击的结点结尾添加 `:>`，并带上要跳转的地址。

- 1 | 结点定义
- 2 | 结点名称=>结点类型:结点说明:>链接地址

默认是在本页进行跳转，可以在链接地址后面添加 `[blank]`，用来在新窗口打开

- 1 | 结点定义
- 2 | 结点名称=>结点类型:结点说明:>链接地址[blank]

条件结点

定义为条件结点后，在流程定义阶段，可以对条件结点进行 `yes` 和 `no` 参数的传递

- 1 | 流程定义
- 2 | 条件结点名称 (yes) -> 结点名称
- 3 | 条件结点名称 (no) -> 结点名称

并行结点

在流程定义阶段，可以对并行结点进行不同的参数设置

疑问

相关

1. [flowchart](#) 官网

