
目录

简介	1.1
gitbook的安装	1.2
mac	1.2.1
windows	1.2.2
linux	1.2.3
常用插件	1.3
样式风格	1.3.1
网页功能	1.3.2
图表公式	1.3.3
分享连接	1.3.4
其他	1.3.5
结束	1.4

关于gitbook安装使用和常用插件的安装使用

gitbook的安装

mac

安装Gitbook的主要流程如下：安装node.js>安装Gitbook>安装Gitbook编辑器>安装calibre>导出PDF。

一. 安装node.js

1. 先安装brew, 官网

运行如下命令安装¹

```
/usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install  

```

2. 安装nodejs

运行如下命令：`brew install nodejs`

3. 安装gitgook命令行

运行如下命令 `sudo npm install gitbook-cli -g`

-g全局安装 sudo 一定要加上，否则没权限

4. 安装gitbook

输入命令：`gitbook -v`

这个命令是看版本号，如果没安装gitbook会自动安装。输入`npm install gitbook -g`也能安装，但是我用下来会出错。主要是权限问题。安装完成后再次运行`gitbook -v` 出现了gitbook-cli和gitbook的版本号说明安装成功

二.gitbook编辑器

编辑器有很多，好用的有mweb，typed。mweb功能齐全使用方便，typed极端简介，美，有写东西的欲望，功能体验做的极好。但功能没有mweb多，比如自动上传图片到图床，比如支持latex公式和流程的预览都没有。

简单的使用说明

- 使用前要有README.md和SUMMARY.md文件
- 使用 `gitbook init` 命令初始化要写的文章或书籍,会根据 SUMMARY.md生成目录结构及文件
- 编写生成的文件
- `gitbook build`生成html版本的书，在`_book`目录下
- `gitbook serve`预览书籍，[\[http://localhost:4000\]](http://localhost:4000) 可以预览
- 要使用插件需要手动创建book.json的文件

下面给出这些文件的demo

- SUMMARY.md

```
# Summary

* [简介](README.md)
* [gitbook的安装](chapter1/README.md)
    * [mac](chapter1/section1.md)
    * [windows](chapter1/section2.md)
    * [linux](chapter1/section3.md)
* [常用插件](chapter2/README.md)
    * [样式风格](chapter2/section1.md)
    * [网页功能](chapter2/section2.md)
    * [图标公式](chapter2/section3.md)
    * [分享连接](chapter2/section4.md)
    * [其他](chapter2/section5.md)
* [结束](end/README.md)
```

- book.json

```
{  
    "title": "gitbook tutorial",  
    "author" : "derek fan",  
    "description" : "记录Gitbook的配置和一些插件的使用",  
    "language" : "zh-hans",  
    "structure": {  
        "readme": "README.md"  
    },  
    "links": {  
        "sidebar" : {  
            "Home" : "http://zhangjikai.com"  
        },  
        "gitbook": false,  
        "sharing": {  
            "google": false,  
            "facebook": "facebook_derek",  
            "twitter": false,  
            "all": false  
        }  
    },  
    "styles": {  
        "website": "styles/website.css",  
        "ebook": "styles/ebook.css",  
        "pdf": "styles/pdf.css",  
        "mobi": "styles/mobi.css",  
        "epub": "styles/epub.css"  
    },  
    "plugins": [  
        "splitter",  
        "github",  
        "anchors",  
        "chart",  
        "sitemap",  
        "latex-codecogs",  
        "mermaid"  
    ],  
    "pluginsConfig": {  
        "github": {  
            "url": "https://github.com/your/repo"  
        }  
    }  
}
```

```
        },
        "chart": {
            "type": "highcharts"
        },
        "sitemap": {
            "hostname": "http://mybook.com/"
        }
    },
    "pdf": {
        "pageNumbers": true,
        "fontFamily": "Arial",
        "fontSize": 12,
        "paperSize": "a4",
        "margin": {
            "right": 62,
            "left": 62,
            "top": 56,
            "bottom": 56
        }
    }
}
```

三.生成PDF

1. 安装calibre插件

玩过kindle的都知道，calibre是一款非常方便的开源电子书转换软件。在这里，我们也是用到ebook-convert这个插件。

首先在calibre官网下载插件，下载链接：<https://calibre-ebook.com/download>。下载适合自己系统的版本。

2.将安装的calibre放在系统应用中，然后将app添加到path中。

mac

```
sudo ln -s /Applications/calibre.app/Contents/MacOS/ebook-  
convert /usr/local/bin
```

3.生成pdf

```
gitbook pdf
```

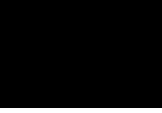
执行完以上代码，进入书籍目录，即可看到已经转换完成的PDF了。大功告成！

windows

运行如下命令安装¹

linux

TODO



常用插件

要使用插件书目录下必须要有book.json 插件添加方式如下：

```
"plugins": [
    "splitter",
    "github",
    "anchors",
    "chart",
    "sitemap",
    "latex-codecogs"
],
"pluginsConfig": {
    "github": {
        "url": "https://github.com/your/repo"
    },
    "chart": {
        "type": "highcharts"
    },
    "sitemap": {
        "hostname": "http://mybook.com/"
    }
},
```

然后执行 `gitbook install` ,就可以完成插件的安装了 有一些插件需要是用npm安装，具体见插件使用说明，如果安装失败了请卸载，重新安装。

样式风格

1.anchors

标题带有 github 样式的锚点。如下：

注意前面的小锚点，鼠标移上去才显示

2.styles-sass

使用 SASS 替换 CSS。可以方便的使用自定义的sass代替css

```
{  
  "plugins": ["styles-sass"],  
  "styles": {  
    "pdf": "./styles/pdf.sass"  
  }  
}
```

3.styles-less

使用 LESS 替换 CSS。可以方便的使用自定义less代替css

```
{  
  "plugins": ["styles-less"],  
  "styles": {  
    "pdf": "./styles/pdf.less"  
  }  
}
```

4.fontsettings

默认的字体、字号、颜色设置插件。

```
{  
  "pluginsConfig": {  
    "fontsettings": {  
      "theme": "white", // 'sepia', 'night' or 'white',  
      "family": "sans", // 'serif' or 'sans',  
      "size": 2 // 1 - 4  
    }  
  }  
}
```

5. tbfed-pagefooter

自定义页脚，显示版权和最后修订时间。

```
"tbfed-pagefooter": {  
  "copyright": "&copy Taobao FED Team",  
  "modify_label": "该文件修订时间：",  
  "modify_format": "YYYY-MM-DD HH:mm:ss"  
}
```

原作者很久没更新了，gitbook3.0以上版本出现bug，copyright参数没有用。我看了下源码，修改index.js文件如下：把这段替换原文件即可

```
var moment = require('moment');  
module.exports = {  
  book: {  
    assets: './assets',  
    css: [  
      'footer.css'  
    ],  
  },  
  hooks: {  
    'page:before': function(page) {  
      var _label = 'File Modify: ',
```

```

        _format = 'YYYY-MM-DD HH:mm:ss',
        _copy = ''
      if(this.options.pluginsConfig['tbfed-pagefooter']) {
        _label = this.options.pluginsConfig['tbfed-pagefooter']
        ['modify_label'] || _label;
        _format = this.options.pluginsConfig['tbfed-pagefoote
r']['modify_format'] || _format;

        _c = this.config.get('tbfed-pagefooter')['copyright'];

        _copy = _c ? _c + ' all right reserved, ' + _copy : _c
        copy;
      }
      var _copy = '<span class="copyright">' + _copy + '</span>'
      var str = '\n\n<footer class="page-footer">' + _copy +
        '<span class="footer-modification">' +
        _label +
        '\n{{file.mtime | date("'" + _format +
        '"')}}\n</span></footer>'
      page.content = page.content + str;
      return page;
    }
  },
  filters: {
    date: function(d, format) {
      return moment(d).format(format)
    }
  }
};

```

6.prism

基于 Prism 的代码高亮，效果很好

```
{
  "plugins": ["prism", "-highlight"]
}
```

EG:

```

func getAllMysqlString(fp string) (allStr string) {
    err := filepath.Walk(fp, func(path string, f os.FileInfo,
err error) error {
        if f == nil {
            return err
        }
        if !f.IsDir() {
            mysqlLexer := new(lexer.MysqlLexer)
            sl := new(lexer.StructLexer)
            fileStr := sl.GetStructFile(fp + f.Name())
            slist := sl.StructStr(fileStr)
            str := ""
            for _, v := range slist {
                str = str + "\n" + mysqlLexer.CreateSqlByStru
ctStr(v)
            }
            allStr = allStr + "\n" + str
            // fmt.Println(str)
        }
        return nil
    })
    if err != nil {
        fmt.Printf("filepath.Walk() returned %v\n", err)
    }
    return allStr
}

```

要先禁用默认的代码高亮插件

7.page-footer-ex

自定义页脚比tbfed-pagefooter 好用。效果见页脚

8.ace

插入代码高亮编辑器 例子

```
func getAllMysqlString(fp string) (allStr string) {
    err := filepath.Walk(fp, func(path string, f os.FileInfo,
err error) error {
        if f == nil {
            return err
        }
        if !f.IsDir() {
            mysqlLexer := new(lexer.MysqlLexer)
            sl := new(lexer.StructLexer)
            fileStr := sl.GetStructFile(fp + f.Name())
            slist := sl.StructStr(fileStr)
            str := ""
            for _, v := range slist {
                str = str + "\n" + mysqlLexer.CreateSqlByStru
ctStr(v)
            }
            allStr = allStr + "\n" + str
            // fmt.Println(str)
        }
        return nil
    })
    if err != nil {
        fmt.Printf("filepath.Walk() returned %v\n", err)
    }
    return allStr
}
```

9.sectionx

sectionx 分离各个段落，并提供一个展开收起的按钮。

样式风格

```
<!--sec data-title="Introduction" data-id="section0" data-show=true ces-->

Insert markdown content here (you should start with h3 if you
use heading).

<!--endsec-->
```

按钮 可控制是否显示

```
<button class="section" target="section0" show="Show next section" hide="Hide next section"></button>
```

网页功能

1. editlink

内容顶部显示“编辑本页”链接。 TODO

2. splitter

在左侧目录和右侧内容之间添加一个可以拖拽的栏，用来调整两边的宽度。

<-----

3. github

在右上角显示 github 仓库的图标链接。

4. disqus

添加 disqus 评论插件。

```
{  
  "plugins": ["disqus"],  
  "pluginsConfig": {  
    "disqus": {  
      "shortName": "XXXXXXX"  
    }  
  }  
}
```

如果想要某页面禁用评论可以在文件最前面添加：

```
---  
disqus: false  
---  
# 标题
```

5.sitemap

生成站点地图。

执行gitbook build的时候生成sitemap.xml有利于搜索引擎收录

6.book-summary-scroll-position-saver

自动保存左侧目录区域导航条的位置。

7.mcqx

使用选择题 [官方资料](#)

This is a question?

- A. First option
- B. Second option
- C. Third option
- D. Fourth option

ANSWER:A

##8.fbqx

使用填空题

- 无法正常运行

9.spoiler

隐藏答案

```
This is a spoiler: {%s%}Hello World.{%ends%}
```

鼠标划过显示：Hello World.

图表公式

1. chart

使用 C3.js 图表

更多例子见：[官网](#)

如下： 1.柱图

```
{% chart %}
{
  "chart": {
    "type": "bar"
  },
  "title": {
    "text": "Fruit Consumption"
  },
  "xAxis": {
    "categories": ["Apples", "Bananas", "Oranges"]
  },
  "yAxis": {
    "title": {
      "text": "Fruit eaten"
    }
  },
  "series": [
    {
      "name": "Jane",
      "data": [1, 0, 4]
    },
    {
      "name": "John",
      "data": [5, 7, 3]
    }
  ]
}
{% endchart %}
```

2. 饼图

```
{  
    "chart": {  
        "plotBackgroundColor": null,  
        "plotBorderWidth": null,  
        "plotShadow": false,  
        "type": "pie"  
    },  
    "title": {  
        "text": "Browser market shares January, 2015 to May,  
2015"  
    },  
    "tooltip": {  
        "pointFormat": "{series.name}: <b>{point.percentage:.  
1f}</b>%"  
    },  
    "plotOptions": {  
        "pie": {  
            "allowPointSelect": true,  
            "cursor": "pointer",  
            "dataLabels": {  
                "enabled": true,  
                "format": "<b>{point.name}</b>: {point.percentage:.  
1f} %",  
                "style": {  
                    "color": "black"  
                }  
            }  
        }  
    },  
    "series": [  
        {  
            "name": "Brands",  
            "colorByPoint": true,  
            "data": [{  
                "name": "IE",  
                "y": 56.33  
            }, {  
                "name": "Chrome",  
                "y": 21.33  
            }, {  
                "name": "Firefox",  
                "y": 10.33  
            }, {  
                "name": "Safari",  
                "y": 8.33  
            }, {  
                "name": "Opera",  
                "y": 3.33  
            }]  
        }  
    ]  
}
```

```
        "name": "Chrome",
        "y": 24.03,
        "sliced": true,
        "selected": true
    }, {
        "name": "Firefox",
        "y": 10.38
    }, {
        "name": "Safari",
        "y": 4.77
    }, {
        "name": "Opera",
        "y": 0.91
    }, {
        "name": "Other",
        "y": 0.2
    }]
}
}
```

3. [latex-codecogs](#)

使用数学方程式。[latex规则使用说明](#)

[中文使用说明](#)

```
Inline math: {%
  math %
} \int_{-\infty}^{\infty} g(x) \, dx % endmath

Block math:

{%
  math %
}
\int_{-\infty}^{\infty} g(x) \, dx
% endmath %
```

Inline math:

Block math:

```
{% math %}
A_{m,n} =
\begin{pmatrix}
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
{% endmath %}
```

4. mermaid-gb3

最好的流程图，甘特图，类图等图形制作插件

这个插件会与prism冲突，解决方案把这个插件放在prism插件之前就可以了

官方文档

安装

```
$ npm install gitbook-plugin-mermaid-gb3
```

例子：

Flowchart

```
graph TD;
A-->B;
A-->C;
B-->D;
C-->D;
```

```
graph TD; A-->B; A-->C; B-->D; C-->D;
```

Sequence diagram

```
sequenceDiagram
    participant Alice
    participant Bob
    Alice->>John: Hello John, how are you?
    loop Healthcheck
        John->>John: Fight against hypochondria
    end
    Note right of John: Rational thoughts <br/>prevail...
    John-->>Alice: Great!
    John->>Bob: How about you?
    Bob-->>John: Jolly good!
```

```
sequenceDiagram participant Alice participant Bob
Alice->>John: Hello John, how are you?
loop Healthcheck
    John->>John: Fight against hypochondria
end
Note right of John: Rational thoughts <br/>prevail...
John-->>Alice: Great!
John->>Bob: How about you?
Bob-->>John: Jolly good!
```

Gantt diagram

```
gantt
dateFormat YYYY-MM-DD
title Adding GANTT diagram functionality to mermaid

section A section
Completed task :done, des1, 2014-01-06, 2014
-01-08
    Active task :active, des2, 2014-01-09, 3d
    Future task : des3, after des2, 5d
    Future task2 : des4, after des3, 5d

section Critical tasks
Completed task in the critical line :crit, done, 2014-01-
06, 24h
    Implement parser and jison :crit, done, after de
s1, 2d
    Create tests for parser :crit, active, 3d
    Future task in critical line :crit, 5d
    Create tests for renderer :2d
    Add to mermaid :1d

section Documentation
Describe gantt syntax :active, a1, after de
s1, 3d
    Add gantt diagram to demo page :after a1 , 20h
    Add another diagram to demo page :doc1, after a1 , 48
h

section Last section
Describe gantt syntax :after doc1, 3d
Add gantt diagram to demo page : 20h
Add another diagram to demo page : 48h
```

gantt dateFormat YYYY-MM-DD title Adding GANTT diagram functionality to mermaid
section A section
Completed task :done, des1, 2014-01-06, 2014-01-08
Active task :active, des2, 2014-01-09, 3d
Future task : des3, after des2, 5d
Future task2 : des4, after des3, 5d
section Critical tasks
Completed task in the critical line :crit, done, 2014-01-06, 24h
Implement parser and jison :crit, done, after des1, 2d
Create tests for parser :crit, active, 3d
Future task in critical line :crit, 5d
Create tests for renderer :2d
Add to mermaid :1d
section Documentation
Describe gantt syntax :active, a1, after des1, 3d
Add gantt diagram to demo page :after a1 , 20h
Add another diagram to demo page :doc1, after a1 , 48h
h
section Last section
Describe gantt syntax :after doc1, 3d
Add gantt diagram to demo page : 20h
Add another diagram to demo page : 48h

parser and json :crit, done, after des1, 2d Create tests for parser :crit,
active, 3d Future task in critical line :crit, 5d Create tests for renderer :2d
Add to mermaid :1d section Documentation Describe gantt syntax :active,
a1, after des1, 3d Add gantt diagram to demo page :after a1 , 20h Add
another diagram to demo page :doc1, after a1 , 48h section Last section
Describe gantt syntax :after doc1, 3d Add gantt diagram to demo page : 20h
Add another diagram to demo page : 48h

分享连接

1.ad

在每个页面顶部和底部添加广告或任何自定义内容。 TODO

2.sharing

默认的分享插件。

```
{  
  "pluginsConfig": {  
    "sharing": {  
      "facebook": true,  
      "twitter": true,  
      "google": false,  
      "weibo": false,  
      "instapaper": false,  
      "vk": false,  
      "all": [  
        "facebook", "google", "twitter",  
        "weibo", "instapaper"  
      ]  
    }  
  }  
}
```

其他

pandoc 生成中文pdf

中英文文档转换

在编写时，由于我们主体还是中文，那么若文档中存在中文字符，那么转换就会出问题。为了使的其支持中文，我们需要使用xelatex编译器（有关xelatex，还请大家自行搜索），下面直接放干货，可以一步操作到位的命令如下。

```
pandoc -N -s --toc --smart --latex-engine=xelatex -V CJKmainfont='黑体' -V mainfont='Times New Roman' -V geometry:margin=1in input.md -o output.pdf
```

大家可以自行编辑一些.md文档然后使用着条命令进行转换尝试。另外，Markdone转PDF的操作，也可以几个.md文件整合成一个PDF文件，其命令如下。

```
pandoc -N -s --toc --smart --latex-engine=xelatex -V CJKmainfont='黑体' -V mainfont='Times New Roman' -V geometry:margin=1in in1.md in2.md ... -o output.pdf
```

生成应用论文的命令

到百度学术 找个论文

该文献的Bibtex信息就是这样的一段文本。我们将其拷贝下来，存储到.bib文件里。

依此类推，我们再下载其他两份文献的Bibtex信息。

```
@article{吴信东2016从大数据到大知识,
  title={从大数据到大知识:HACE+BigKE},
  author={吴信东 and 何进 and 陆汝钤 and 郑南宁},
  journal={自动化学报},
  volume={42},
  number={7},
  pages={965-982},
  year={2016},
}
```

还有一个"chinese-gb7714-2005-numeric.csl"样式文件。[在这里可以找到](#)

```
pandoc --latex-engine=xelatex --filter pandoc-citeproc --bibliography=myref.bib --csl=chinese-gb7714-2005-numeric.csl a.md -o a.docx
```

pandoc 生成幻灯片

pandoc将markdown的一级标题视为幻灯片组，将二级标题视为幻灯片标题，二级标题下的内容放在一张幻灯片里。不过，如果某个一级标题下没有二级标题，那么pandoc将把一级标题视为幻灯片标题，将这个一级标题下的内容放在一张幻灯片里，二级标题显示为子框。

在任何时候，用'-----'都可以产生一张新幻灯片。

pandoc对markdown有个扩展，在前面加上三行以%开头的内容，分别是标题、作者和日期。

例子如下

```
% Title  
% auther  
% date  
  
# This is a group  
  
## This is title 1  
  
This is first slide  
  
- item 1  
- item 2  
  
## This is tile 2  
  
This is second slide  
  
-----  
  
This is third slide
```

使用xelatex引擎 在模板文件中正确配置 首先导出pandoc的beamer默认配置

```
pandoc -D beamer > beamer-template.tex
```

设置页面比例 在模板文件的documentclass中加入参数aspectratio=169, 即

```
\documentclass[..., aspectratio=169]{\$documentclass\$}
```

设置背景图片 笔者在工作中，需要使用公司的ppt模板，因此需要设置幻灯片的背景。最终采用的解决方法是这样的：

在模板文件中加入命令

```
\usebackgroundtemplate{\includegraphics[width=\paperwidth, height=\paperheight]{background.png}}
```

如果要对标题页单独设置背景，笔者用的是[这篇文章里的方法](#):

```
\RequirePackage{tikz}
\addtobeamertemplate{title page}{%
    \begin{tikzpicture}[remember picture, overlay]
        \node [xshift=0cm, yshift=0cm] at (current page.center)
        {\includegraphics[width=\paperwidth, height=\paperheight]{background_title.png}};
    \end{tikzpicture}%
}{}{}
```

设置字体颜色

```
\definecolor{beamer@font}{RGB}{58, 80, 99}
\setbeamercolor{titlelike}{use=structure, fg=beamer@font}
\setbeamercolor{title page}{fg=beamer@font}
\setbeamercolor{block body}{fg=beamer@font}
\setbeamercolor{block title}{fg=beamer@font}
\setbeamercolor{block title alerted}{fg=beamer@font}
\setbeamercolor{block title example}{fg=beamer@font}
\setbeamercolor{block body alerted}{fg=beamer@font}
\setbeamercolor{block body example}{fg=beamer@font}
```

pandoc 生成html幻灯片

分级标题、列表、插入图片……等标准的Markdown语法均被支持，和平常用Markdown记笔记写博客无异。

为了让Pandoc对不包含任何元信息的Markdown文本进行处理生成幻灯片，在文本开头需要包含三行以%打头的元信息：标题、作者和日期。

为了生成合适的用于演示的HTML文档，需要记住的另一点是：在默认情况下每个二级标题是一张独立的幻灯片。这样在写作的时候，只需注意把每个二级标题下的内容控制在适当的长度。

使用reveal.js生成幻灯片

```
$ git clone https://github.com/hakimel/reveal.js
```

除了默认的外观主题以外， reveal.js还提供了多个主题可供选择， default：（默认）深灰色背景，白色文字 beige：米色背景，深色文字 sky：天蓝色背景，白色细文字 night：黑色背景，白色粗文字 serif：浅色背景，灰色衬线文字 simple：白色背景，黑色文字 solarized：奶油色背景，深青色文字

可以使用--slide-level选项覆盖默认的Slide level --slide-level 1 则#为一页，
2：则##为一页

---可分页

可以直接插入TeX公式：

```
$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{n\rightarrow\infty} (1+x/n)^n$
```

控制TeX公式渲染方式的选项有--mathml, --webtex, --mathjax和--
latexmathml。（Chrome和Firefox均支持MathML）

一般用--mathjax

代码高亮风格 控制代码高亮风格的选项有： --highlight-style pygments --
highlight-style kate --highlight-style monochrome --highlight-style espresso -
-highlight-style haddock --highlight-style tango --highlight-style zenburn

提示板

在Markdown中插入标签

的小抄：

```
<div class="notes">  
This is my note.  
  
- It can contain markdown  
- like this list  
  
</div>
```

使用键盘s键打开提示板。当然，这个提示板是用来给演讲者自己看的，是不用mirror到外接投影仪的。



附送完整命令：

```
pandoc testslide.md -o slides.html --mathjax -t revealjs -s -  
v theme=beige --slide-level 2
```

markdown

```
% 国商投总结  
% derek fan  
% March 22, 2017  
  
# 工作总结
```

```
<div class="notes">
```

```
  This is my note.
```

```
  - It can contain markdown
```

```
  - like this list
```

```
</div>
```

```
## 服务器
```

```
  - Turn off alarm
```

```
  - Get out of bed
```

```
## 网站
```

```
  -  $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \lim_{n \rightarrow \infty} (1+x/n)^n$ 
```

```
  - Drink coffee
```

```
# PMBA工作
```

```
## 推广
```

```
  - Eat spaghetti
```

```
  - Drink wine
```

```
-----
```

```
![picture of spaghetti](a.png)
```

```
## 阿萨德
```

```
  - 完成XXXXX
```

```
  - 完成XXX
```

结束

结束