

# Markdown写作进阶：Pandoc入门浅谈

草稿。慎入。

熟悉Markdown的朋友，不由地奇怪，为什么它不支持表格、不支持数学公式、不支持很多你想要的功能。这一切，与Markdown自身的设计哲学有关系，提供尽可能少的，用于创作的功能，将对它的扩展，交给各路开源爱好者。

于是，我们可以看到multimarkdown，我们也可以看到maruku等各路扩展。

然而，这些扩展某种意义上，并不是针对写作的。直到有一天，开源界传说中的高帅富社区，haskell社区的作品Pandoc开始诞生，我们有了更好的选择。

haskell是什么？一种编程语言。但是，每位资深的开发者学习它，都会有些不知所措，因为，它的编程思路不同于通常意义上类似于C#、Python、Java这些流行的编程语言。它来自一群高智商的开发者的贡献，据说人人有PHD学位，是函数式编程语言的典范。

文本的格式转换，在文青看来，是一个很简单的东西。但是，每位开发过这方面程序的技术青年，都知道它有多么痛苦。开发出的程序有多么丑陋。甚至可以说，这是个无底黑洞。所以，当基于haskell开发的Pandoc刚刚诞生时，不相信者众多，它真的能实现文本格式的优雅转换吗？

事实证明，函数式编程的确是干这脏活、累活的最恰当选择。Pandoc的确成功了，并已经成功在短期内构建一个完整的生态链，于是，有了我们这篇小文。

---

## 1. Pandoc写作环境

### 1.1 Pandoc 安装

### 1.2 Pandoc Markdown 编辑器

- 作者用的是开源的[TextMate]
- [Mou]已支持引用功能，但尚不支持表格。
- 作者在努力说服其他编辑器作者默认支持Pandoc，请大家静候。

---

## 2. Pandoc Markdown语法特色

详情参见：[Pandoc](#)

### 2.1 Pandoc写表格

如下所示：

测量指标	低程度中心性	低亲近中心性
高程度中心性		“自我”所嵌入的聚类 远离网络中其他节点
高亲近中心性	是联系重要他人或 活跃人物的关键人物	

： 表3-1 不同中心性测量指标之间的关系<sup>[16]</sup>

## 2.2 Pandoc写参考文献

使用[Pandoc Markdown]格式的写法是在文中使用：

[<sup>1</sup>]<sup>[2]</sup>

然后在文末使用：

[<sup>1</sup>]:  
[<sup>2</sup>]:

如下例所示：

正文中是：

当一个人在看社会网络时，首先提出的问题之一就是：在社会网络中谁是更重要的人？或者谁拥有更大的权力？在介绍部分（参考第一章介绍的非正式网络的力量），我们发现社会网络是由一个强大的（但大部分是无形的）的“秘书”统治<sup>[7]</sup>，类似的观点亦可从一系列社会网络测量指标（参见 *中心性*）中获得。

然后在文章末尾写上：

[<sup>7</sup>]:译者注：作者此处秘书指的是第一章中“非正式网络的力量”一节介绍的ACME咨询公司的例子中

---

## 3. Pandoc文档转换

### 3.1 Markdown转html、word与PDF

如下例所示：

转html格式

```
pandoc 01-chapter2.markdown -o chapter2.html -c Github.css
```

转word格式

```
pandoc 01-chapter2.markdown -o chapter2.docx -c Github.css
```

### 3.2 Markdown转PDF

安装LaTeX

待补。

转pdf格式

```
pandoc 01-chapter2.markdown -o 01-chapter2.pdf --latex-engine=xelatex -V m
```

---

## 4 Pandoc生成epub、html与在线电子书

### 4.1 安装nodejs与grunt、bower

下载nodejs安装包，网址为：<http://nodejs.org/>

下载之后，安装默认安装。安装完之后，确认安装是否成功。

```
npm -v
```

如果如我一样，使用的是zsh。在.zshrc文件中添加nodejs的路径。

```
mate ~/.zshrc
```

在文件末尾添加：

```
export PATH=/usr/local/share/npm/bin:$PATH
```

然后，安装grunt。

```
sudo npm install -g grunt
```

安装bower等。

---

```
sudo npm install -g bower
```

## 4.2 使用wbb生成epub、html与在线电子书示范

以著名的progit为例。

```
git clone https://github.com/PascalPrecht/wbb test2012
cd test2012
```

然后检查clone下来的项目，是否已经包含了grunt.js等文件。各项生成代码写在那里面。

书稿则写在/Users/ouyang/dev/mywrite/test2012/src/chapters 这样的目录下面。src/chapters下面。

写作书稿时，完全按照正常git流程管理。书稿预览效果，是在test2012的根目录下面，运行：

```
grunt wbb:html
```

自定义各项参数，请修改grunt.js文件。

生成的epub、html与leanpub在线电子书，在builds目录下面。

---

## 5 Pandoc的科技写作

### 5.1 Pandoc与bibdesk、APA文献格式的配合

### 5.2 Pandoc+数学公式

### 5.3 Pandoc+R语言图表

---

## 6 其他

### 6.1 酷！Pandoc转slide

### 6.2 Pandoc转wiki

### 6.3 Pandoc转Jekyll