# 贡献者文档格式

**受众**: 文档作者和编辑

虽然此样式指南还将参考使用ReStructured Text（也称为RST）的最佳实践，但总体而言，我们建议使用Markdown编写文档，因为这是一种更普遍被接受的文档标准。但是，这两种格式都是可用的，如果您决定在RST中编写主题（或者正在编辑RST主题），请务必参考本样式指南。

**如果有疑问，请使用文档本身来指导如何修改内容格式**

- RST格式.
- Markdown格式.

如果您只想查看内容的格式，可以通过单击页面右上角的`Edit on Github`链接导航到Fabric repo查看原始文件。然后单击`Raw`选项卡。这将显示文档的格式。**不要试图在Github上编辑文件。**如果要进行更改，请克隆仓库并按照[贡献]（./CONTRIBUTING.html)来创建PR。

## 措辞

**避免使用 “白名单(whitelist)”, “黑名单（blacklist）”, “主（master）”, or “从（slave）”.**

除非使用这些词是绝对必要的（例如，在引用使用它们的代码部分时），否则不要使用这些词。要么更加明确（例如，描述“白名单”的实际作用），要么找到替代词，如“allowlist”或“blocklist”。

**教程的顶部应该有一个步骤列表。**

教程开头的步骤列表（带有指向相应部分的链接）可以帮助用户找到他们感兴趣的特定步骤。例如，查看在Fabric上使用私有数据。

**“Fabric”, “Hyperledger Fabric” 还是 “HLF”?**

优先使用“Hyperledger Fabric”，次选“Fabric”。不要使用HLF且不要单独使用Hyperledger。

**Chaincode vs. Chaincodes?**

单独一个链码是 “chaincode”. 如果你想谈论多个链码, 使用 “chaincodes”.

**Smart contracts(智能合约)?**

通俗地说，智能合约被视为等同于链码，但在技术层面上，更正确的说法是“Smart contract（智能合约）”是链码内部的业务逻辑，它包含更大的封装和实现。

**JSON vs .json?**

使用 “JSON”。 其它文件格式同理 (例如, YAML)。

**curl vs cURL.**

那个工具叫 “cURL”. 而那条命令是 “curl”。

**Fabric CA.**

不要叫它 “fabric-CA”, “fabricCA”, 或 FabricCA。 它应该是 Fabric CA。 然而 Fabric CA 二进制客户端可以被成为 `fabric-ca-client` 。

**Raft and RAFT.**

“Raft” 不是首字母缩写。 不要叫它 “RAFT ordering service”。

**提及读者.**

最好使用 “你” 或者 “我们”. 避免使用 “我”.

**和号 (&).**

不能代替“和”这个词。除非您有理由使用它（例如在包含它的代码片段中），否则请避免使用它们。

**缩写.**

首字母缩略词的第一个用法应该详细说明，除非它是一个使用广泛的首字母缩略词，否则不要这么使用。例如，第一次使用“软件开发工具包（SDK）”。然后使用“SDK”。

**尽量不要使用一个词过于频繁**

如果你能避免一个词在一个句子中重复使用，请尽量这样做。在一个段落中不要使用一个词两次以上更好。当然，有时可能无法避免这一点——一份关于正在使用的状态数据库的文档可能会充满“数据库”或“账本”一词。但是过度使用任何一个词都会让读者感觉麻木。

**文件如何命名?**

在单词之间使用下划线。另外，教程也应该这样命名。例如， `identity_use_case_tutorial.md` . 虽然并非所有文件都使用此标准，但新文件应遵守该标准。

# 格式和标点符号

**列长.**

如果您查看文档的原始版本，您将看到许多主题符合大约70个字符的行长度。这个规则不再是必需遵守的了，所以您愿意写多长都可以。

**何时加粗?**

别太常用就行。最好的用法是作为一个总结，或者作为一种方式，提醒读者注意你想谈论的概念。“区块链网络包含一个账本、至少一个链码和Peers”，尤其是如果你要在那一段中谈论这些事情。避免仅仅为了强调一个词而使用它们，比如“区块链网络**必须**使用适当的安全协议”。

**什么时候把文字框起来 `nnn` ?**

这有助于提醒人们注意那些在纯英语中没有意义的单词，或者在引用部分代码时（尤其是在文档中放了代码片段的情况下）。例如，当谈到fabric samples目录时，用back tics(~键下面那个符号)包围 `fabric samples` 。代码函数一样 `hf.Revoke` . 如果您在代码上下文中引用的话，将那些在纯英语中有意义的单词放回原处也可能是有意义的。例如，当使用 `attribute` 作为访问控制列表的一部分时。

**使用破折号是否合适?**

破折号可能非常有用，但在技术层面上不一定合适于分割陈述句。让我们来看看这个例句:

```
这给我们留下了一个精简的JSON对象--- config.json，位于first network内的fabric samples文件夹中---它将作为配置更
```

有很多方法可以表示相同的信息，但在这种情况下，破折号会将信息分解，同时将其作为同一句话的一部分。如果使用破折号，请确保使用“em”破折号，它比连字符长三倍。这些破折号前后应该有一个空格。

**何时使用连字符?**

连字符通常用作“复合形容词”的一部分。例如，“喷气式-汽车（jet-powered car）”。注意复合形容词必须紧跟在被修饰名词之前。换句话说，“jet powered”本身不需要连字符。如果有疑问，可以使用Google，因为复合形容词很棘手，而且是语法论坛上的热门讨论。

（译者注:这里说的是英文拼写规则，选择性遵守）

**句号后有多少空格?**

一个

**数字应该如何呈现?**

数字0到9应该由文字表示。一、二、三、四等。10后的数字表示为数字。

例外情况是来自代码的用法。在这种情况下，使用代码中的任何内容。还有Org1之类的例子。别把它写成OrgOne。 （译者注:这里说的是英文拼写规则，选择性遵守）

**文档标题大小写规则**

应遵循句子大写的标准规则。换句话说，除非一个词是标题中的第一个词或专有名词，否则不要将其首字母大写。例如，”Understanding Identities in Fabric”(“理解Fabric中的身份”)应该是”Understanding identities in Fabric”。虽然并不是每个文档都遵循这个标准，但它是我们正在转向的标准，对于新的主题应该遵循这个标准。

主题内的标题应遵循相同的标准。

（译者注:这里说的是英文规则，中文语境下选择性遵守）

**用牛津逗号?**

是的，它更好

经典例子是, “我想感谢我的父母，安·兰德和上帝(I’d like to thank my parents, Ayn Rand and God)”, 更应该写成: “我想感谢我的父母，安·兰德，和上帝（I’d like to thank my parents, Ayn Rand, and God）”

**标题.**

这些应该是斜体字，这是我们文档中斜体字唯一真正有效的用法。

**计算机命令.**

通常，将每个计算机命令放在自己的代码段中。它读起来更好，尤其是当命令很长的时候。除了解释一大串环境变量的时候。

**代码段.**

在Markdown中，如果要发布示例代码，请使用三个back tics来分隔代码段。例如：

```
Code goes here.

Even more code goes here.

And still more.
```

在RST中，您需要使用类似于以下格式的格式设置代码段：

```
.. code:: bash

   Code goes here.
```

在适当的情况下，可以用`bash`代替Java或Go之类的语言。

**markdown中的列表枚举**

注意，在Markdown中，如果用空格分隔数字，则列表枚举将不起作用。Markdown认为这是一个新列表的开始，而不是旧列表的延续（每个数字都是`1.`）。如果需要枚举列表，则必须使用RST。

Markdwon中子标题最好使用无序列表，并且被推荐作为列表标记。

**链接.**

当链接到另一个文档时，使用相对链接，而不是直接链接。当命名一个链接时，不要只称它为“链接”。使用更具创造性和描述性的名称。出于可访问性的原因，链接名称还应清楚地表明它是一个链接。

**所有文档都必须以许可证声明结尾。**

在RST应该用:

```
.. Licensed under Creative Commons Attribution 4.0 International License
   https://creativecommons.org/licenses/by/4.0/
```

在markdown中:

```
<!--- Licensed under Creative Commons Attribution 4.0 International License
https://creativecommons.org/licenses/by/4.0/ -->
```

**缩进有多少空格?**

这将取决于用途。通常需要缩进两个空格，尤其是在RST中，尤其是在代码块中。在一个RTD的`.. note::`中，您必须缩进到注释后冒号后面的空格，如下所示：

```
.. note:: Some words and stuff etc etc etc (line continues until the 70 character limit line)
          the line directly below has to start at the same space as the one above.
```

**何时使用哪种类型的标题。**

在RST应该用:

```
Chapter 1 Title
===============

Section 1.1 Title
-----------------

Subsection 1.1.1 Title
~~~~~~~~~~~~~~~~~~~~~~

Section 1.2 Title
-----------------
```

请注意，标题下的内容长度必须与标题本身的长度相同。这在Atom中不是问题，Atom默认情况下为每个字符指定相同的宽度（这称为“monospacing”，如果您遇到问题的话！）需要这些信息。

在markdown中更简单，你可以这么用：

```
# The Name of the Doc (this will get pulled for the TOC).

## First subsection

## Second subsection
```

两种文件都不太支持不合规则的排序格式。例如，你想用`####`作为`#`标题后的第一行。Markdown不支持这样做。同样的，RTS将会默认将以你给定的格式排序（它会出现在你文档的第一行）。

**应尽可能使用相对链接。**

对于RST，建议遵循的语法是：

```
:doc:`anchor text <relativepath>`
```

不要将.rst后缀放在文件路径的末尾。

对于Markdown，建议遵循的语法是：

```
[anchor text](<relativepath>)
```

对于其他文件（如文本或YAML文件），请使用指向中的文件的直接链接，以github为例：

https://github.com/hyperledger/fabric/blob/master/docs/README.md

遗憾的是，在浏览RST文件时，github上的相对链接不起作用。