## 1 从0开始学习 GitHub 系列之「加入 GitHub」

<https://segmentfault.com/a/1190000005611460>

上面认识了 GitHub 的基本面貌之后，你需要了解一些 GitHub 的基本概念，这些概念是你经常会接触并遇到的。

（1）基本概念

Repository

仓库的意思，即你的项目，你想在 GitHub 上开源一个项目，那就必须要新建一个 Repository ，如果你开源的项目多了，你就拥有了多个 Repositories 。

Issue

问题的意思，举个例子，就是你开源了一个项目，别人发现你的项目中有bug，或者哪些地方做的不够好，他就可以给你提个 Issue ，即问题，提的问题多了，也就是 Issues ，然后你看到了这些问题就可以去逐个修复，修复ok了就可以一个个的 Close 掉。

Star

这个好理解，就是给项目点赞，但是在 GitHub 上的点赞远比微博、知乎点赞难的多，如果你有一个项目获得100个star都算很不容易了！

Fork

这个不好翻译，如果实在要翻译我把他翻译成分叉，什么意思呢？你开源了一个项目，别人想在你这个项目的基础上做些改进，然后应用到自己的项目中，这个时候他就可以 Fork 你的项目，这个时候他的 GitHub 主页上就多了一个项目，只不过这个项目是基于你的项目基础（本质上是在原有项目的基础上新建了一个分支，分支的概念后面会在讲解Git的时候说到），他就可以随心所欲的去改进，但是丝毫不会影响原有项目的代码与结构。

Pull Request

发起请求，这个其实是基于 Fork 的，还是上面那个例子，如果别人在你基础上做了改进，后来觉得改进的很不错，应该要把这些改进让更多的人收益，于是就想把自己的改进合并到原有项目里，这个时候他就可以发起一个 Pull Request（简称PR） ，原有项目创建人就可以收到这个请求，这个时候他会仔细review你的代码，并且测试觉得OK了，就会接受你的PR，这个时候你做的改进原有项目就会拥有了。

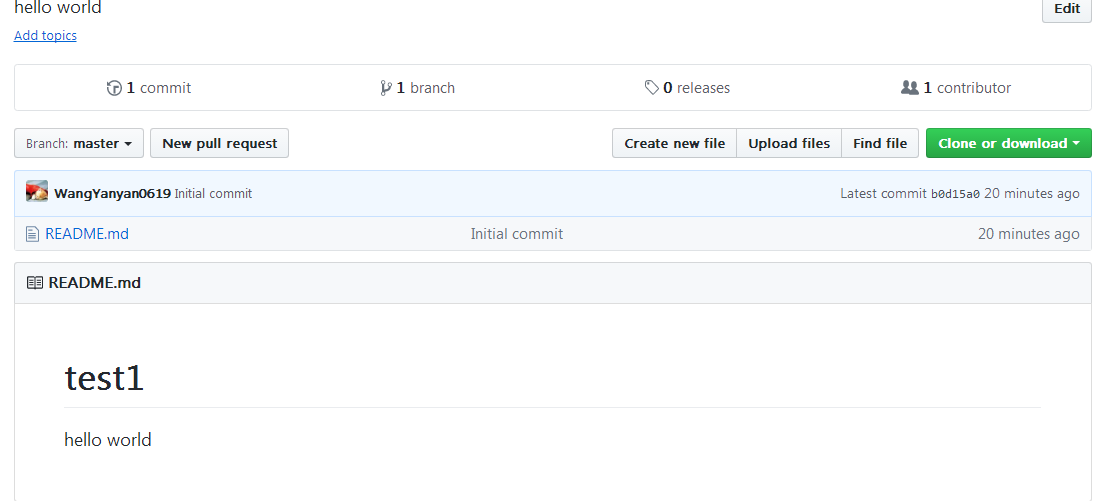
Watch

这个也好理解就是观察，如果你 Watch 了某个项目，那么以后只要这个项目有任何更新，你都会第一时间收到关于这个项目的通知提醒。

Gist

有些时候你没有项目可以开源，只是单纯的想分享一些代码片段，那这个时候 Gist 就派上用场了！

（２）创建项目



这是我创建的第一个项目，很简单，只有一个说明文件

可以看到这个项目只包含了一个 README.md 文件，但是它已经是一个完整的 Git 仓库了，你可以通过对它进行一些操作，如watch、star、fork，还可以 clone 或者下载下来。

这里提一下 README.md ，GitHub 上所有关于项目的详细介绍以及 Wiki 都是基于 Markdown 的，甚至之后在 GitHub 上搭建博客，写博客也是如此，所以如果还不懂 Markdown 语法的，建议先去学习下。推荐一篇学习 Markdown 的文章。

## ２　献给写作者的 Markdown 新手指南

https://www.jianshu.com/p/q81RER

（１）标题

这是最为常用的格式，在平时常用的文本编辑器中大多是这样实现的：输入文本、选中文本、设置标题格式。

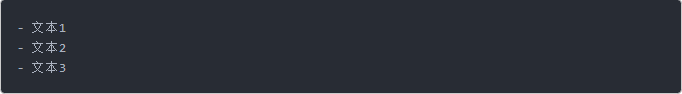
而在 Markdown 中，你只需要在文本前面加上 # 即可，同理、你还可以增加二级标题、三级标题、四级标题、五级标题和六级标题，总共六级，只需要增加 # 即可，标题字号相应降低。例如：



注：# 和「一级标题」之间建议保留一个字符的空格，这是最标准的 Markdown 写法。

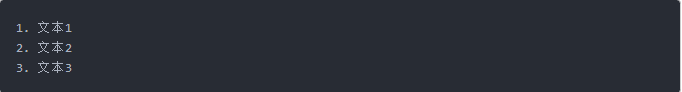
（２）列表

列表格式也很常用，在 Markdown 中，你只需要在文字前面加上 - 就可以了，例如：



如果你希望有序列表，

也可以在文字前面加上 1. 2. 3. 就可以了，例如：



注：*-*、*1.*和文本之间要保留一个字符的空格。

（３）引用

在我们写作的时候经常需要引用他人的文字，这个时候引用这个格式就很有必要了，在 Markdown 中，你只需要在你希望引用的文字前面加上 > 就好了，例如：



注：*>*和文本之间要保留一个字符的空格。

（４）粗体和斜体

Markdown 的粗体和斜体也非常简单，用两个 \* 包含一段文本就是粗体的语法，用一个 \* 包含一段文本就是斜体的语法。例如：



（５）代码引用

需要引用代码时，如果引用的语句只有一段，不分行，可以用 ` 将语句包起来。

如果引用的语句为多行，可以将```置于这段代码的首行和末行。

代码引用的案例截图：



（６）表格

相关代码：

