**Git –GitHub sprint**

Git是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。

Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

Git 与常用的版本控制工具 CVS, Subversion 等不同，它采用了分布式版本库的方式，不必服务器端软件支持。

# Git 远程仓库

Git 并不像 SVN 那样有个中心服务器。

目前我们使用到的 Git 命令都是在本地执行，如果你想通过 Git 分享你的代码或者与其他开发人员合作。 你就需要将数据放到一台其他开发人员能够连接的服务器上。

本例使用了 Github 作为远程仓库，你可以先阅读我们的 [Github 简明教程。](http://www.w3cschool.cc/w3cnote/git-guide.html" \t "_blank)

# User Case Sprint

## Github Sprint (not work properly)

**C:\Users\pc\Documents\GitHub>**

## git status

## git clone [git@github.com:ellwinn/readme.md](mailto:git@github.com:ellwinn/readme.md)

## \*\*open notepad++ do edit

## git add .

## git add -p

## Git Sprint (not work properly)

**Install git--**

**C:\software\download\**

**Git-2.8.2-64-bit**

* 完成安装之后，就可以使用命令行的 git 工具（已经自带了 ssh 客户端）了，另外还有一个图形界面的 Git 项目管理工具。 在开始菜单里找到"Git"->"Git Bash"，会弹出 Git 命令窗口，你可以在该窗口进行 Git 操作。
* $ dir 当前目录
* $cd c:/working 选择要使用的入工作目录

$ git config --global user.name "Ellwinn"

$ git config --global user.email "ellwinyu@hotmail.com"

* $ git config --list
* Git clone git://github.com/Ellwinn/testrepo needtorecite
* $ git add \*.\*
* $ git add README
* $ git commit -m '初始化项目版本'

## Git 配置

Git 提供了一个叫做 git config 的工具，专门用来配置或读取相应的工作环境变量。

这些环境变量，决定了 Git 在各个环节的具体工作方式和行为。这些变量可以存放在以下三个不同的地方：

* /etc/gitconfig 文件：系统中对所有用户都普遍适用的配置。若使用 git config 时用 --system 选项，读写的就是这个文件。
* ~/.gitconfig 文件：用户目录下的配置文件只适用于该用户。若使用 git config 时用 --global 选项，读写的就是这个文件。
* 当前项目的 Git 目录中的配置文件（也就是工作目录中的 .git/config 文件）：这里的配置仅仅针对当前项目有效。每一个级别的配置都会覆盖上层的相同配置，所以 .git/config 里的配置会覆盖 /etc/gitconfig 中的同名变量。

在 Windows 系统上，Git 会找寻用户主目录下的 .gitconfig 文件。主目录即 $HOME 变量指定的目录，一般都是 C:\Documents and Settings\$USER。

此外，Git 还会尝试找寻 /etc/gitconfig 文件，只不过看当初 Git 装在什么目录，就以此作为根目录来定位。

### 用户信息

配置个人的用户名称和电子邮件地址：

$ git config --global user.name "runoob"

$ git config --global user.email test@runoob.com

如果用了 **--global** 选项，那么更改的配置文件就是位于你用户主目录下的那个，以后你所有的项目都会默认使用这里配置的用户信息。

如果要在某个特定的项目中使用其他名字或者电邮，只要去掉 --global 选项重新配置即可，新的设定保存在当前项目的 .git/config 文件里。

### 文本编辑器

设置Git默认使用的文本编辑器, 一般可能会是 Vi 或者 Vim。如果你有其他偏好，比如 Emacs 的话，可以重新设置：:

$ git config --global core.editor emacs

### 差异分析工具

还有一个比较常用的是，在解决合并冲突时使用哪种差异分析工具。比如要改用 vimdiff 的话：

$ git config --global merge.tool vimdiff

Git 可以理解 kdiff3，tkdiff，meld，xxdiff，emerge，vimdiff，gvimdiff，ecmerge，和 opendiff 等合并工具的输出信息。

当然，你也可以指定使用自己开发的工具，具体怎么做可以参阅第七章。

### 查看配置信息

要检查已有的配置信息，可以使用 git config --list 命令：

$ git config --list

http.postbuffer=2M

user.name=runoob

user.email=test@runoob.com

有时候会看到重复的变量名，那就说明它们来自不同的配置文件（比如 /etc/gitconfig 和 ~/.gitconfig），不过最终 Git 实际采用的是最后一个。

这些配置我们也可以在 **~/.gitconfig** 或 **/etc/gitconfig** 看到，如下所示：

vim ~/.gitconfig

显示内容如下所示：

[http]

postBuffer = 2M

[user]

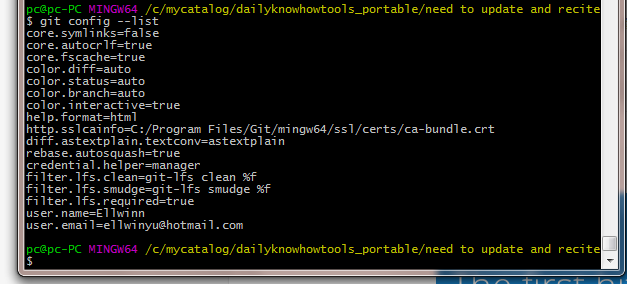
name = runoob

email = test@runoob.com

也可以直接查阅某个环境变量的设定，只要把特定的名字跟在后面即可，像这样：

$ git config user.name

runoob



git init

git init newrepo

pc@pc-PC MINGW64 /c/mycatalog/dailyknowhowtools\_portable

$ git init need\ to\ update\ and\ recite

Initialized empty Git repository in C:/mycatalog/dailyknowhowtools\_portable/need to update and recite/.git/

pc@pc-PC MINGW64 /c/mycatalog/dailyknowhowtools\_portable

$ git add \*.\*

fatal: Not a git repository (or any of the parent directories): .git

pc@pc-PC MINGW64 /c/mycatalog/dailyknowhowtools\_portable

$ cd need\ to\ update\ and\ recite

pc@pc-PC MINGW64 /c/mycatalog/dailyknowhowtools\_portable/need to update and recite (master)

$ git add \*.\*

初始化后，会在 newrepo 目录下会出现一个名为 .git 的目录，所有 Git 需要的数据和资源都存放在这个目录中。

如果当前目录下有几个文件想要纳入版本控制，需要先用 git add 命令告诉 Git 开始对这些文件进行跟踪，然后提交：

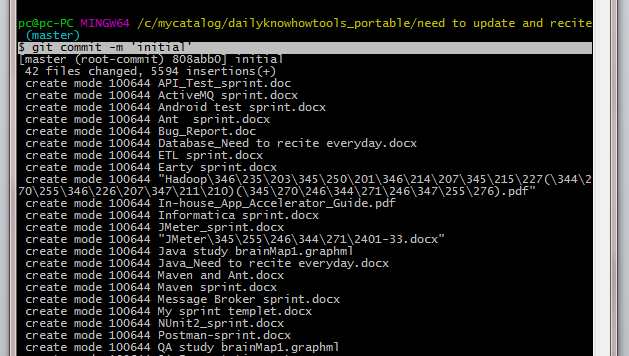
$ git add \*.\*

$ git add README

$ git commit -m '初始化项目版本'

$ git add \*.\*

$ git commit -m 'initial'



pc@pc-PC MINGW64 /c/mycatalog/dailyknowhowtools\_portable/need to update and recite (master)

$ git clone git://github.com/schacon/grit.git mygrit

Cloning into 'mygrit'...

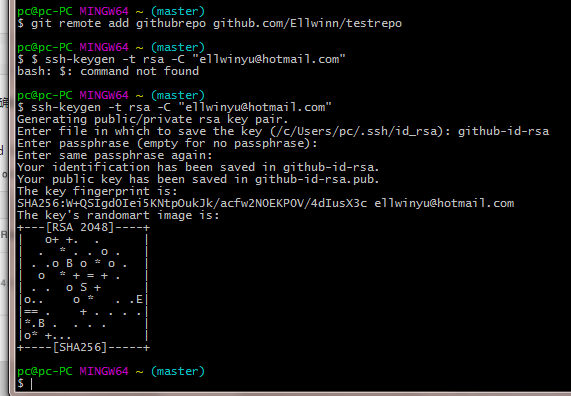
remote: Counting objects: 4051, done.

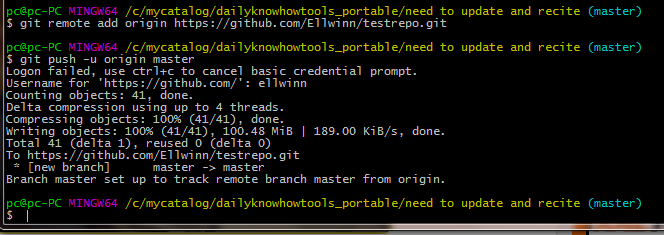
remote: Total 4051 (delta 0), reused 0 (delta 0), pack-reused 4051

Receiving objects: 100% (4051/4051), 2.30 MiB | 1.96 MiB/s, done.

Resolving deltas: 100% (1148/1148), done.

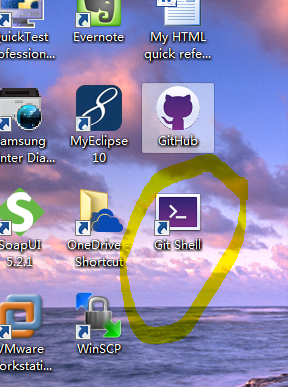
Checking connectivity... done.

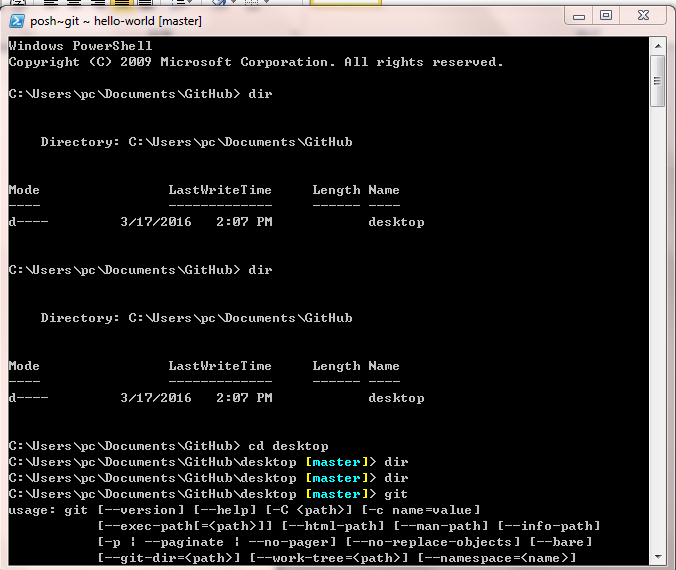


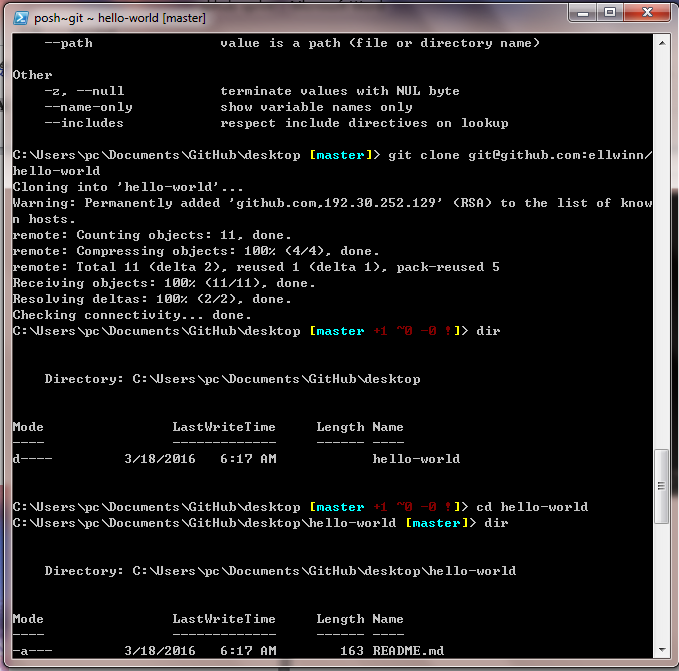


# My Activities:

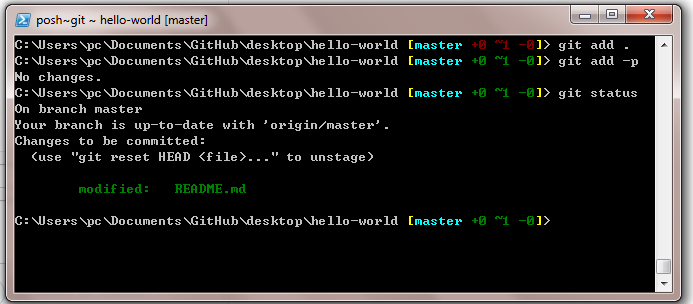
# Diagram







Open notepad++ to edit the file readme.md



# All excerpt