

git 安装:

Windows 平台上安装

在 Windows 平台上安装 Git 比较简单，有个叫做 `msysGit` 的项目提供了安装包，可以到 GitHub 的页面上下载 `exe` 安装文件并运行：

安装包下载地址：<http://msysgit.github.io/>



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。

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 "haha"
```

```
$ git config --global user.email test@haha.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。

#### 使用 github 开发流程:

1. 首先一个同学建立一个主仓库, 选定一个分支进行开发
2. 其他同学 fork 到自己的远程仓库
3. clone 到本地分支进行开发
4. 将代码 push 到自己的远程仓库
5. 发起 pull request 给主仓库, 让管理主仓库的同学看到你的代码的变化
6. 如果代码正确并且没有代码冲突, 管理主仓库的同学就会把代码 merge 到主仓库中