**1.4 安装 Git**

**在 Windows 上安装**

在 Windows 上安装 Git 同样轻松，有个叫做 msysGit 的项目提供了安装包，可以到 Google Code 的页面上下载 exe 安装文件并运行：

<https://git-for-windows.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。

 用 git 把 Git 项目仓库克隆到本地，以便日后随时更新：

**$ git clone git://git.kernel.org/pub/scm/git/git.git**

**1.5 初次运行 Git 前的配置**

一般在新的系统上，我们都需要先配置下自己的 Git 工作环境。

配置工作只需一次，以后升级时还会沿用现在的配置。当然，如果需要，你随时可以用相同的命令修改已有的配置。

Git 提供了一个叫做 **git config** 的工具（译注：实际是 git-config 命令，只不过可以通过 git 加一个名字来呼叫此命令。），**专门用来配置或读取相应的工作环境变量**。而正是由这些环境变量，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**

第一个要配置的是你个人的用户名称和电子邮件地址。这两条配置很重要，每次 Git 提交时都会引用这两条信息，说明是谁提交了更新，所以会随更新内容一起被永久纳入历史记录：

**$ git config --global user.name "John Doe"**

**$ git config --global user.email johndoe@ example.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**

接下来要设置的是默认使用的文本编辑器。Git 需要你输入一些额外消息的时候，会自动调用一个外部文本编辑器给你用。默认会使用操作系统指定的默认编辑器，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**

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**。

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 Scott Chacon**

**1.6 获取帮助**

想了解 Git 的各式工具该怎么用，可以阅读它们的使用帮助，方法有三：git

**$ git help**

**$ git --help**

**$ man git-**

比如，要学习 config 命令可以怎么用，运行：

**$ git help config**

我们随时都可以浏览这些帮助信息而无需连网。不过，要是你觉得还不够，可以到 Frenode IRC 服务器（irc.freenode.net）上的 #git 或 #github 频道寻求他人帮助。这两个频道上总有着上百号人，大多都有着丰富的 git 知识，并且乐于助人。