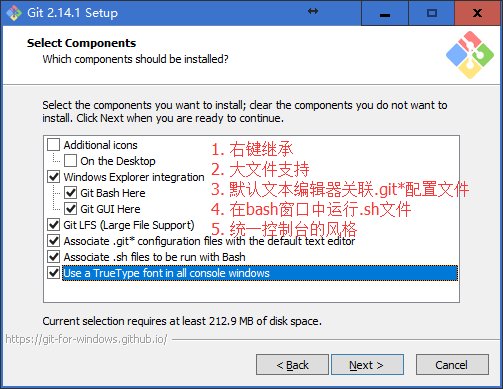
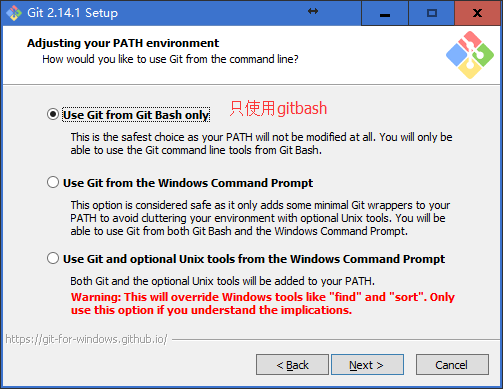
Git使用

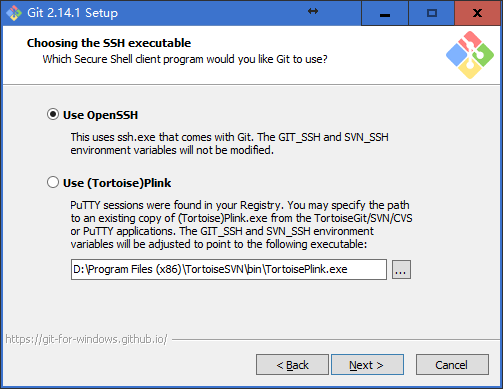
# 概述

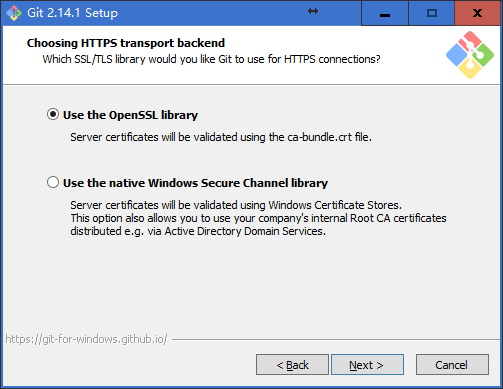
## Window下面的安装

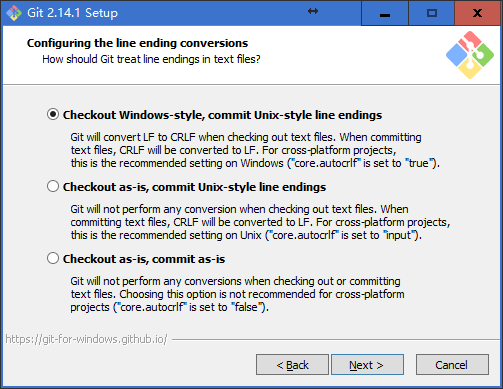
下面是比较重要的安装步骤，做简单说明。

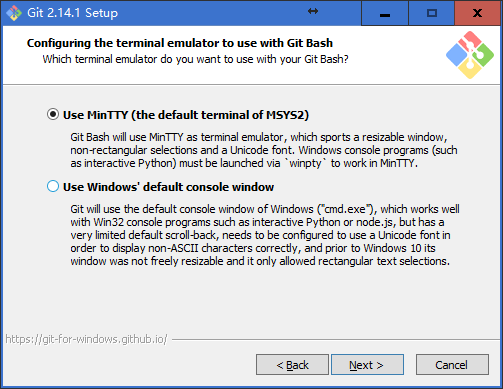


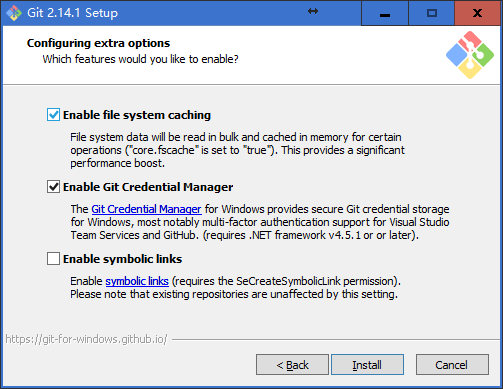












## 基本概念



### 工作区

### 暂存区

### 版本库

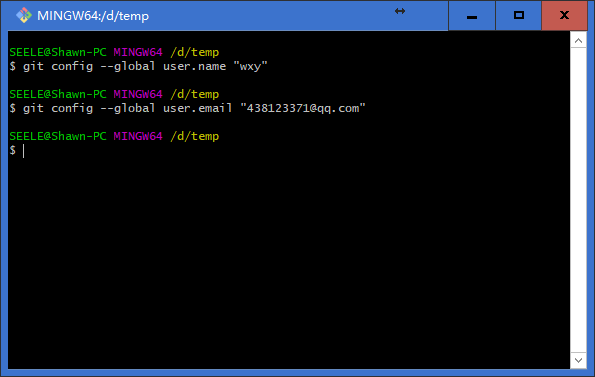
## 命令规律

可以参考Linux命令的使用。

# Git的基本操作

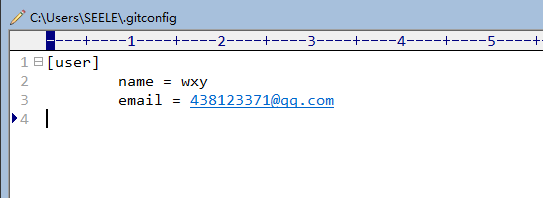
## 基本设置

在工作目录下面右键，打开GitBash，执行【git config –global user.name “yourName”】和【git config –global user.email “438123371@qq.com”】命令。



执行完成之后可以发现在用户目录下面生成一个.gitconfig文件。当然，如果以后想要修改这两个变量的值，也可以直接在这个文件中直接修改。

在C:\Users\SEELE目录下面生成.gitconfig文件

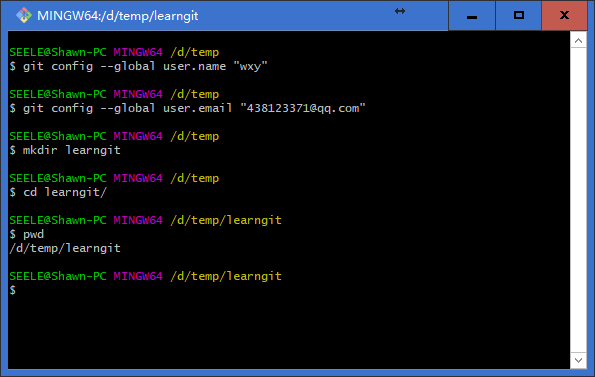


## 应用一

假如有一个工作场景：将文件夹learngit下的testGit.txt文件【testGit.txt文件中有一句“hello git”内容】上传到Git上面去。

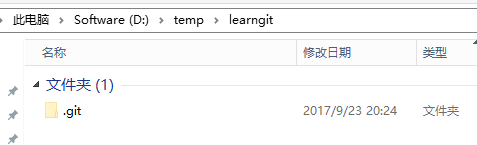
### 创建目录

在工作目录下面建立learngit文件夹，有两种方式，第一种方式使用GitBash的mkdir命令进行创建；第二种方式是直接使用系统创建。



### 创建版本库

在工作目录区域，执行【git init】命令，就可以初始化一个版本库。在初始化的地方，git会生成一个.git文件夹，默认是隐藏状态的，里面是Git版本库，不可以修改！



### 创建文件

直接使用编辑器创建一个testGit.txt文件，并在里面写上“hello git”。当然，也可以使用Linux上面的vi命令进行创建。

### 添加到暂存区

使用git add命令进行添加，将文件添加到暂存区。

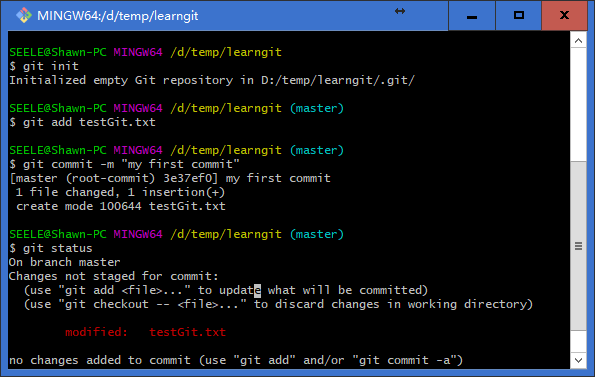
### 上传

使用git commit命令进行上传，将文件上传到版本库中。

### 查看暂存区和版本库的差别

使用git status命令查看暂存区和版本库之间的差别。

### 总结



### 查看状态

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git status  On branch master  Changes not staged for commit:  (use "git add <file>..." to update what will be committed)  (use "git checkout -- <file>..." to discard changes in working directory)  modified: testGit.txt  no changes added to commit (use "git add" and/or "git commit -a") |

### 查看历史

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git log  commit 3e37ef0a11324568c188b513467dbe19a71761ba (HEAD -> master)  Author: wxy <438123371@qq.com>  Date: Sat Sep 23 20:27:06 2017 +0800  my first commit |

## 应用二

假设另一个应用场景：对testGit.txt文件进行修改，添加一个行“hello git commit”，提交到版本库。

### 添加到暂存区

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git add testGit.txt  SEELE@Shawn-PC MINGW64 /d/temp/learngit (master) |

### 提交前查看状态

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git status  On branch master  Changes to be committed:  (use "git reset HEAD <file>..." to unstage)  modified: testGit.txt |

### 提交

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git commit -m "my second commit"  [master 6541d54] my second commit  1 file changed, 3 insertions(+), 1 deletion(-) |

### 提交后查看状态

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git status  On branch master  nothing to commit, working tree clean |

### 查看提交日志

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git log  commit 6541d547efc1307a656318b16af565319a18ccab (HEAD -> master)  Author: wxy <438123371@qq.com>  Date: Sat Sep 23 21:01:36 2017 +0800  my second commit  commit 3e37ef0a11324568c188b513467dbe19a71761ba  Author: wxy <438123371@qq.com>  Date: Sat Sep 23 20:27:06 2017 +0800  my first commit |

## 应用三

假设在testGit.txt文件中再次添加一行“test git again”，然后将文件添加到暂存区，并进行提交。提交完成之后再次修改testGit.txt文件，再最后再添加一行“git diff”。

### 添加到暂存区

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git add testGit.txt |

### 提交

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git commit -m "my third commit"  [master 34608c1] my third commit  1 file changed, 2 insertions(+) |

### 查看提交日志

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git log  commit 34608c1ebc20af9156d08f0336af82f1e1eb364b (HEAD -> master)  Author: wxy <438123371@qq.com>  Date: Sat Sep 23 21:20:10 2017 +0800  my third commit  commit 6541d547efc1307a656318b16af565319a18ccab  Author: wxy <438123371@qq.com>  Date: Sat Sep 23 21:01:36 2017 +0800  my second commit  commit 3e37ef0a11324568c188b513467dbe19a71761ba  Author: wxy <438123371@qq.com>  Date: Sat Sep 23 20:27:06 2017 +0800  my first commit |

### 工作目录与版本库差别

提交完成之后再在文件末尾添加一个一行“git diff”，然后使用【git diff】命令进行查看。

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git diff testGit.txt  **diff --git a/testGit.txt b/testGit.txt**  **index d1ab116..df4b2f0 100644**  **--- a/testGit.txt**  **+++ b/testGit.txt**  @@ -3,3 +3,7 @@ hello git!  hello git commit  test git again  +  +  +git diff  + |

也可以加参数：

|  |
| --- |
| SEELE@Shawn-PC MINGW64 /d/temp/learngit (master)  $ git diff HEAD -- testGit.txt  **diff --git a/testGit.txt b/testGit.txt**  **index d1ab116..df4b2f0 100644**  **--- a/testGit.txt**  **+++ b/testGit.txt**  @@ -3,3 +3,7 @@ hello git!  hello git commit  test git again  +  +  +git diff  + |

# 命令总结

|  |  |
| --- | --- |
| git config | 设置 |
| git init | 初始化版本库 |
| git add | 将文件添加到暂存区 |
| git commit | 提交文件到版本库 |
| git status | 查看版本库和暂存区的差别 |
| Git log |  |
| Git diff |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Git基础

## 获取git仓库

### 在空目录下初始化仓库

### 在已有目录下面初始化仓库

### 克隆已有的远程仓库

## 记录每次更新到仓库

### 工作区、暂存区、仓库概念

### 仓库中文件的状态

|  |  |
| --- | --- |
| Untracked | 未被追踪的文件。即从未添加到仓库中，只存在于工作目录的文件 |
| Unmodified | 跟仓库中没有差别文件。  1.从仓库克隆下来还没有进行过更新的文件；  2.从仓库中克隆下来经过更新又重新提交到仓库没有更新的文件 |
| Modified | 跟仓库有差别的还未添加到暂存区中的文件。 |
| Staged | 暂存区上面的文件。 |

### 检查当前文件状态

### 忽略文件

### 查看已暂存和未暂存的修改

git diff

git diff --cached

git diff --staged

### 提交更新

### 跳过使用暂存区直接提交

### 移除文件

#### 移除本地文件式

##### 未添加到暂存区

如果想要移除未添加到暂存区的文件时，使用git rm，然后使用git commit提交即可。

##### 已添加到暂存区

如果想要移除已经添加到暂存区的文件时，使用git rm –f，然后git commit提交即可。

#### 不移除本地文件式

$ git rm --cached README

### 移动文件

## 查看提交历史

## 撤销操作

### 合并式提交

如果某次提交时漏提交或多提交一些文件，可以使用git commit --amend

如漏提交：git commit、git add、git commit --amend，提交->添加到暂存->再次提交

如多提交：git commit、git rm、 git commit --amend，提交->删除暂存中多提交的文件->再次提交。

### 取消暂存文件

### 撤销对文件的修改

## 远程仓库的使用【git remote】

### 查看远程仓库

git remote

git remote -v

### 添加远程仓库

git remote add <shortname> <url>

拉取别人库中有但你自己库中没有的

$ git fetch [remote-name]

### 推送到远程仓库

git push [remote-name] [branch-name]

### 查看远程仓库

git remote show [remote-name]

### 远程仓库的移除和重命名

git remote rename

## 打标签【git tag】

标签它只是一个特定提交的引用。

## Git别名

https://git-scm.com/book/zh/v2/Git-%E5%88%86%E6%94%AF-%E5%88%86%E6%94%AF%E7%AE%80%E4%BB%8B

http://git.oschina.net/progit/1-%E8%B5%B7%E6%AD%A5.html