git使用总结

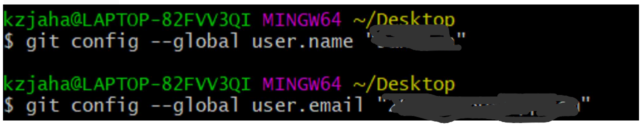
2021年4月21日 19:00

1.安装完git以后，打开gitbash，鼠标右键可以看到。

2.安装完成后，还需要最后一步设置，在命令行输入：

```
git config --global user.name "Your Name"
git config --global user.email "email@example.com"
```

注意空格！



```
kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop
$ git config --global user.name "[redacted]"

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop
$ git config --global user.email "[redacted]"
```

3.创建版本库（目录）：

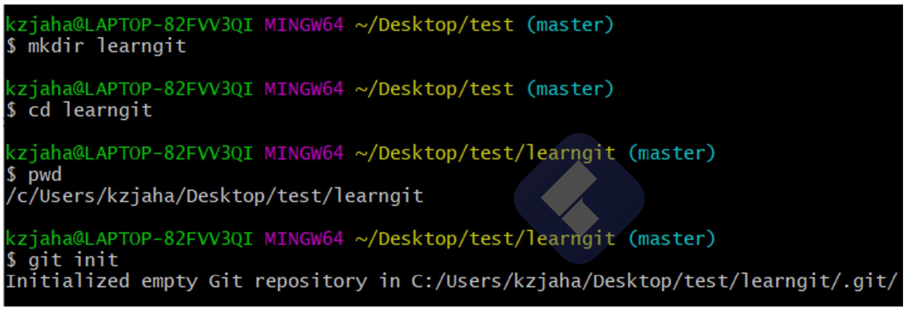
```
mkdir learngit
cd learngit
pwd
/Users/michael/learngit
```

pwd命令用于显示当前目录。这个仓库位于/Users/michael/learngit。

注意：Windows系统下目录不要含中文名；

然后，通过git init命令把这个目录变成Git可以管理的仓库：

```
git init
Initialized empty Git repository in /Users/michael/learngit/.git/
```



```
kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test (master)
$ mkdir learngit

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test (master)
$ cd learngit

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ pwd
/c/Users/kzjaha/Desktop/test/learngit

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git init
Initialized empty Git repository in C:/Users/kzjaha/Desktop/test/learngit/.git/
```

4.先在上面建立的仓库目录下用notepad++写一个.txt文件，然后再git上操作。

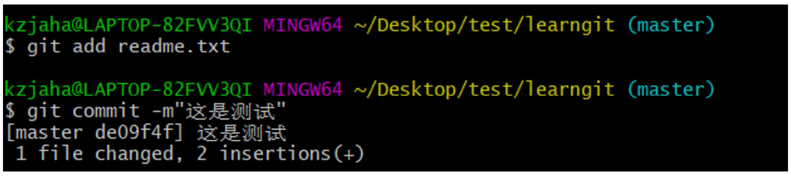
第一步，用命令git add告诉Git，把文件添加到仓库：

```
$ git add readme.txt
```

执行上面的命令，没有任何显示，这就对了，Unix的哲学是“没有消息就是好消息”，说明添加成功。

第二步，用命令git commit告诉Git，把文件提交到仓库：

```
git commit -m "wrote a readme file"
[master (root-commit) eaadf4e] wrote a readme file
 1 file changed, 2 insertions(+)
 create mode 100644 readme.txt
```



```
kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git add readme.txt

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git commit -m"这是测试"
[master de09f4f] 这是测试
 1 file changed, 2 insertions(+)
```

在我提交的时候没出现最后一句，不知道为什么，但明显已经提交成功了。

-m后面写的是解释，可以写任何东西，最好是解释，读起来方便。

git commit命令执行成功后会告诉你，1 file changed：1个文件被改动（我们新添加的readme.txt文件）；2 insertions：插入了两行内容（readme.txt有两行内容）。

为什么Git添加文件需要`add`，`commit`一共两步呢？因为`commit`可以一次提交很多文件，所以你可以多次`add`不同的文件，比如：

```
$ git add file1.txt
$ git add file2.txt file3.txt
$ git commit -m "add 3 files."
```

5.git status命令可以让我们时刻掌握仓库当前的状态，上面的命令输出告诉我们，readme.txt被修改过了，但还没有准备提交的修改。

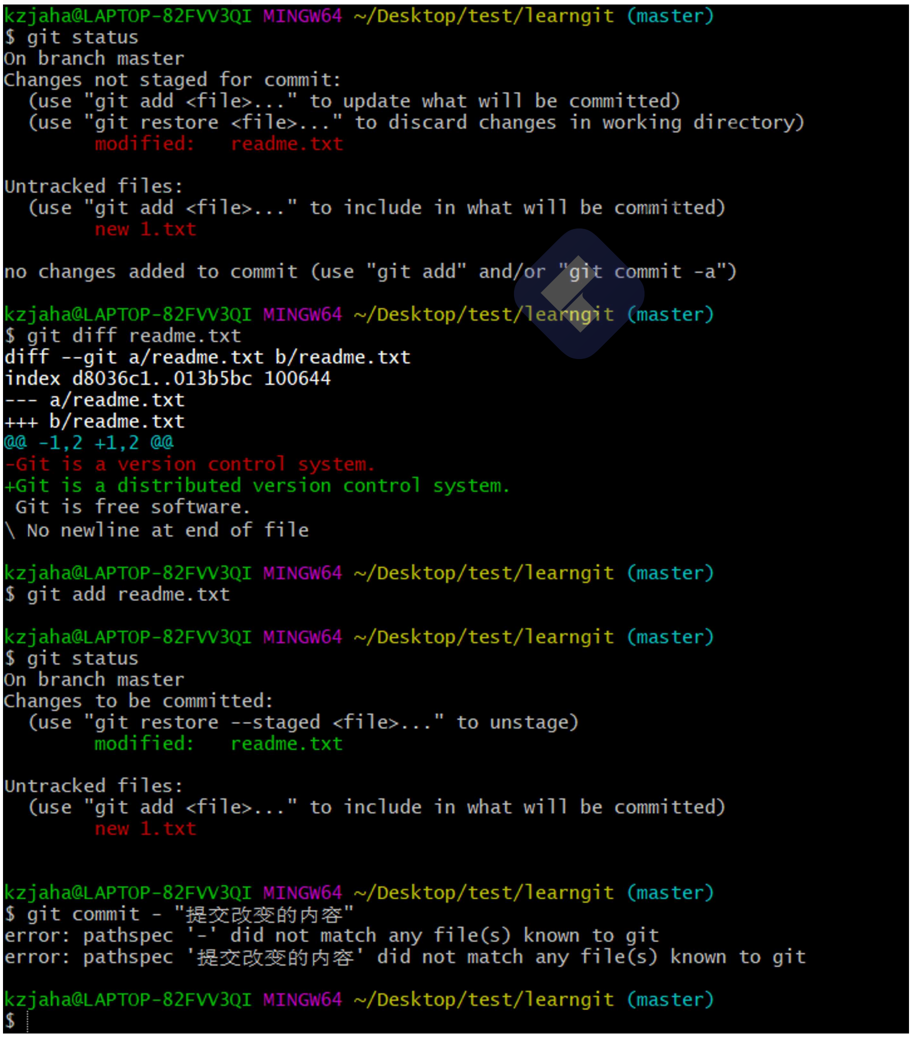
```
git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

虽然Git告诉我们`readme.txt`被修改了，但如果能看看具体修改了什么内容，自然是很好的。比如你休假两周从国外回来，第一天上班时，已经记不清上次怎么修改的`readme.txt`，所以，需要用`git diff`这个命令看看：

```
git diff readme.txt
diff --git a/readme.txt b/readme.txt
index 46d49bf..9247db6 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,2 +1,2 @@
-Git is a version control system.
+Git is a distributed version control system.
 Git is free software.
```



```
kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        new 1.txt

no changes added to commit (use "git add" and/or "git commit -a")

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git diff readme.txt
diff --git a/readme.txt b/readme.txt
index d8036c1..013b5bc 100644
--- a/readme.txt
+++ b/readme.txt
@@ -1,2 +1,2 @@
-Git is a version control system.
+Git is a distributed version control system.
 Git is free software.
\ No newline at end of file

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git add readme.txt

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        new 1.txt


kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$ git commit - "提交改变的内容"
error: pathspec '-' did not match any file(s) known to git
error: pathspec '提交改变的内容' did not match any file(s) known to git

kzjaha@LAPTOP-82FVV3QI MINGW64 ~/Desktop/test/learngit (master)
$
```

使用前面两个命令以后，我们可以重新提交以下，再用git status看看仓库下文件现状；