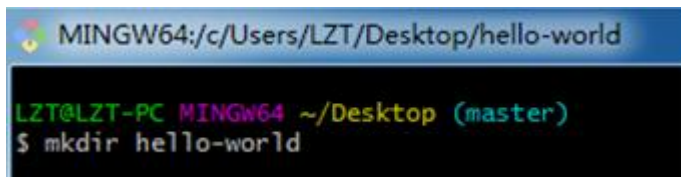


使用客户端创建本地仓库

1、创建一个本地 hello-world 仓库并初始化

在桌面上点击鼠标右键，选择 Git Bash here。注意，你的代码路径也应是纯英文的。

(1) 在命令行输入 `mkdir hello-world`，创建一个新文件夹。

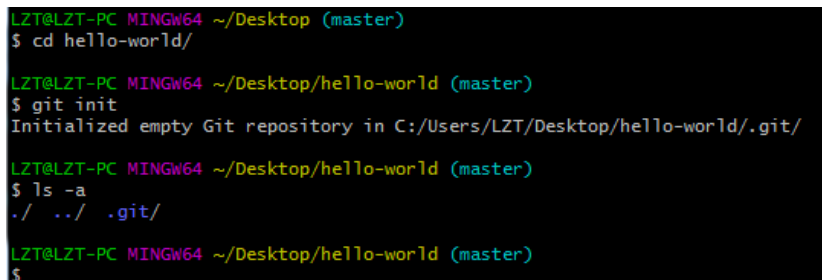


```
MINGW64:/c/Users/LZT/Desktop/hello-world
LZT@LZT-PC MINGW64 ~/Desktop (master)
$ mkdir hello-world
```

(2) 输入 `cd hello-world/` 进入新文件夹，注意在输入命令时，你可以用 `Tab` 键来自动补全。

(3) 输入 `git init` 初始化 Git 仓库(要使用 Git 进行版本管理，必须先初始化仓库)。此时用 `ls -a` 查看当前目录。如果初始化成功，执行了 `git init` 命令的目录下就会生成 `.git` 目录。此目录里存储着管理当前目录内容所需的仓库数据。

在 Git 中，我们将这个目录的内容称为“附属于该仓库的工作树”。文件的编辑等操作在工作树中进行，然后记录到仓库中，以此管理文件的历史快照。如果想将文件恢复到原先的状态，可以从仓库中调取之前的快照，在工作树中打开。开发者可以通过这种方式获取以往的文件。具体操作指令我们将在后面详细解说。



```
LZT@LZT-PC MINGW64 ~/Desktop (master)
$ cd hello-world/
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git init
Initialized empty Git repository in C:/Users/LZT/Desktop/hello-world/.git/
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ ls -a
./ ../ .git/
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$
```

注意，在此处使用的所有命令，如果你是在 Linux 环境下开发，用法都是完全一样的。所以对于完全没有 Linux 使用经验的学员，这也是一个开始接触 Linux 工作方式的好方法。

2、使用 `touch` 命令新建一个空文件，或者使用 `echo` 命令创建一份简单的说明文件，并提交到版本库中。

(1) 输入 `touch README.MD` 创建一个空文件，或者输入 `echo "This is a simple practise" > readme.txt`，创建一个 `readme.txt` 文件。你可以使用 `ls` 命令来查看当前目录下有哪些文件和文件夹。

```

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ touch README.MD

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git status
On branch master

No commits yet

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

        README.MD

nothing added to commit but untracked files present (use "git add" to track)

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ echo "This is a simple practise" >readme.txt

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ ls
README.MD  readme.txt

```

(2) 输入 `git status` 查看当前版本库状态，在 Untracked files(未跟踪文件)下，会出现红色的 README.MD 和 readme.txt，代表这两个文件还未被 Git 所管理。

```

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git status
On branch master

No commits yet

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

        README.MD
        readme.txt

nothing added to commit but untracked files present (use "git add" to track)

```

(3) 使用 `git add readme.txt`，将该文件加入缓冲区，如果你确定所有的修改都需要提交，可以使用 `git add .`来加入所有修改。现在用 `git status` 查看，将看到文件名变为绿色。

```

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git add readme.txt
warning: LF will be replaced by CRLF in readme.txt.
The file will have its original line endings in your working directory.

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   readme.txt

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

        README.MD

```

(4) 使用 `git commit -m "This is my first commit via Git!"`来提交修改，-m 后面所带的参数是本次提交信息，一般用来记录本次提交的主要意图。

```

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git commit -m "This is my first commit via Git"
[master (root-commit) 2b3940e] This is my first commit via Git
1 file changed, 1 insertion(+)
create mode 100644 readme.txt

```

(5) 提交成功后，可以用 `git log` 查看历史提交记录。每个记录都会有提交 id，作者和提交

日期。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git log
commit e0c84b56f197aafa1ded17016272a7c09379ecb3 (HEAD -> master)
Author: ZitingLou <louziting@163.com>
Date: Thu Nov 2 15:55:45 2017 +0800

    This is another file

commit 2b3940e763246978bc9baf1fc7e100ee550d257c
Author: ZitingLou <louziting@163.com>
Date: Thu Nov 2 15:39:03 2017 +0800

    This is my first commit via Git
```

如果只想让程序显示第一行简述信息，可以在 `git log` 命令后加上 `--pretty=short`。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git log --pretty=short
commit e0c84b56f197aafa1ded17016272a7c09379ecb3 (HEAD -> master)
Author: ZitingLou <louziting@163.com>

    This is another file

commit 2b3940e763246978bc9baf1fc7e100ee550d257c
Author: ZitingLou <louziting@163.com>

    This is my first commit via Git
```

如果只想显示该目录下的日志，可以在 `git log` 命令后加上目录名。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git log README.MD
commit e0c84b56f197aafa1ded17016272a7c09379ecb3 (HEAD -> master)
Author: ZitingLou <louziting@163.com>
Date: Thu Nov 2 15:55:45 2017 +0800

    This is another file
```

如果想查看提交所带来的改动，可以在 `git log` 命令后加上 `-p` 参数，文件的前后差别就会显示在提交信息之后。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git log -p
commit e0c84b56f197aafa1ded17016272a7c09379ecb3 (HEAD -> master)
Author: ZitingLou <louziting@163.com>
Date: Thu Nov 2 15:55:45 2017 +0800

    This is another file

diff --git a/README.MD b/README.MD
new file mode 100644
index 0000000..e69de29

commit 2b3940e763246978bc9baf1fc7e100ee550d257c
Author: ZitingLou <louziting@163.com>
Date: Thu Nov 2 15:39:03 2017 +0800

    This is my first commit via Git

diff --git a/readme.txt b/readme.txt
new file mode 100644
index 0000000..dc60c9d
--- /dev/null
+++ b/readme.txt
```

在 `-p` 参数后再加上文件名，可以只查看该文件的提交日志以及提交前后的差别。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git log -m readme.txt
commit 2b3940e763246978bc9baf1fc7e100ee550d257c
Author: ZitingLou <louzing@163.com>
Date: Thu Nov 2 15:39:03 2017 +0800

    This is my first commit via Git
```

3、使用 git diff 查看更改前后的差别

git diff 命令可以查看工作树、暂存区、最新提交之间的差别。

我们在刚刚提交的 README.MD 中写点东西(在本地打开 README.MD，加入 “This is a readme file”)。

执行 git diff 命令，查看当前工作树与暂存区的差别。其中 “+” 号标出的是新添加的行，被删除的行则用 “-” 号标出。可以看到，这次只添加了一行。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git diff
diff --git a/README.MD b/README.MD
index e69de29..88fa7a7 100644
--- a/README.MD
+++ b/README.MD
@@ -0,0 +1 @@
+This is a readme file
\ No newline at end of file
```

由于我们尚未用 git add 命令向暂存区添加任何东西，所以程序只会显示工作树与最新提交状态之间的差别。

用 git add 命令将该文件加入暂存区。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git add README.MD
```

如果现在执行 git diff 命令，由于工作树和暂存区的状态并无差别，结果什么都不会显示。

要查看与最新提交的差别，请执行 git diff HEAD 命令。这里的 HEAD 是指向当前分支中最新一次提交的指针。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git diff

LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git diff HEAD
diff --git a/README.MD b/README.MD
index e69de29..88fa7a7 100644
--- a/README.MD
+++ b/README.MD
@@ -0,0 +1 @@
+This is a readme file
\ No newline at end of file
```

执行 git commit 命令。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git commit -m "Add index"
[master a02bf7e] Add index
1 file changed, 1 insertion(+)
```

保险起见，我们查看一下提交日志，确定提交是否成功。

```
LZT@LZT-PC MINGW64 ~/Desktop/hello-world (master)
$ git log
commit a02bf7e7555d7f5874cb1578a45677e2c1f32f7d (HEAD -> master)
Author: ZitingLou <louziting@163.com>
Date: Thu Nov 2 16:19:11 2017 +0800

    Add index
```

成功查到了第二次提交。