

在本文章教程中，我们将演示如何查看 Git 存储库的文件和提交文件记录，并对存储库中的文件作修改和提交。

注意：在开始学习本教程之前，先克隆一个存储库，有关如何克隆存储库，请参考：[http://www.yiibai.com/git/git\\_clone\\_operation.html](http://www.yiibai.com/git/git_clone_operation.html)

在前面的文章中，都在要本地编写文件代码和提交，维护管制自己的文件版本，然后这种“自娱自乐”的方式，意义不是很大，在这里将介绍如何与其它的开发人员协同开发工作：每个开发人员都可以提交自己贡献的代码，并让其他人看到和修改。

要协同多人一起工作，可通过修改操作将代码文件最后一个确定版本提交，然后再推送变更。推送(Push)操作将数据永久存储到 Git 仓库。成功的推动操作后，其他开发人员可以看到新提交的变化。

执行 `git log` 命令查看提交的详细信息。最后一次提交的代码的提交 ID 是：  
`51de0f02eb48ed6b84a732512f230028d866b1ea`，如下所示 -

```
$ git log
commit 51de0f02eb48ed6b84a732512f230028d866b1ea
Author: your_name <your_email@mail.com>
Date:   Fri Jul 7 23:04:16 2017 +0800

    add the sum of a & b

commit be24e214620fa072efa877e1967571731c465884
Author: your_name <your_email@mail.com>
Date:   Fri Jul 7 18:58:16 2017 +0800

    ?? mark

commit 5eccf92e28eae94ec5fce7c687f6f92bf32a6a8d
Author: your_name <your_email@mail.com>
Date:   Fri Jul 7 18:52:06 2017 +0800

    this is main.py file commit mark use -m option

commit 6e5f31067466795c522b01692871f202c26ff948
Author: your_name <your_email@mail.com>
Date:   Fri Jul 7 18:42:43 2017 +0800

    this is main.py file commit mark without use "-m" option
```

```
commit 290342c270bc90f861ccc3d83afa920169e3b07e
Author: Maxsu <769728683@qq.com>
Date:   Fri Jul 7 16:55:12 2017 +0800
```

Initial commit

在推送(push)操作之前，如想要检查文件代码变化，可使用 `git show` 命令指定提交 ID 来查看具体的变化。

```
$ git show 51de0f02eb48ed6b84a732512f230028d866b1ea
commit 51de0f02eb48ed6b84a732512f230028d866b1ea
Author: your_name <your_email@mail.com>
Date:   Fri Jul 7 23:04:16 2017 +0800
```

add the sum of a & b

```
diff --git a/main.py b/main.py
index 657c8d0..25eb22b 100644
--- a/main.py
+++ b/main.py
@@ -3,5 +3,9 @@

 print ("Life is short, you need Python !")

-# this is a comment line

+a = 10
+
+b = 20
+c = a + b
+print("The value of c is ", c)
\ No newline at end of file
```

注意：每一行代码前面的 `-` 号和 `+` 号。`-` 号表示删除，`+` 号表示添加。如下 -

```
-# this is a comment line

+a = 10
+
```

```
+b = 20
+c = a + b
+print("The value of c is ", c)
```

如果对上面的提交修改没有疑义，则我们就可以将文件代码推送到远程存储库中，从而让其它开发人员可看查看和修改这些代码，现在就来看看怎么提交这些写好的代码，使用以下命令 -

```
$ git push origin master
```

上述命令将产生以下结果：

```
$ git push origin master
Username for 'http://git.oschina.net': 76972883@qq.com <输入帐号>
Password for 'http://76972883@qq.com@git.oschina.net': <输入登录密码>
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 1.20 KiB | 0 bytes/s, done.
Total 12 (delta 3), reused 0 (delta 0)
To http://git.oschina.net/yiibai/sample.git
    290342c..51de0f0  master -> master
```

在上面命令中，需要您提供(<http://git.oschina.net>)用户名和密码。

如上所示，现在代码已经成功地提交到了远程存储库(<http://git.oschina.net>)中了。要验证提交的结果，远程存储库中的内容是否是最后一次提交的信息，我们可以在另外一个空的目录中或在另外一台机器上使用 **git clone** 克隆出完整的文件代码，例如，在目录：

**E:\workspace** 下执行以下命令 -

```
$ git clone http://git.oschina.net/yiibai/sample.git
Cloning into 'sample'...
remote: Counting objects: 15, done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 3), reused 0 (delta 0)
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
```

在执行上面命令后，打开文件：**E:\workspace\sample\main.py**，其代码内容如下 -

```
#!/usr/bin/python3
#coding=utf-8

print ("Life is short, you need Python !")
```

```
a = 10  
  
b = 20  
c = a + b  
print("The value of c is ", c)
```

可以看到此文件与最后一个版本的内容一样。