git Cheatsheet

Contents

[git Cheatsheet 1](#_Toc392570177)

[Clone an existing repository and add a file to it 2](#_Toc392570178)

[Change a File in a Local Repository and Commit the Changes 5](#_Toc392570179)

# Clone an existing repository and add a file to it

### First, clone the repository to a local directory

### of your choice. In this case, I am cloning it to

### a directory I created called c:\gitrepo

C:\gitrepo>git clone https://github.com/BAPrettyfield/CheatSheets.git

Cloning into 'CheatSheets'...

remote: Counting objects: 6, done.

remote: Compressing objects: 100% (4/4), done.

remote: Total 6 (delta 2), reused 0 (delta 0)

Unpacking objects: 100% (6/6), done.

Checking connectivity... done.

### Now, if I take the dir of the gitrepo directory,

### I see that a copy of the github repository has been made.

### I also see that there is a CheatSheets sub folder which

### now exists in my local repository

C:\gitrepo>dir

Volume in drive C has no label.

Volume Serial Number is 7837-80BD

Directory of C:\gitrepo

07/08/2014 04:21 AM <DIR> .

07/08/2014 04:21 AM <DIR> ..

07/08/2014 04:22 AM <DIR> CheatSheets

0 File(s) 0 bytes

3 Dir(s) 340,697,288,704 bytes free

C:\gitrepo>cd CheatSheets

C:\gitrepo\CheatSheets>dir

Volume in drive C has no label.

Volume Serial Number is 7837-80BD

Directory of C:\gitrepo\CheatSheets

07/08/2014 04:22 AM <DIR> .

07/08/2014 04:22 AM <DIR> ..

07/08/2014 04:22 AM 153 README.md

1 File(s) 153 bytes

2 Dir(s) 340,725,723,136 bytes free

C:\gitrepo\CheatSheets>cd ..

### Next, I put the file I want to be tracked, into

### the CheatSheets directory I cloned on my local

### machine. This file is “Python Cheatsheet.docx”

### If I perform a git status, the on the CheatSheets

### repository, I see that it is in sync with the master

### repository on github, but there is a new untracked

### file cal

C:\gitrepo>cd CheatSheets

C:\gitrepo\CheatSheets>git status

On branch master

Your branch is up-to-date with 'origin/master'.

Untracked files:

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

Python Cheatsheet.docx

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

### Now use git add to add it to the local repository

### copy on my machine.

C:\gitrepo\CheatSheets>git add "Python Cheatsheet.docx"

C:\gitrepo\CheatSheets>git status

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: Python Cheatsheet.docx

### Now commit the new document. This commits

### it to the local repository

C:\gitrepo\CheatSheets>git commit -m "Added Python Cheatsheet.docx"

[master 70ef9d0] Added Python Cheatsheet.docx

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 Python Cheatsheet.docx

C:\gitrepo\CheatSheets>git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working directory clean

### Finally, push the committed file to the

### github repository

C:\gitrepo\CheatSheets>git push

warning: push.default is unset; its implicit value is changing in

Git 2.0 from 'matching' to 'simple'. To squelch this message

and maintain the current behavior after the default changes, use:

git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

git config --global push.default simple

When push.default is set to 'matching', git will push local branches

to the remote branches that already exist with the same name.

In Git 2.0, Git will default to the more conservative 'simple'

behavior, which only pushes the current branch to the corresponding

remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.

(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode

'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': BAPrettyfield

Password for 'https://BAPrettyfield@github.com':

Counting objects: 4, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 25.85 KiB | 0 bytes/s, done.

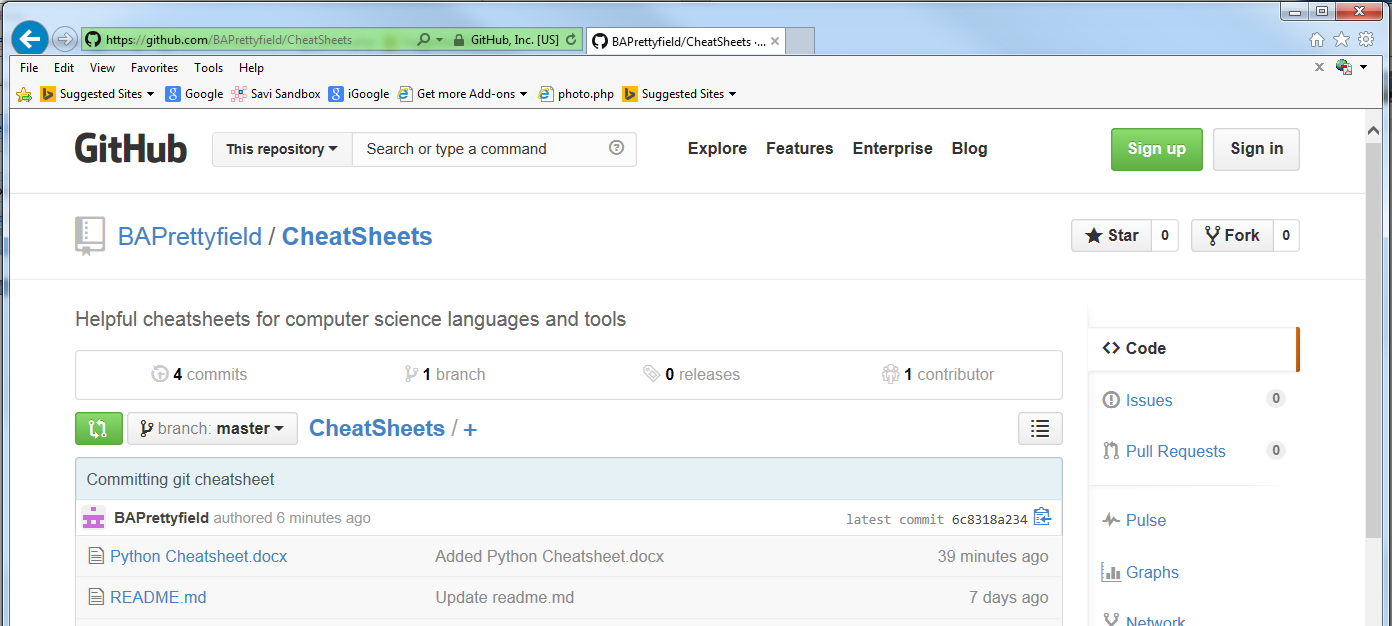
Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/BAPrettyfield/CheatSheets.git

d9c7f4f..70ef9d0 master -> master

C:\gitrepo\CheatSheets>

Now, if I refresh the github showing in the browser, I see that my new python Cheatsheet is updated with the new “Python Cheatsheet.docx” document:



# Change a File in a Local Repository and Commit the Changes

c:\gitrepo\CheatSheets>git status

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

deleted: git cheatsheet.txt

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: Python Cheatsheet.docx

c:\gitrepo\CheatSheets>git add -A

c:\gitrepo\CheatSheets>git commit -m "Update python cheatsheet"

[master 67b182b] Update python cheatsheet

2 files changed, 123 deletions(-)

rewrite Python Cheatsheet.docx (66%)

delete mode 100644 git cheatsheet.txt

c:\gitrepo\CheatSheets>git push

Username for 'https://github.com': BAPrettyfield

Password for 'https://BAPrettyfield@github.com':

Counting objects: 5, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 24.93 KiB | 0 bytes/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/BAPrettyfield/CheatSheets.git

e31b7e5..67b182b master -> master

c:\gitrepo\CheatSheets>