GIT Cheat Sheet

init — Create an empty GIT repository in your development directory

1. Go to directory
2. git init

status — Show the current state of the repository including un-added and un-committed files

- git status

add — Add a file to the repository staging area

1. Create a file, e.g. ***file.txt***
2. git add ***file.txt***

Add all new or changed files to the repository staging area (the period means all)

- git add **.**

commit — Commit all changes to the repository for first time (-m means message)

- git commit –m “***Initial commit***”

Commit all changes to the repository for later activities

- git commit –m “***Description of changes being made to project***”

branch — Create a new branch of the project

1. Choose a name for the new branch (original branch is ***master***), e.g. ***test***
2. git branch ***test***

List all branches (* appears next to current branch)

- git branch

checkout — Switch branches and check-out all files (e.g. to ***test*** branch)

- git checkout ***test***

Create a new branch and check-out files in one command

- git checkout –b ***test***

merge — Merge two branches together (go to the destination branch first, e.g. ***master***)

1. git checkout ***master***
2. git merge ***test***

(delete) — Delete a branch that you no longer need (e.g. after a merge)

- git branch ***test*** –d

Or to force the delete:

- git branch ***test*** –D

log — View commit history (including long commit ID numbers)

- git log

revert — Revert all files back to a previous commit point

- git revert <long commit ID from the **log** command>

rm -cache

- Removes cache from a specific file so that it can be added to the .gitignore

```
#Initialize git empty repository
git init

#Check files that are not yet added to the repository
git status

#Create a file to upload to repository
touch sample.txt

#Add a file to the repository by using
git add <file_name>

#To commit a file to repository
git commit -m 'my initial commit'

git status

#To create a git branch
git branch <branch name>

#To check all branches and current branch
git branch
#The * represents the branch you are on, it will commit to that branch once you checkin

# To switch to another branch we need to checkout the branch
git checkout <branch_name>

#To create and check out to a branch in one command:
git checkout -b <Branch Name>

#To delete a branch
git branch --delete <branch name>

#To check difference between 2 files
git diff

#To add all files in a folder
git add .

#To commit the updated files
git commit -m 'Files updated'

#To checkout another branch
git checkout master

#To merge a branch
git merge <branch name>

#To delete a branch
git branch <branch name> -d
```

```
#To force delete a branch
git branch <branch name> -D

#To check all our commits logs
git log
#Copy the commit id you want to revert to and run below command

To check difference between 2 git commits:
git diff
dcc7dfde2fa623c532e206f320226d10d444e8ce..ba1f00173170625853b04a5964dd7f447dd90
837

#To revert back to old version
git revert <commit id>
ctrl + X

#To ignore particular files
touch .gitignore
/secrets
passwords.txt

git commit -m 'added git ignore'

#If you want to remove a file from git and add it to gitignore
git rm --cached testing.txt
#Then add it to the gitignore file

######################
Git Hub Repository
######################
#Create a new repository on git hub by clicking on plus and choose new repository

#To add our repository to git hub
git remote add origin <URL of Repository>.git

#To push a file to the master
git push -u origin master

#To upload the updated file back to github
git add .
git commit -m <message>
git push -u origin master


#README is usually used for project setup details

#To clone a repository
git clone url
```

Compare specific file between two branches
In some cases, you may want to see all changes done to a specific file on the current branch you are working on.

In order to see the differences done to a file between two branches, use the “git diff” command, specify the two branches and the filename.

```
git diff master..feature -- <file>
```

To check difference between 2 git commits:

```
git diff dcc7dfde2fa623c532e206f320226d10d444e8ce..ba1f00173170625853b04a5964dd7f447dd90837
```

#Questions:
How do you initialize a git repository?

```
git init
```

How do I create a branch and switch to it?

```
git checkout -b <branch name>
```

Which commands should be used first after creating a new repository?

```
git init
git add .
git commit -m  "Initial Commit"
```

How to merge the branch you are currently in with another branch?

```
git merge <branch name>
```

To check commit id

```
git log
```

How do I revert my project to a previous commit?

```
git -revert <commit ID>
```