**GitHub Tutorial**

# Install the Git Client

* For Windows: <https://windows.github.com/>
* For Mac: <https://mac.github.com/>

## Storing your Credentials in Git

* Open Command Prompt / Terminal
* Execute the following commands:

git config –-global user.name “YOUR-GITHUB-USERNAME”

git config –-global user.email “YOUR-EMAIL”

### Storing Password – Windows

git config –-global credential.helper wincred

### Storing Password – Mac

git credential-osxkeychaing

* This will tell you if the osxkeychain credential helper is already installed. If you installed Git using Homebrew it will likely be installed already
* If the osxkeychain helper is missing, install it:

curl –s –o \https://github-media-downloads.s3.amazonaws.com/osx/git-credential-osxkeychain

chmod u+x git-credential-osxkeychain

sudo mv git-credential-osxkeychain \”$(dirname $(which git))/git-credential-osxkeychain”

Password: [YOUR-GIT-PASSWORD]

git config –-global credential.helper osxkeychain

# Creating A Repository (aka “Repo”): https://help.github.com/articles/create-a-repo/

* Log in to github
* On the right side of the nav-bar, click the plus sign / Create New / Repository
* Name the Repository
* Add a description (optional)
* Check Initialize with a README
* Click Create

## Committing a Change Directly on GitHub

* Click the README file
* Click the Edit icon
* Add some text the file and click Preview Changes. New content will be marked with green in the sidebar
* In the Commit Changes area, enter a short description of the changes (this is Required) and click Commit Changes

# Forking an Existing Repository (i.e. downloading a local copy of an application from GitHub) - <https://help.github.com/articles/fork-a-repo/>

* Open a browser tab to <https://github.com/octocat/Spoon-Knife>
* Click Fork. This copies the repository to your github account

## Create a Local Clone of your Fork

* Open a Windows Command Prompt / Mac Terminal Session
* Navigate to the directory where you want to keep a local copy of the application (use “cd” to change directories)
* Type the following command and hit Enter to download a local copy of the repo:

git clone <https://github.com/YOUR-USERNAME/Spoon-Knife>

## Configure Git to Automatically Download Changes from the Original Repo to your Local Copy

* In a browser tab, go to <https://github.com/octocat/Spoon-Knife>
* On the right sidebar, click the copy button beside the Clone URL box
* At the command prompt / terminal, to view the currently configured remote repo, type and Enter:

git remote –v

* Then type and Enter:

git remote add upstream [right-click and paste the URL you just copied here]

* To verify the new upstream repo, type and Enter again:

git remote –v

* Fetch the branches and their commits from the upstream repo. Your commits will be stored in a local branch “upstream/master”

git fetch upstream

* To make changes to your local copy, checkout the master branch:

git checkout master

* Merge changes from upstream/master to your local master branch (sync without losing local changes):

git merge upstream/master

# Uploading Local Changes and Additions to GitHub

* Open the README file in a text editor and add a random line of text
* Save the file
* Create a new text file called file1.txt and add some random text to it
* Save the file
* To ensure any new files (i.e. file1.txt and potentially any others not part of the original forked repo) are uploaded to github, execute: (-A stands for all. If you only want to add a certain file(s) just add them specifically by name in separate git add commands)

git add –A

* Before you can upload, you need to create a new COMMIT
* A commit is 1 snapshot or archive of your code repository
* Each commit must include a commit message that is generally used to describe the round of changes made: (-m stands for “message”)

git commit –m “changed Readme and added file1”

* This creates a commit but does not automatically upload to github
* To upload, run:

git push origin master

* Go back to your repo’s web page on github.com
* You will see your new commit with a uniquely generated Commit ID, push timestamp, and the modified Readme and new file1.txt
* Try creating file2.txt with some random text, then adding, committing and pushing the new commit to github

# Further Reading:

* <https://www.codeschool.com/courses/try-git> (requires Free registration)

## Windows Users

* <https://help.github.com/categories/bootcamp/>

## Mac Users

* <http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1>
* <http://readwrite.com/2013/10/02/github-for-beginners-part-2>