# Facebook iOS University

## Submitting Assignments

This page explains the steps for submitting assignments via Github.

1. \*\*GitHub Repository\*\* => Create a new GitHub repository for each week's assignment.

\* \*\*Note\*\*: Do not create a new repository if the assignment is a continuation of the same project from the previous week. In this case it's preferred to use the same repository for all weeks of the assignment.

\* Add a [.gitignore](https://raw.githubusercontent.com/github/gitignore/master/Swift.gitignore) to your project to avoid storing compiled and local files in git.

\* If needed, review how to work with Git and GitHub with [this handy tutorial on using Git from the command line](https://guides.codepath.com/ios/Using-Git-with-Terminal).

2. \*\*README Template\*\* => Every submission must be accompanied by a `README.md`using the readme template provided with the assignment.

\* In your project repository, add a `README.md` file in the root directory that contains the contents of the README template for that assignment.

\* If the assignment is a continuation of the same project from a preview week, add the new template to the end of the previous README instead of creating a new file.

\* Make sure to check off the user stories you've completed.

3. \*\*GIF Walkthrough\*\* => Your `README.md` must also contain a GIF walkthrough using [LiceCap](https://www.cockos.com/licecap/) of the app demonstrating how it works with the user stories completed. [Imgur](https://imgur.com/) is a great service for hosting the GIF walkthrough and then linking to it from the README.

\* When using Imgur, you can right-click on the gif and choose "Copy Image Address" to get the correct address. Make sure the address has a `.gif`extension. If you end up with a url that has a `.gifv` extension, removing the `v` and changing this to `.gif` will ensure the gif renders on GitHub.

4. \*\*Make sure you've pushed all your latest code up to GitHub\*\* => To check this, you can browse your repository on GitHub to make sure some of your latest changes are present there.

5. \*\*Submit Assignment\*\* => Once you have \*\*completed all required user stories and added a README as described above\*\* then you are ready to notify us that you are ready for an instructor review, go to the "Assignment Tab" for the project and click the "Submit" button on the right-hand side:

![](https://i.imgur.com/6d66toL.png)

In the dialog that appears, enter the required information about your project:

![](https://i.imgur.com/HeCHd64.png)

Then press "Submit" at the bottom to finalize your submission. Note that \*\*you can always update your submission at any time\*\* in case you want to re-upload the GIF or update the hours spent.

\*\*NOTE\*\*: If you recorded your app walkthrough as a video file instead of a GIF then please use LiceCap to record a preview of the full video included in the project

README.

## Ensuring your Submission is Valid

To ensure you get credit for your submission, review the following items:

1. Can you see your \*\*latest code\*\* on GitHub?

2. Does your README use the provided \*\*readme template\*\*?

3. When going to your main \*\*repository page\*\* on GitHub, can you see your README and does your \*\*GIF walkthrough\*\* start playing?

4. Does your app have all of the required User Stories?

## Assignment Scoring

Check out the project rubric for a detailed break down of how assignments are evaluated.

## Terminal Quickstart

\*\*Starting the Terminal\*\*

A quick primer on the terminal can be found in this guide. Open up the \*\*terminal application\*\* on your Mac by opening up your `Applications` folder, then open the `Utilities` folder:

![](https://i.imgur.com/Ni5VrjX.png)

You can also use spotlight pressing "Cmd + Space" and then typing "Terminal". Now inside the terminal, we need to switch to your project folder, such as `~/Desktop/iOSTipCalculator` :

```

cd /path/to/my/project/code

```

Next, make sure the files look correct and the folders are listed from your app:

```

ls

```

## Repository URLs

There are two types of repository URLs, [HTTPS and SSH](https://help.github.com/articles/which-remote-url-should-i-use/):

\* HTTPS: `https://github.com/homer/duff\_project.git`

\* SSH: `git@github.com:homer/duff\_project.git`

The general workflow is the same for both URLs, but there are some differences in the specifics of commands.

## SSH Setup

You'll first need to [register your machine's SSH keys](https://help.github.com/articles/connecting-to-github-with-ssh/) with your Github account. You only need to perform this step \*\*once\*\* for each machine you use.

Once you've added your SSH key, the SSH repository URL will look like `git@github.com:myusername/reponame.git` but for your username and project. You

won't need to provide credentials (username/password) for any further git commands.

## HTTPS Setup

There is no additional setup required for using HTTPS. However, you'll need to enter your username and password each time you run the `git pull` or `git push` commands:

```

git push origin master

<enter your username at promt>

<enter your password at prompt>

```

## Pushing Code

To push your code, simply initialize your git repository and then connect the repository to your local project:

```

$ git status

```

If the repository \*\*is not setup\*\* , then create the git repository with:

```

$ git init

```

\*\*Setup a [.gitignore](https://raw.githubusercontent.com/github/gitignore/master/Swift.gitignore) in the root of the directory\*\* which prevents certain files from being committed.

```

$ touch .gitignore

(...fill in .gitignore with contents in the link above...)

```

Check if you \*\*already have an origin\*\* with the following command:

```

$ git remote -v

```

If the above command prints something out like below, then you have an existing origin connected.

```

origin [git@github.com:homer/duff\_project.git](mailto:git@github.com) (fetch)

origin git@github.com:homer/duff\_project.git (push)

```

Note the above URL is SSH, but it may also be an HTTPS URL.

You can remove the existing origin with:

```

$ git remote rm origin

```

Next, you want to add the remote repository for your Github repo:

\*\*HTTPS:\*\*

```

$ git remote add origin https://github.com/myusername/reponame.git

```

(remember to replace `myusername` and `reponame` with your user name and repo, respectively).

\*\*SSH:\*\*

```

$ git remote add origin git@github.com:myusername/reponame.git

```

(remember to replace `myusername` and `reponame` with your user name and repo, respectively).

Since you connected a new origin, you'll need to make sure your local repository is synchronized with the remote repository (`origin`). That is, you want to make sure your local repository has all of the files that are in the remote repository. You can do this with the pull command:

```

$ git pull origin master

```

In the above command `origin` is the name of the remote repo (which we added with `git remote add ` step, and `master` is the name of the branch.

Sometimes the `git pull` command will need to merge updates from the remote repository into your local repository. As long as there are no conflicts, you don't need to worry about this automatic step. The `git pull` command may open a vi editor asking you to provide a comment. If you don't know vi, you can

quit entering the key commands `:wq`.

You can now add the files to git using the SourceTree client or by typing:

```

$ git add.

$ git commit -am "Initial commit"

```

and now go ahead and push the code to Github with:

```

$ git push origin master

```

If you use the command:

```

$ git push -u origin master

```

then you won't need to provide the `origin master` arguments anymore for this repository. The `-u` flag tells git to set `origin master` as the defaul repository and branch for all further commands, including `git pull`.

You can also use your favorite Git GUI (for example [SourceTree](https://www.sourcetreeapp.com/)) to do a lot of this process as well.

## Git References

Additional references for Git:

\* [Official Git Tutorial](https://git-scm.com/docs/gittutorial)

\* [Tensory's HowToGit](https://github.com/tensory/HowToGit/blob/master/README.md)