

# USK00 - Let's Git!

---

Last edited on : Thu Mar 12 14:45:49 +06 2020 by @yaakovazat

Author: Yaakov Azat

UID: 0f40b3be-643e-11ea-bc55-0242ac130003

Branch: Alpha

## License

---

This work by Yaakov Azat is licensed under "Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)" license and you are free to copy and redistribute the material in any medium or format, remix, transform, and build upon the material under the following terms:

1. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
2. NonCommercial — You may not use the material for commercial purposes.
3. You should add credits for this material anywhere you share as this is a material part of "UniSat" or "UniSat Software Kit".

For further information about the license, please refer to the license page: [License](#)

## Preface

---

### Let me tell you a story

I know that you have a lot of questions about this material (I suppose that you are a curious super girl~), one of them maybe, "What is Git and why we use it?". Before I tell you more about Git and Github, I wanna talk to you about a recent story of mine.

Before I tell you more about Git and Github, I wanna talk to you about a recent story of mine. The time was last year, I was preparing my master thesis. I did a lot of research and written almost a hundred pages of the article, then all of a sudden, my computer shut down, I was focusing too much on the work and hadn't found that the computer is already out of battery! Then, it's not hard to imagine, I lost my last changes to the article, not all of them but almost latest 10 pages. After that, I opened my computer, open the "Autosave" feature of the Microsoft Office, but that does not solve all of my problems:

1. the ability to save all my changes to the document that I am editing.
2. the ability to edit the same document with my friends.
3. the ability to label a change
4. the ability to give a detailed explanation of why a change was made
5. the ability to move between different versions of the same document
6. the ability to undo change A, make edit B, then get back change A without affecting edit B

The version control tool, Git, can do all of those things - *and more!!!* (bet you didn't see *that* coming!) So have I sold you yet on the awesomeness that is Git? I hope so, cause we're about to dive into it in the next section.

## Quick Start

---

### About Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. [1](#)

### Download and Install Git

You can check your git version by typing the command below on your terminal (You can check wheather you have installed Git or not by this command too):

```
git --version
```

If you see something like:

`git version 2.21.1 (Apple Git-122.3)` on Mac OS or `git version 2.17.1` on your linux computer, then it means `git` has already been installed, if not, please refer the installation instructions below:

### Linux

**Note :** We recommend you to use Ubuntu, Raspbian OS, Kali Linux or Other Debian Based Linux Distributions for this class.

Open your favorite terminal and type:

```
sudo apt install git
```

### Mac OS

There are several means to install `git` on Mac:

#### 1. GUI installation :

1. Open <https://git-scm.com/> and click `Download 2.23.0 for Mac' :



2. Open your downloads folder and double click to open the `git-2.23.0-intel-universal-mavericks.dmg` file you just downloaded.
  3. The install process is the same as other programmes and ignored
- ## 2. Terminal installation with brew

1. Install `brew`: paste these commands to your terminal and click `Enter`

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

2. Install git with `brew`, type `brew install git` in your terminal and press `Enter`

## Windows

To install Git on windows:

1. open <https://git-scm.com/download/win> on your browser and git should downloaded automatically, if not, click `Click here to download manually` as below:

## Downloading Git



### Your download is starting...

You are downloading the latest (**2.25.1**) **32-bit** version of **Git for Windows**. This is the most recent **maintained build**. It was released **22 days ago**, on 2020-02-19.

**Click here to download manually**, if your download hasn't started.

### Other Git for Windows downloads

**Git for Windows Setup**

**32-bit Git for Windows Setup.**

**64-bit Git for Windows Setup.**

**Git for Windows Portable ("thumbdrive edition")**

**32-bit Git for Windows Portable.**

**64-bit Git for Windows Portable.**

The current source code release is version **2.25.1**. If you want the newer version, you can build it from [the source code](#).

2. When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.

### 3. Select Git Bash during installation

That's all for installing git.

## Configuring Git

Configure user information for all local repositories:

1. To set the email you want attached to your commit transactions:

```
git config --global user.name "[name]"
```

for example I will do this on my computer:

```
git config --global user.name "Yaakov Azat"
```

2. To set the email you want attached to your commit transactions:

```
git config --global user.email "[email address]"
```

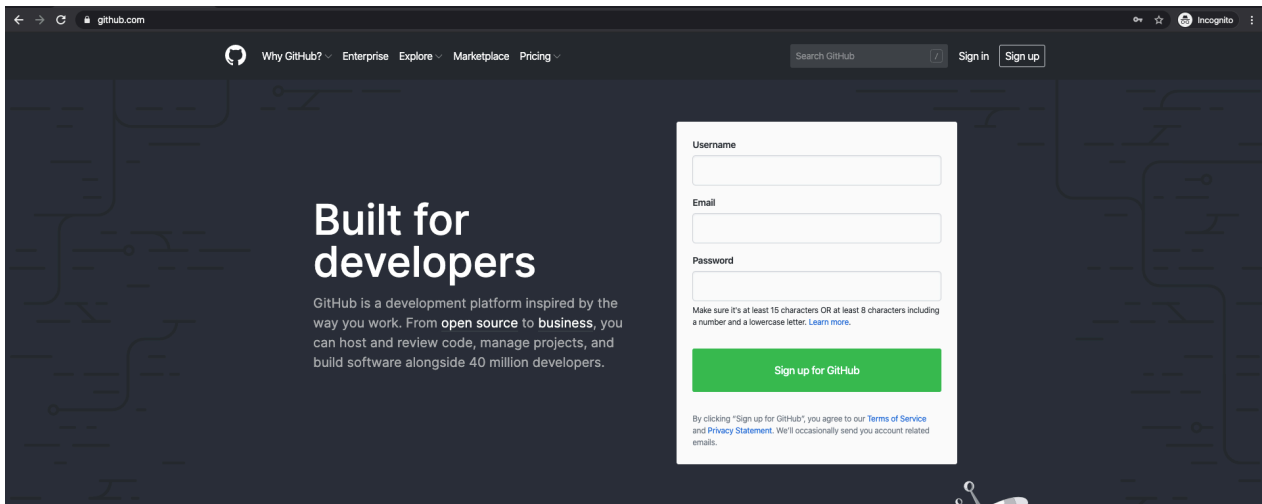
for example, I will do this for myself:

```
git config --global user.email "yaakovazat@gmail.com"
```

## Registering on Github

Github is a platform that provides hosting for software development version control using Git (We will talk about this very in detail later on the course.

- To register your Github account, first you should open Github homepage (<https://github.com/>) on your browser.



- Click `sign up` on the right corner for registration.

Join GitHub

# Create your account

**Username \***

**Email address \***

**Password \***

Make sure it's **at least 15 characters** OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)


**Email preferences**

☒ Send me occasional product updates, announcements, and offers.

**Verify your account**

Please solve this puzzle so we know you are a real person

[Verify](#)



[Select a plan](#)

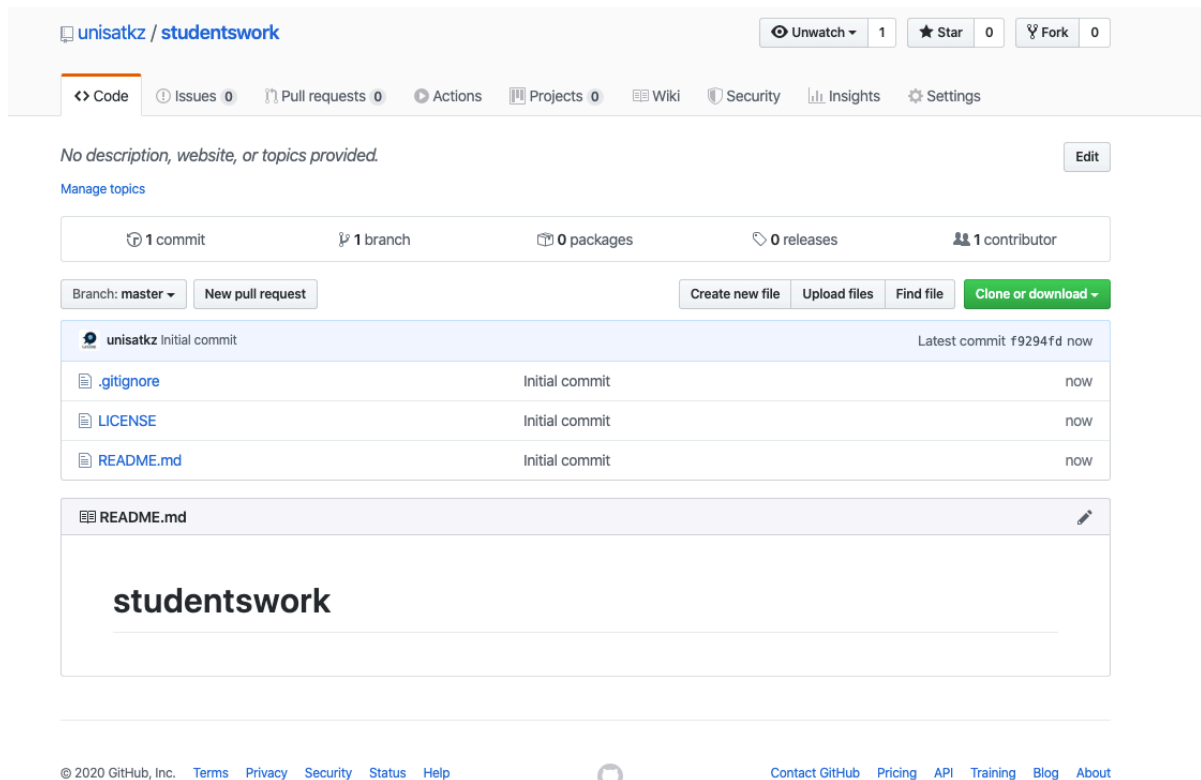
- Pick a nice `username`, enter your email and fill the password.

- Next, you need to verify your identity.
- Then, finally, you can click `Select a plan`
- Select the **free** account and that's all.

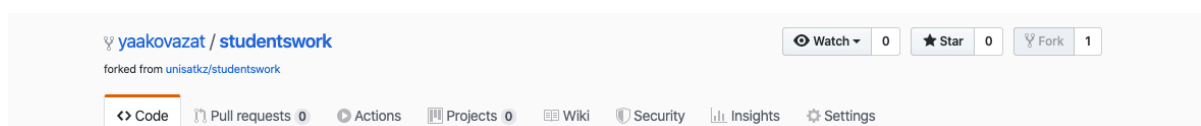
## Using Git

### Fork the `unisat` repo:

- Open <https://github.com/unisatkz/studentswork> on your browser.



- Click the star icon and then click `fork`.



### Forking unisatkz/studentswork

It should only take a few seconds.

[Refresh](#)



### Clone the `unisat` Repo:

To submit your content to the repository, you should first clone the repository to your local computer, to do this, open your terminal (or git bash) then type:

```
git clone https://github.com/[yourgithubusername]/studentswork
```

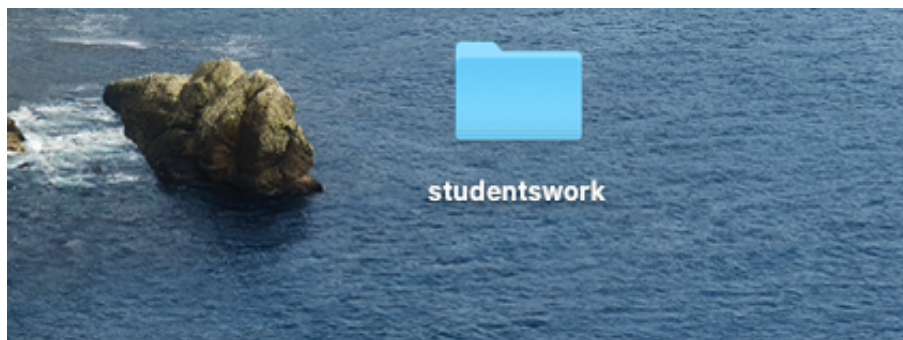
!!! YOU SHOULD REPLACE [yourgithubusername] ABOVE WITH YOURS !!!

for me, I will do this as below:

```
git clone https://github.com/yaakovazat/studentswork
```

This will clone the repository to your desktop:

```
azat@Azats-Mac-Pro ~/Desktop$ git clone https://github.com/yaakovazat/studentswork
Cloning into 'studentswork'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
azat@Azats-Mac-Pro ~/Desktop$ |
```



Open the repository folder, drag or copy your documents in to the `studentswork` folder,



Add all of your documents that should be added

## Add the changes.

After putting all the stuff inside the repository folder `studentswork`, now you are able to push it to your Github, to do so, **first enter the `studentswork` folder in your terminal** (search how to enter a folder in Windows/Linux/Mac terminal if you don't know), and then type this command:

```
git add .
```

## Comment the changes

You can comment your changes, that means your changes now meaningful with some labels so that you can easily understand whats done for your documents/codes later on.

To do this, type this:

```
git commit -m 'Any thing to comment for the change'
```

I will write this for example:

```
git commit -m 'first commit using github'
```

## Push the changes

To save the changes on your Github account, you should push the repository with this command:

```
git push -u origin master
```

```
azat@Azats-Mac-Pro ~/Desktop > cd studentwork
azat@Azats-Mac-Pro ~/Desktop/studentwork > master git add .
azat@Azats-Mac-Pro ~/Desktop/studentwork > master + git commit -m 'first git comment on unisat repo'
[master d254a71] first git comment on unisat repo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 demo.png
azat@Azats-Mac-Pro ~/Desktop/studentwork > master git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 169.57 KiB | 28.26 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/yaakovazat/studentwork
 f9294fd..d254a71 master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
azat@Azats-Mac-Pro ~/Desktop/studentwork > master |
```

## Pull Request

After finishing the steps above, you should be able to upload all of your publishing documents to the Github repository. In the next step, we are going to contribute and share our changes with other `unisat` team members, to do so, we should create a `pull request`

- open your Github account and open the repository `studentwork`



yaakovazat / studentswork  
forked from unisatzk/studentswork

Watch 0 Star 0 Fork 1

Code Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

2 commits 1 branch 0 packages 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

This branch is 1 commit ahead of unisatzk:master. Pull request Compare

yaakovazat first git comment on unisat repo Latest commit d254a71 3 minutes ago

.gitignore	Initial commit	20 minutes ago
LICENSE	Initial commit	20 minutes ago
README.md	Initial commit	20 minutes ago
demo.png	first git comment on unisat repo	3 minutes ago

README.md

# studentswork

- click on **New pull request** button:

unisatzk / studentswork

Watch 1 Unstar 2 Fork 1

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: unisatzk/studentswork base: master head repository: yaakovazat/studentswork compare: master

✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Mar 12, 2020

yaakovazat first git comment on unisat repo d254a71

Showing 1 changed file with 0 additions and 0 deletions. Unified Split

BIN +170 KB demo.png

Binary file not shown.

No commit comments for this range

This will send our changes to the main repository.

- Click the green 'Create pull request' button:

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: unisatzk/studentswork base: master head repository: yaakovazat/studentswork compare: master

✓ **Able to merge.** These branches can be automatically merged.

first git comment on unisat repo

Write Preview

AA B i “ < > ↻ ⋮ ⋮ ⋮ @ 📎 ↶

I did some changes on the repository and so .....

Attach files by dragging & dropping, selecting or pasting them.

✓ **Allow edits from maintainers.** [Learn more](#)

Create pull request

Helpful resources  
[GitHub Community Guidelines](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Mar 12, 2020

yaakovazat first git comment on unisat repo d254a71

Showing 1 changed file with 0 additions and 0 deletions.

Unified Split

Write some comments if you want, after finish typing, just click **Create pull request** again:

first git comment on unisat repo #1

Edit

Open yaakovazat wants to merge 1 commit into unisatzk:master from yaakovazat:master

Conversation 0 Commits 1 Checks 0 Files changed 1 +0 -0

yaakovazat commented now

I did some changes on the repository and so .....

first git comment on unisat repo d254a71

Add more commits by pushing to the **master** branch on **yaakovazat/studentswork**.

✓ **This branch has no conflicts with the base branch**  
Only those with [write access](#) to this repository can merge pull requests.

Write Preview

AA B i “ < > ↻ ⋮ ⋮ ⋮ @ 📎 ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close pull request Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

ProTip! Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.

Reviewers  
No reviews

Assignees  
No one assigned

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

Linked issues  
Successfully merging this pull request may close these issues.  
None yet

Notifications  
Customize  
Unsubscribe  
You're receiving notifications because you authored the thread.

That's all!

If you are lucky, the UniSat Github repository admin will soon reply to your comments and merge your work to the main repository.

NOTICE: THIS IS ONLY THE PART OF THE WHOLE WORK AND PUBLISHED AS A BETA. YOU DO NOT HAVE TO WORRY IF YOU were NOT ABLE TO FINISH THESE TASKS, YOU WILL BE taught VERY CONCRETEly ON THE CLASS.