

The background is a dark blue gradient. It features several vertical white lines of varying lengths scattered across the frame. Interspersed among these lines are small squares in three colors: pink, orange, and teal. Some squares are solid, while others are outlined. The overall aesthetic is modern and minimalist.

GIT & GITHUB WORKSHOP

A SMALL INTRODUCTION

WHAT IS GIT ?

- Git is a distributed version-control system tracking source code changes during software development.
- It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.
- Its goals include speed, data integrity, and support for distributed, non-linear workflows.



WHY SHOULD I CARE ABOUT IT?



You can keep track of changes in your file/project.

- That means no more `last.c`, `lastlast.c`, `truelast.c` etc

You can have different workflows on the same project, without worrying about collateral damage (up to a certain point of course)

- More on that later...

HOW IT WORKS THEN?

1. First, you start by initializing the repository.

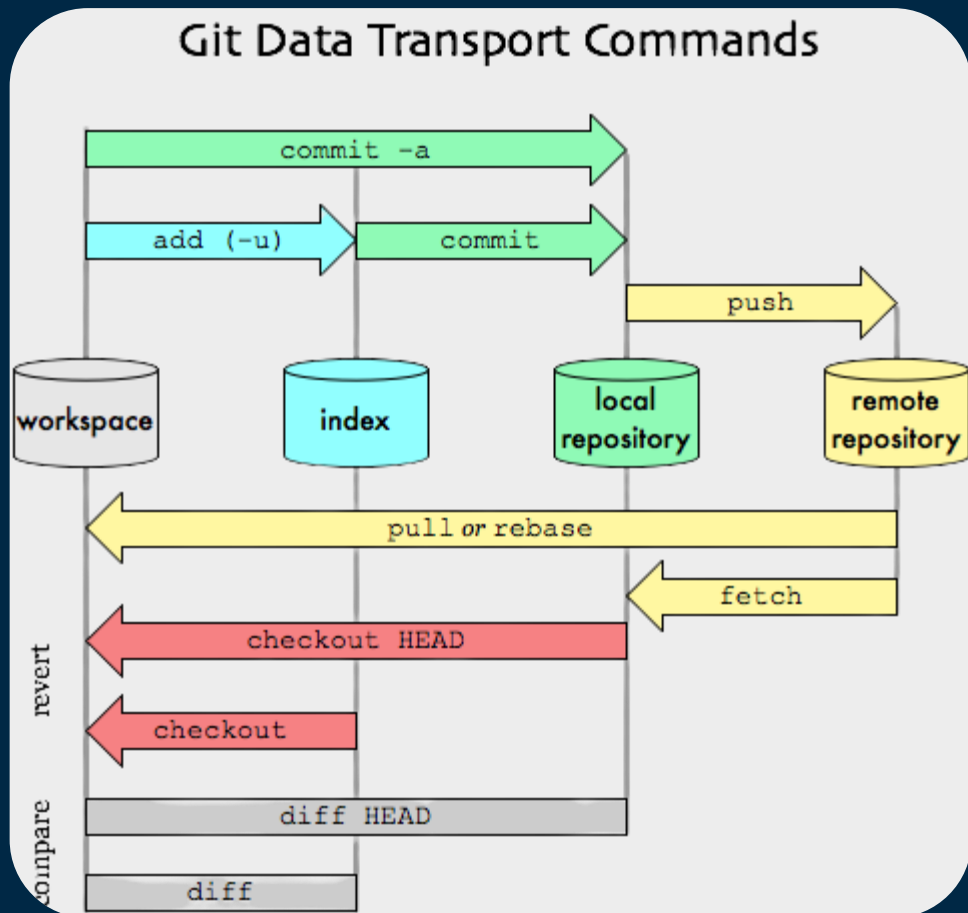
- Repository is the word used to describe your project & all the complementary files git adds.

2. Start your project.

3. After a while, stage your project.



How git
staging/saving
works:



HOW DOES IT WORK? (again 😊)

1. First, you start by initializing the repository.
(Initial Step)
2. Start your project.
3. Commit the staged file(s) to the local repository. **(Whenever)**
4. Push your commit to the remote repository.
(Whenever)

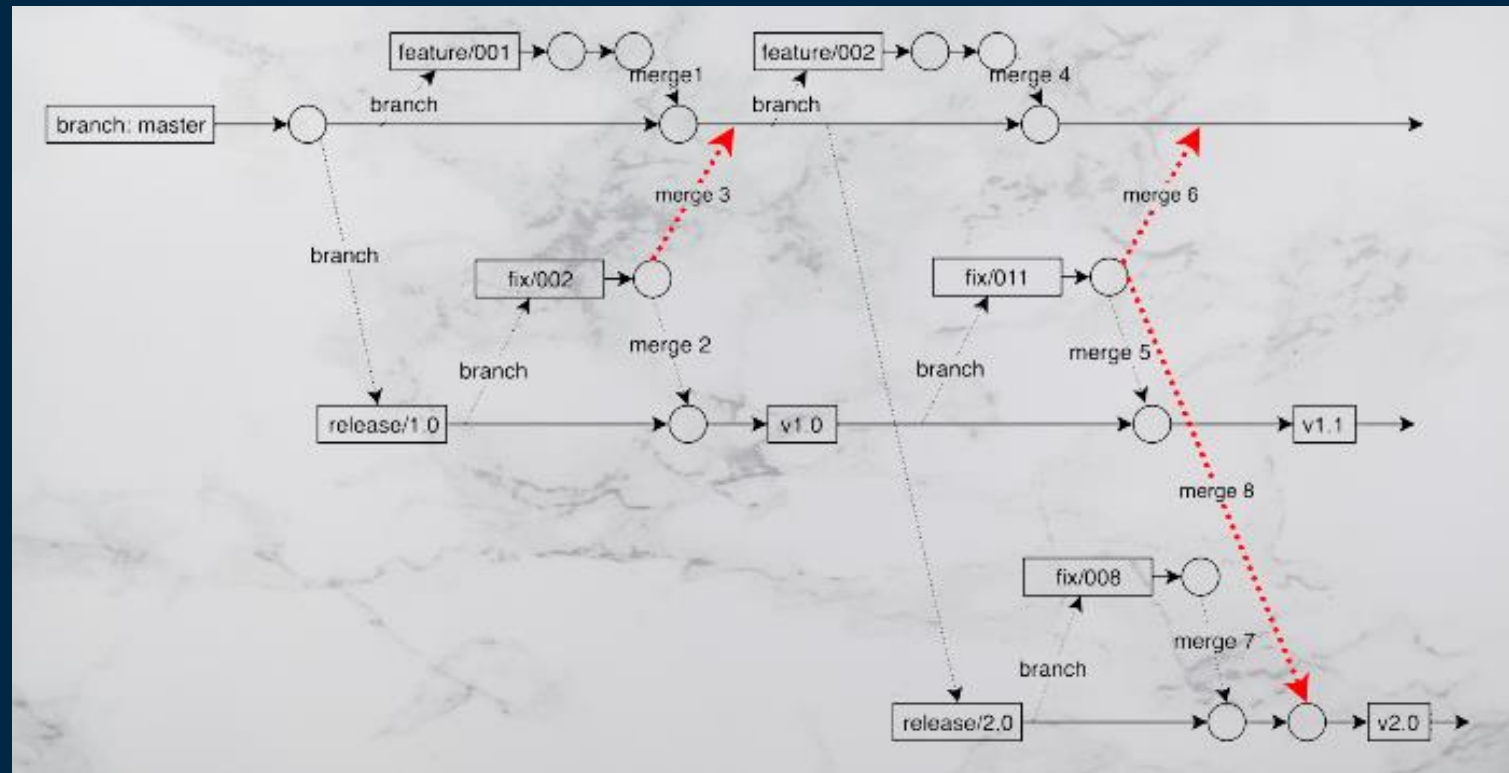
100



- 7



BRANCH



Top **Source Code** Repository **Hosts**



GitHub



WHAT IS GITHUB?

- GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.
- It supports remote repositories, which are either public or private, and for those, it supports wikis, custom webpages, issue tracking, etc.
- GitHub is the largest code hosting platform.

WHY I CHOUSE GITHUB

- IS THE BEST GIT PROVIDER
- UNLIMITED PUBLIC AND PRIVATE REPOS
- UNLIMITED COLLABORATORS
- OPEN SOURCE
- GITHUB ACTIONS
- SECURITY OF THE CODE
- INSIGHTS
- MARKETPLACE
- CODESPACE
- STUDENT DEVELOPER PACK
- COPILOT

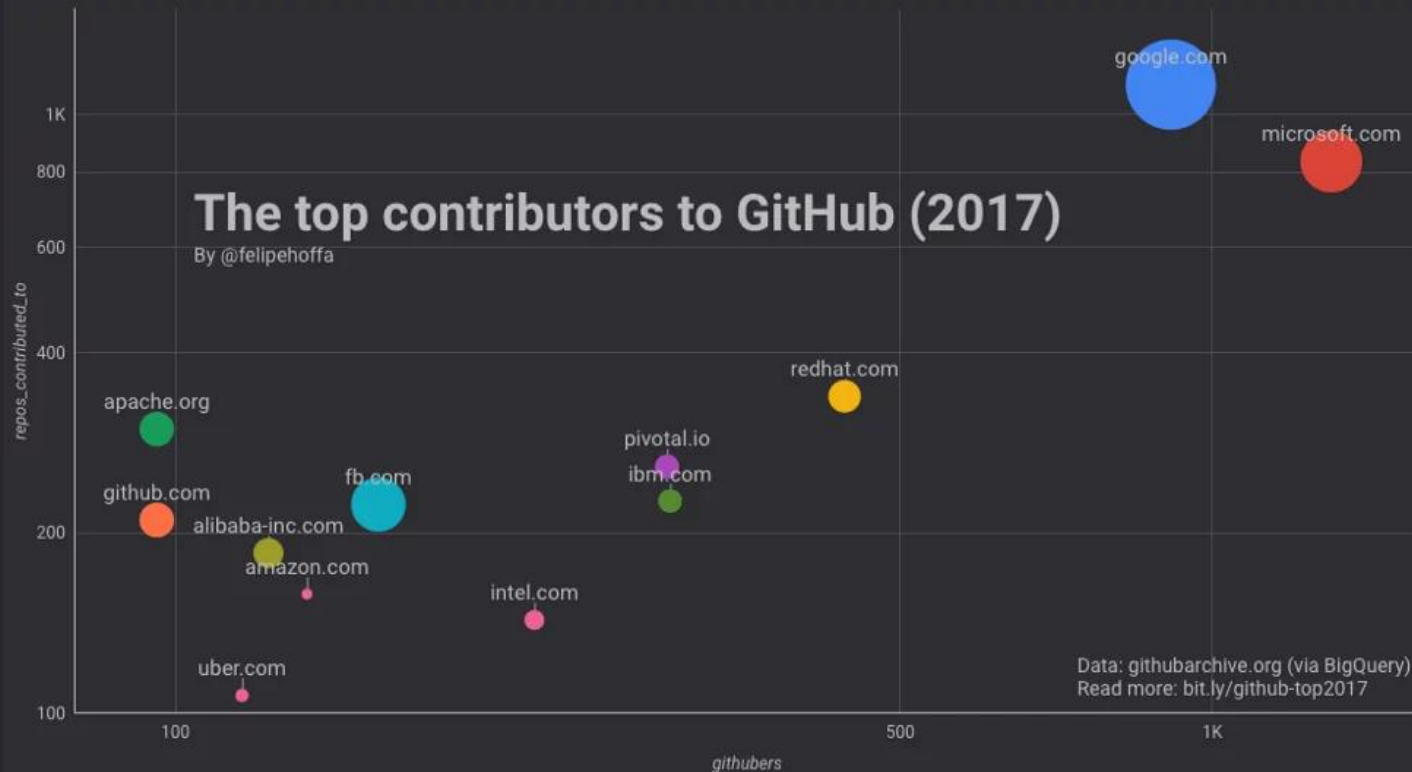
100 MILLIONS

Developers on GitHub



The top contributors to GitHub (2017)

By @felipehoffa

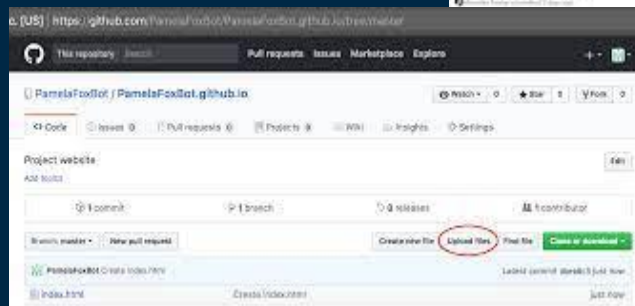
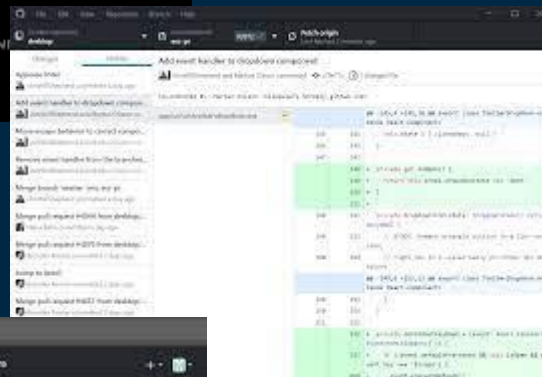


The Workshop

METHODS FOR GITHUB

- Git
- Web application
- GitHub Desktop
- Extensions from IDEs
- GitHub command line

```
$ git init
Initialized empty Git repository in /tmp/tmp.IMBYSY7R8Y/.git/
$ cat > README << 'EOF'
> Git is a distributed revision control system.
> EOF
$ git add README
$ git commit
[master (root-commit) e4dcc69] You can edit locally and push
to any remote.
1 file changed, 1 insertion(+)
create mode 100644 README
$ git remote add origin git@github.com:cdow
$ git push -u origin master
```



Install Git

Linux

Different by Linux distribution
<https://git-scm.com/download/linux>

Mac OS

<https://git-scm.com/download/mac>

Windows

<https://git-scm.com/download/win>

For all!



Step 0: Configuration

Linux

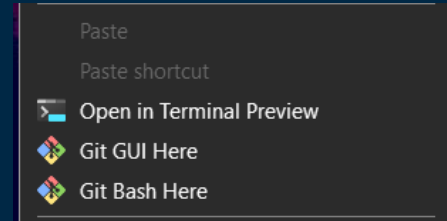
Open a terminal

Mac Os

Open a terminal

Windows

Right-click in an empty space.



Click on the option:

Git Bash Here

Step 0: Configuration

Write the following command on the terminal:

- `git config --global user.name "Your name"`

The name must be written the same way as it is on GitHub.

- `git config --global user.email "Your email"`

The email must be written the same way as it is on GitHub.

Step 1: Initialization

- Create a folder with the name:
MyFirstGitRepo

Step 1: Initialization

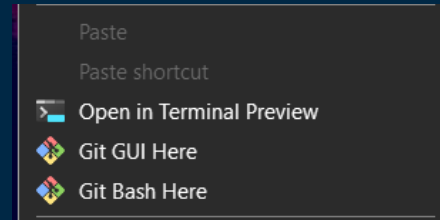
Linux / Mac /
Win. bash

Open a terminal, go to the new folder and type the following command:

```
git init
```

Windows GUI

Right-click in an empty space.



Click on the option:

Git GUI Here

Step 1: Initialization

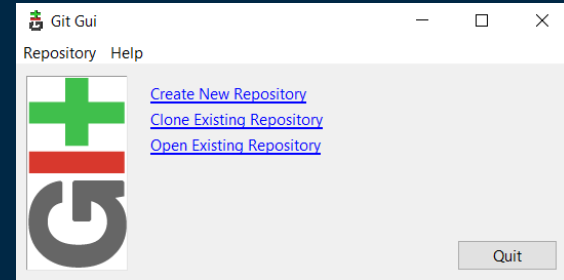
Linux / Mac /
Win. bash

Open a terminal, go to the new folder and type the following command:

```
git init
```

Windows GUI

Click on the option:
Create New Repository



Step 1: Initialization

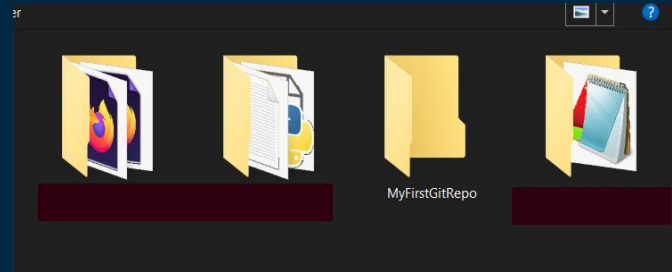
Linux / Mac /
Win. bash

Open a terminal, go to the new
folder and type the following
command:

```
git init
```

Windows GUI

**Use the navigator to find
the folder**



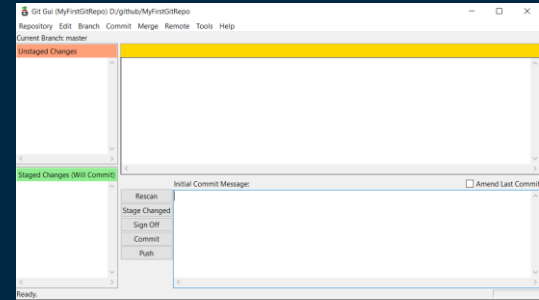
Step 1: Initialization

Linux / Mac /
Win. bash

Open a terminal, go to the new
folder and type the following
command:

```
git init
```

Windows GUI



Step 2: Your Project

Create a text file with the name:
`git_test.txt`

Inside write the following sentence:
Hi, my name is <your_name>!

Step 3: Staging

Linux / Mac /
Win. bash

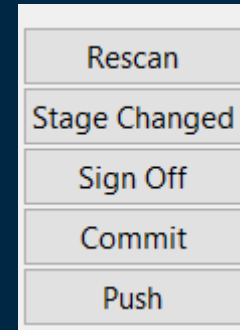
Type the following command on
the terminal:

```
git status
```

Windows GUI

On the Git GUI, click the button:

Rescan



Step 3: Staging

Linux / Mac /
Win. bash

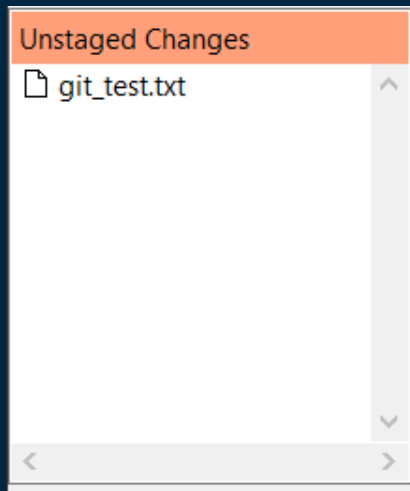
```
→ MyFirstGitRepo git:(master) X git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    git_test.txt

nothing added to commit but untracked files present (use "git add" to track)
→ MyFirstGitRepo git:(master) X
```

Windows GUI



Step 3: Staging

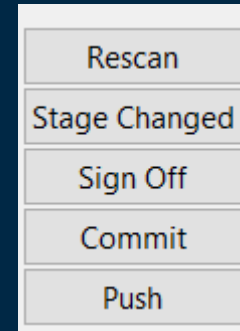
Linux / Mac /
Win. bash

Type the following command on
the terminal:

```
git add -A
```

Windows GUI

On the Git GUI, click the button:
Stage Changed



Step 3: Staging

Linux / Mac /
Win. bash

Type the following command on
the terminal:

git status

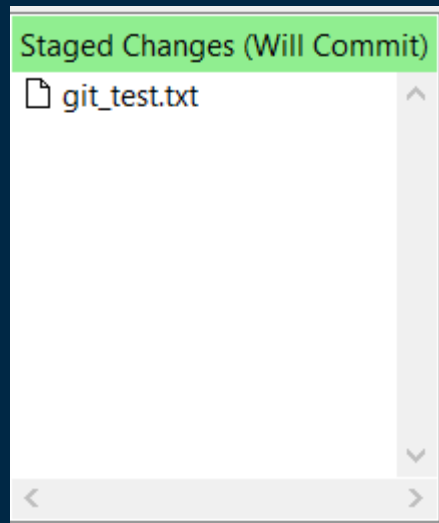
```
→ MyFirstGitRepo git:(master) X git add -A
→ MyFirstGitRepo git:(master) X git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   git_test.txt

→ MyFirstGitRepo git:(master) X |
```

Windows GUI



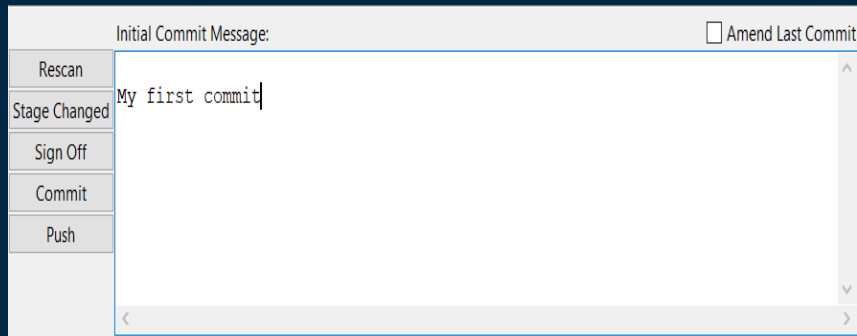
Step 4: Commit

Linux / Mac /
Win. bash

Type the following command on
the terminal:

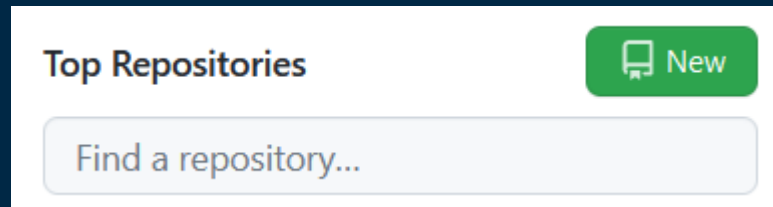
```
git commit -m "My first commit"
```

Windows GUI



Step 5: Remote Repo / GitHub

Go to github.com, log in, and
click on the new button.




Step 5: Remote Repo / GitHub

Name the repository:
MyFirstGitRepo
& Click Create repository

Repository template
Start your repository with a template repository's contents.


No template ▾


Owner * **Repository name ***

 ptoloudis ▾ / MyFirstGitRepo ✓

Great repository names a MyFirstGitRepo is available. need inspiration? How about **fuzzy-rotary-phone**?

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

`https://github.com/ptoloudis/MyFirstGitRepo.git`



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# MyFirstGitRepo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ptoloudis/MyFirstGitRepo.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/ptoloudis/MyFirstGitRepo.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Windows Only User

Because this step is more complex for the purposes of this Workshop, the Windows user will have to follow the next steps on the terminal.

Step 5: Remote Repo / GitHub

Write the following command on the terminal:

```
git remote add origin  
https://github.com/<your_user_name>/MyFirstGitRepo.git
```

Step 5: Remote Repo / GitHub

Write the following command on the terminal:

```
git push origin master
```



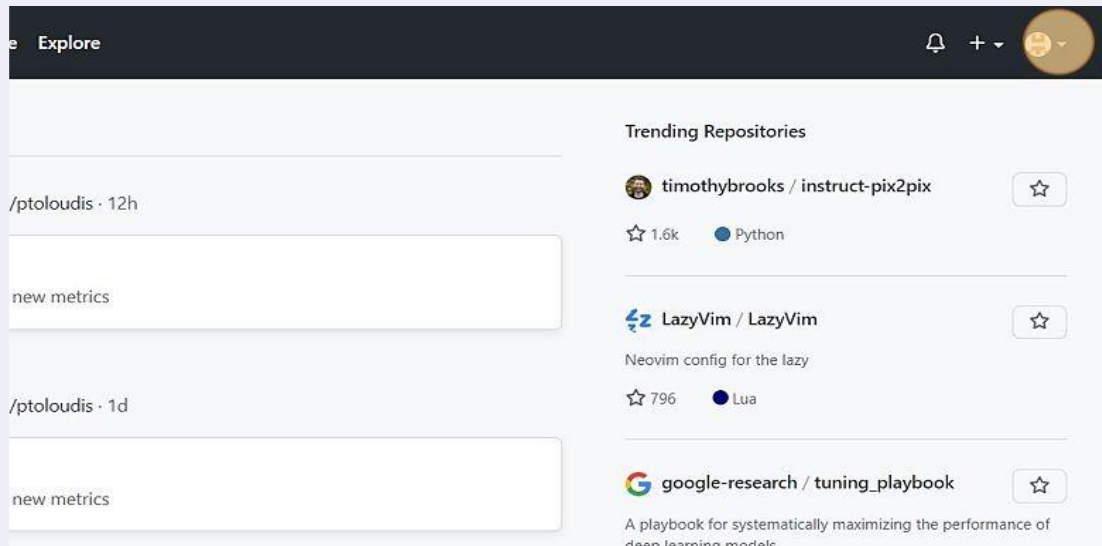
Your credentials for GitHub will be required.

Step 5: Remote Repo / GitHub

If the password does not work:
use the Personal access tokens

2

Click here.



3 Click "Settings"

new metrics

/ptoloudis · 1d

new metrics

/ptoloudis · 2d

new metrics

/ptoloudis · 3d

 LazyVim / LazyVim

Neovim config for the lazy

☆ 796

• Lua

 google-research / tu

A playbook for systematically
deep learning models.

☆ 8.8k

See more

Your projects

Your stars

Your gists

Your sponsors

Try Enterprise

Feature preview

Help

Settings

Sign out

5 Click "Personal access tokens"



Search or jump to...



Pull requests

Issues

Codespaces


Marketplace

Explore

Settings / Developer settings

 GitHub Apps

 OAuth Apps

 Personal access tokens



GitHub Apps

Want to build something that integrates with and extends GitHub the GitHub API. You can also read more about building GitHub



© 2023 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)


[Docs](#)

[Contact GitHub](#)

[Pi](#)


4 Click "Developer settings"

Security

 Code security and analysis


Integrations

 Applications

 Scheduled reminders

Archives

 Security log

 Sponsorship log

<> Developer settings

☐ Display current local time

Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, as out, you're giving us consent to share this data wherever your user profile, our privacy statement to learn more about how we use this information.

[Update profile](#)

Contributions & Activity

☐ Make profile private and hide activity

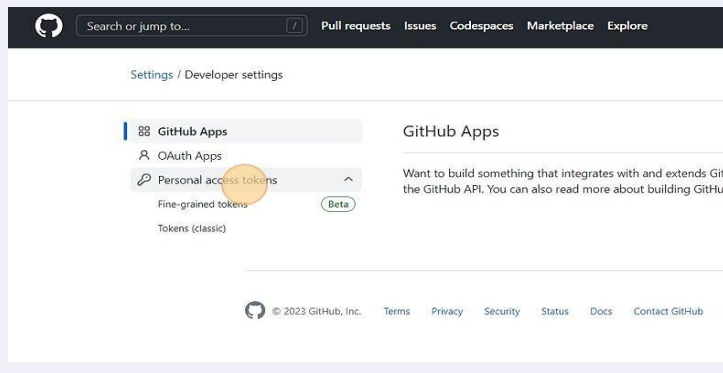
Enabling this will hide your contributions and activity from your GitHub leaderboards and releases.

☐ Include private contributions on my profile

Your contribution graph, achievements, and activity overview will show organization information. [Read more.](#)

6

Click "Personal access tokens"

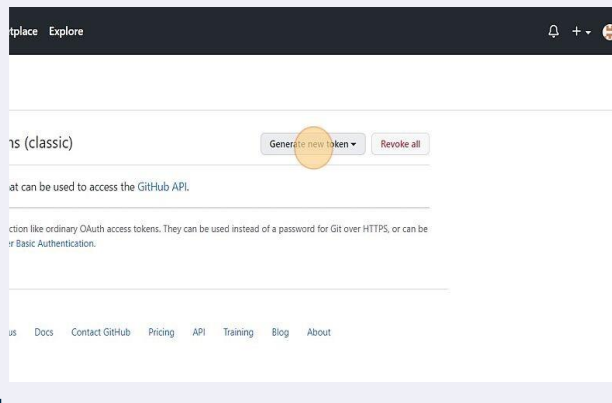


7

Click "Tokens (classic)"

8

Click "Generate new token"



10 Click the "Note" field.

ngs

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

What's this token for?

Expiration *

30 days The token will expire on Wed, Feb 22 2023

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

<input type="checkbox"/> repo	Full control of private repositories
-------------------------------	--------------------------------------

11 Type "workshop" and click the repo.

12 Click "Generate token"

<input type="checkbox"/> project	Full control of projects
<input type="checkbox"/> read:project	Read access of projects
<input type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys


Generate token Cancel

13 Copy the code like this

```
ghp_VrgMblOAXnYKo68RLKoBa8dnuPXc2B2D43zE
```



```
→ MyFirstGitRepo git:(master) git push origin master
Username for 'https://github.com': ptoloudis
Password for 'https://ptoloudis@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 242 bytes | 1024 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ptoloudis/MyFirstGitRepo.git
 * [new branch]      master -> master
→ MyFirstGitRepo git:(master) git status
On branch master
nothing to commit, working tree clean
→ MyFirstGitRepo git:(master) |
```

 master ▾

 1 branch

 0 tags

[Go to file](#)

[Add file ▾](#)

[Code ▾](#)



ptoloudis My first commit

a383a23 53 minutes ago  1 commit



git_test.txt

My first commit

53 minutes ago

Help people interested in this repository understand your project by adding a README.

[Add a README](#)



THANK YOU FOR YOUR TIME!

CREDITS: This presentation template was created by [Slidesgo](#),
including icons by [Flaticon](#), and infographics & images by [Freepik](#)