Basic of git

Introduction to git and collaborative working

Nikhil Nath R

What we will do

- Key concepts
- Usage: Simple
- Branching
- Connecting to remote
- Contribute to project

Key concepts

Git is a distributed version control system that tracks versions of files.

What is version control

- System which keeps a track of your changes
- Allows for collaborative development
- Allows you to know who made what change and when
- Allow you to go back to any previous state

snapshots

- Save point of the code you are writing
- Records how your files look at a given point of time
- You decide when to take a snapshot

commit

- The act of creating a snapshot
- A project is made up of a bunch of commits
- Has 3 information
  1. How files changed from previous commit
  2. A hash code, uniquely identify the commit
  3. A reference to the previous commit

staging area

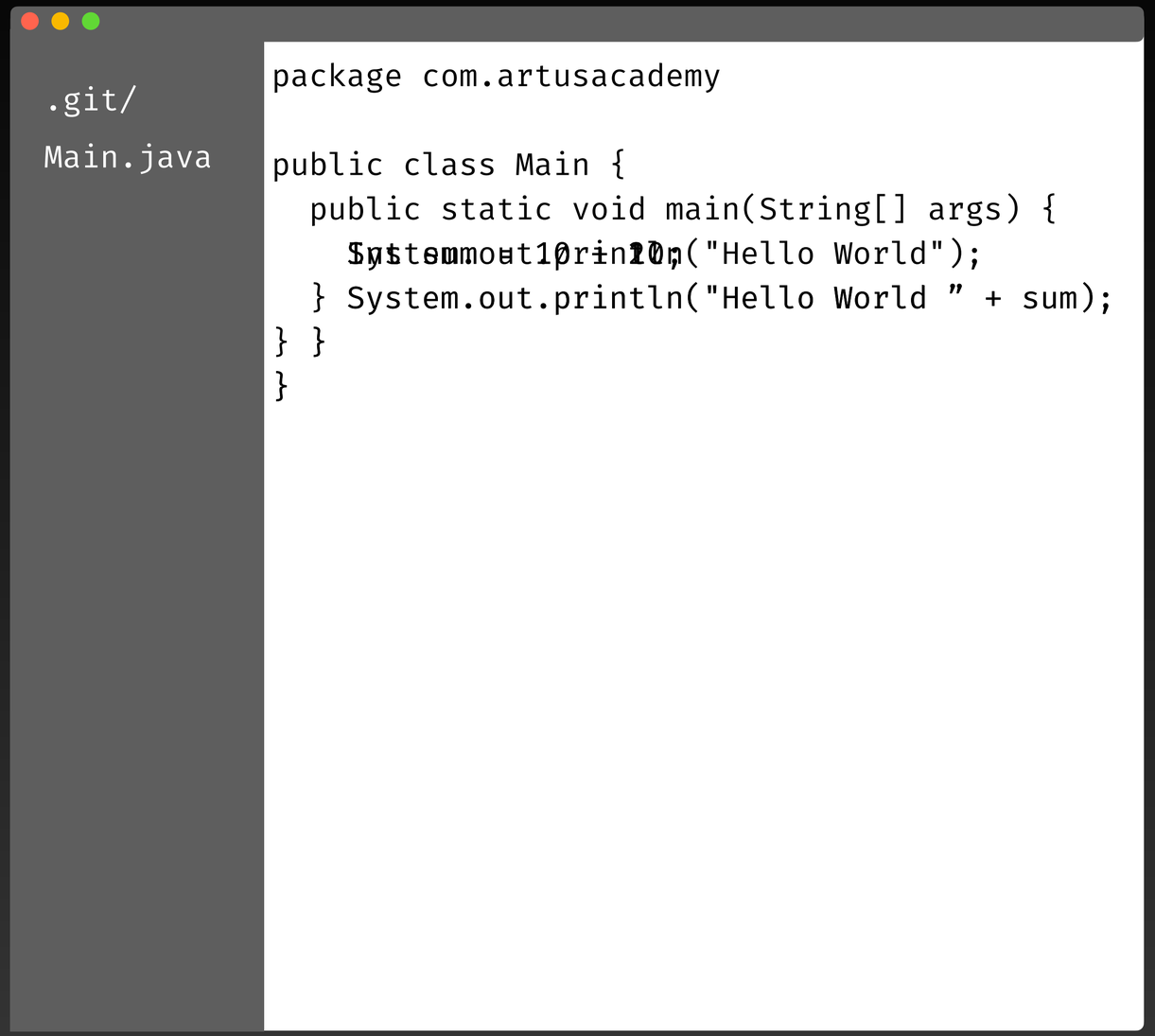
- Called the **staging area** or **index**
- An intermediate area where commits can be formatted and reviewed before completing the commit

repository

- A collection of all files and all the history of the project
- Can be stored in local machine or a **remote** (GitHub, GitLab etc.)
- Often shortened to **repo**
- The act of copying from remote to machine is called **cloning**

HEAD

- HEAD is a reference to the current check-out commit in your repository
- By default it a pointer or symbolic reference to the latest commit in your branch


.git/
Main.java
package com.artusacademy
public class Main {
public static void main(String[] args) {
System.out.println("Hello World");
} }
}

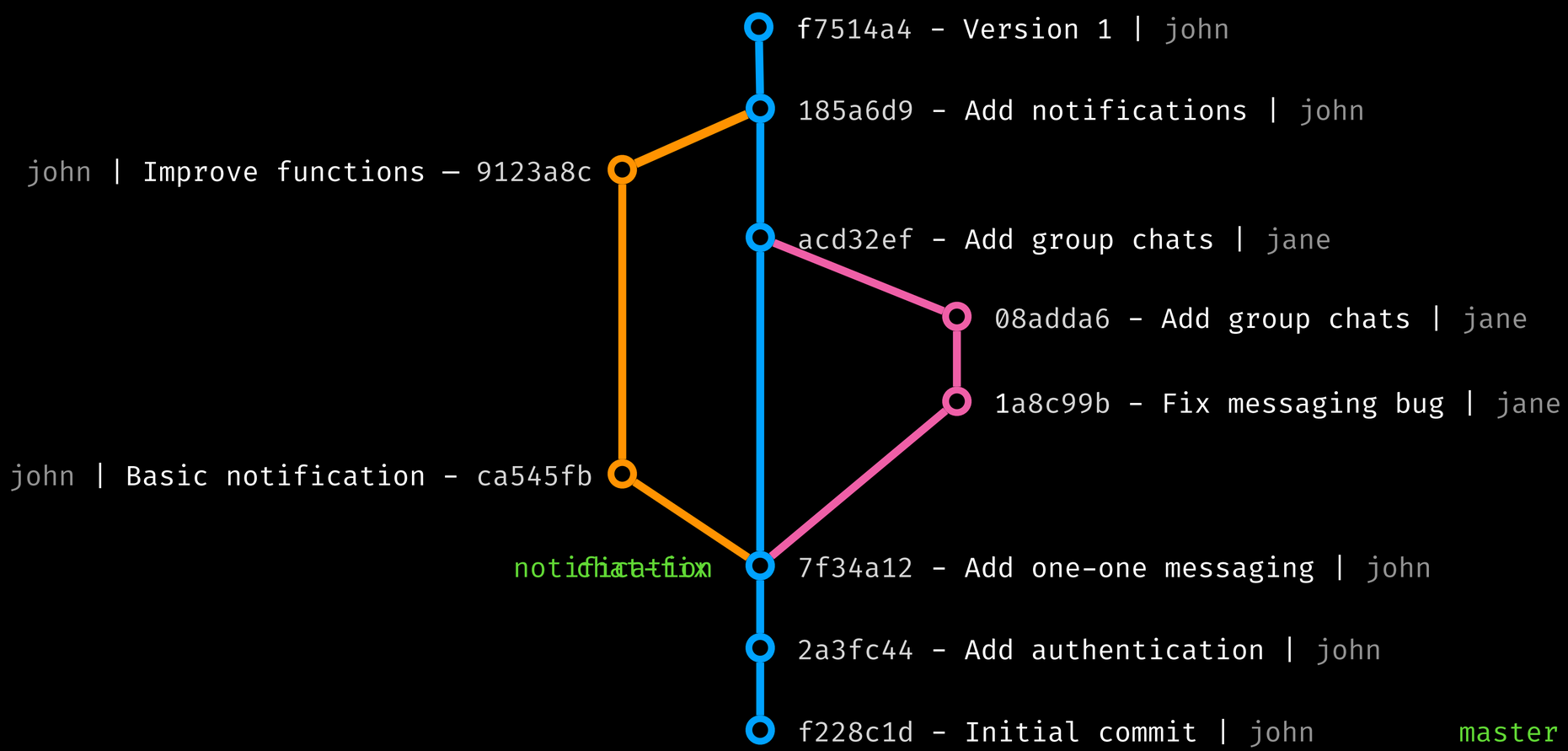

git repository
7f34a12 - Change the sum
2a3fc44 - Add sum function
f228c1d - Hello World [HEAD] master

branches

- A bunch of commits linked together
- Branches can be worked from independently from each other
- The main branch is called **master** branch

merge

- The process of combining to separate branches in to a single one
- Allows to combine the code or text from 2 separate features or developers


f7514a4 - Version 1 | john
185a6d9 - Add notifications | john
john | Improve functions – 9123a8c
acd32ef - Add group chats | jane
08adda6 - Add group chats | jane
1a8c99b - Fix messaging bug | jane
john | Basic notification - ca545fb
notification-fix
7f34a12 - Add one-one messaging | john
2a3fc44 - Add authentication | john
f228c1d - Initial commit | john
master

630M

Total projects

1.12B

Total contribution in the year

Github Octaverse insights 2025

Usage: Simple

Commands

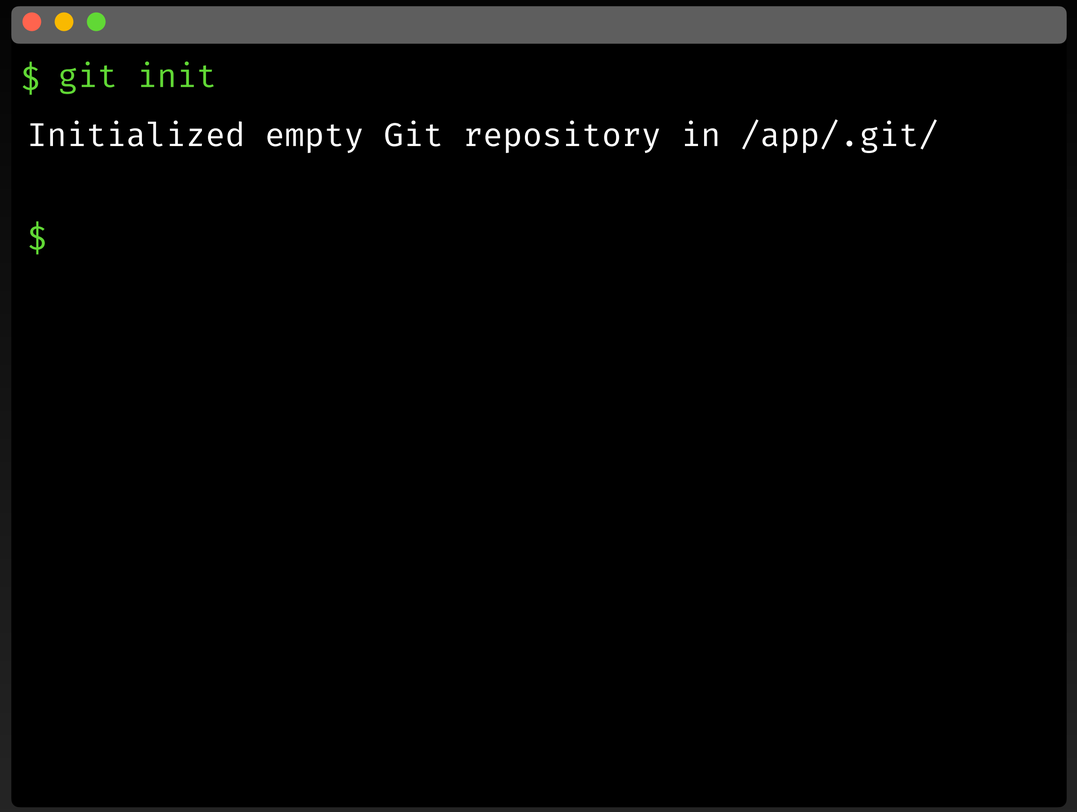
- `git init`
- `git add`
- `git commit`
- `git status`
- `git diff`
- `git log`


$ git config --global user.name “John Doe”
$ git config --global user.email “john@example.com"
$

git init

https://git-scm.com/docs/git-init

- A one-time command you use during the initial setup of a new **repository**.
- This will create a new master (main) **branch**.


$ git init
Initialized empty Git repository in /app/.git/
$

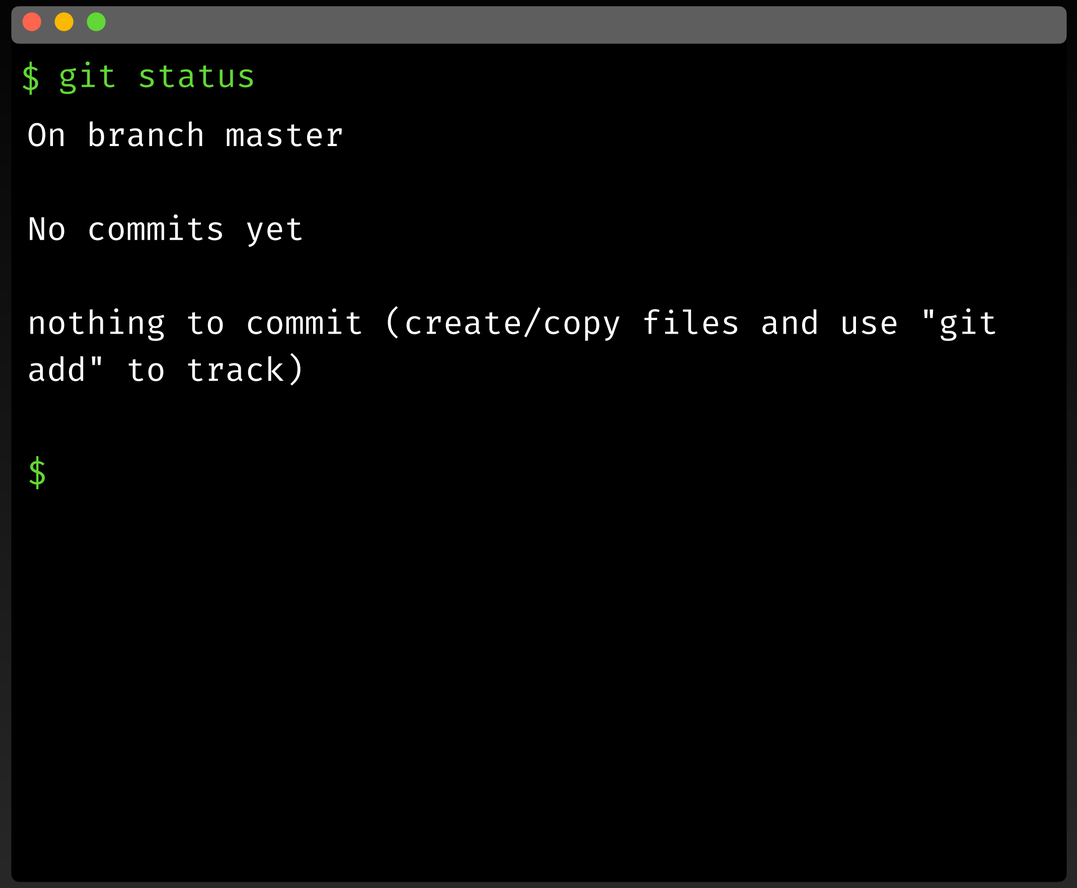
created a .git folder

- This is the "thing" which makes your project a Git repository
- **.git** contains all information required for version control.
- If you want to clone your repository, copying **.git** is enough.

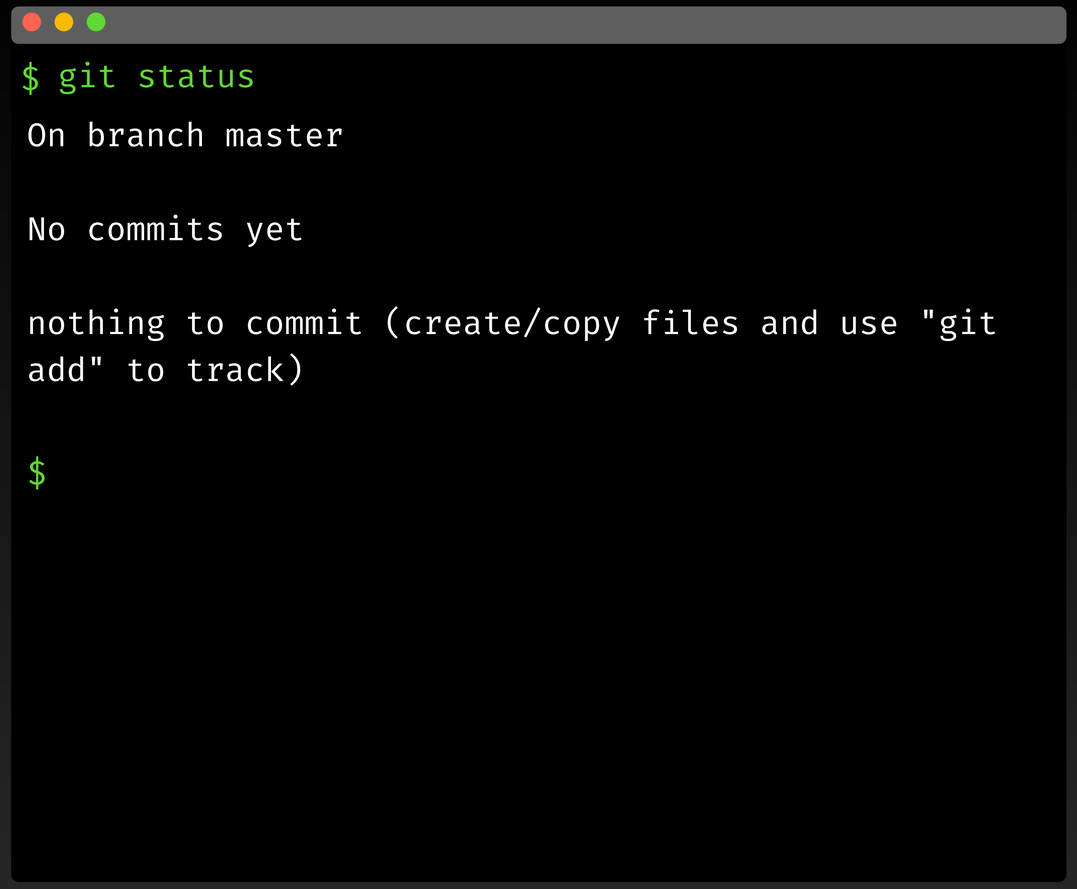
git status

https://git-scm.com/docs/git-status

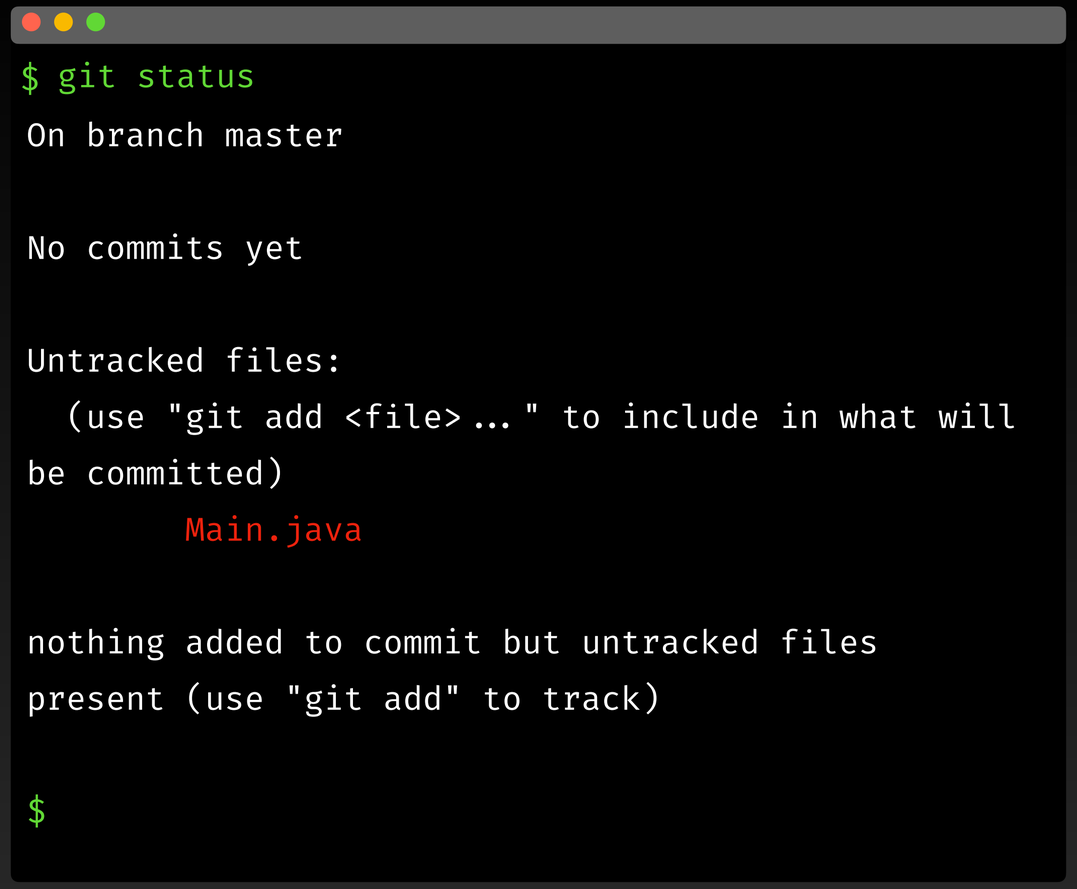
- Show the working tree status
- Gives information about the files which are not added to the repository.


$ git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
$


.git/
Main.java
package com.artusacademy
public class Main {
public static void main(String[] args) {
System.out.println("Hello World");
}
}


$ git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
$


.git/
Main.java
package com.artusacademy
public class Main {
public static void main(String[] args) {
System.out.println("Hello World");
}
}

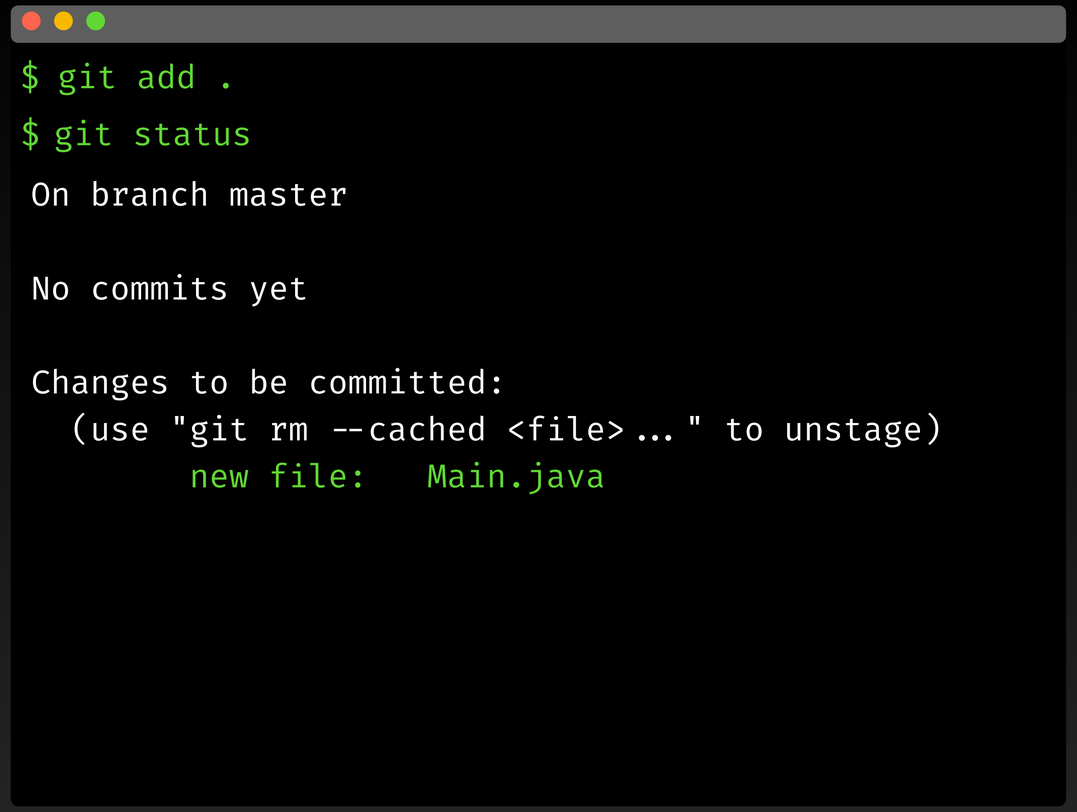

$ git status
On branch master
No commits yet
Untracked files:
(use "git add <file>..." to include in what will be committed)
Main.java
nothing added to commit but untracked files present (use "git add" to track)
$

git add

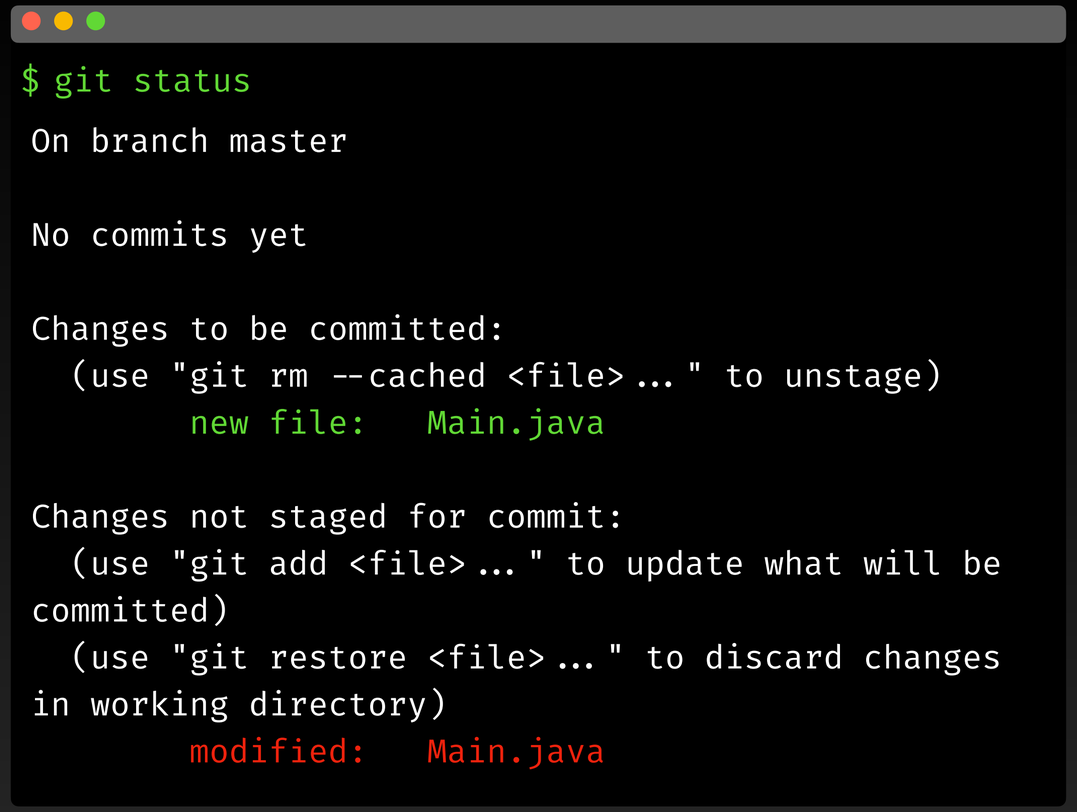
https://git-scm.com/docs/git-add

- This command updates the index using the current content found in the working tree, to prepare the content staged for the next commit.
- `git add .` - add all untracked changes to stage
- `git add <path>` - add all files in given path to stage


.git/
Main.java
package com.artusacademy
public class Main {
public static void main(String[] args) {
System.out.println("Hello World”);
}
}


$ git add .
$ git status
On branch master
No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: Main.java


.git/
Main.java
package com.artusacademy
public class Main {
public static void main(String[] args) {
System.out.println("Hello World!”);
}
}

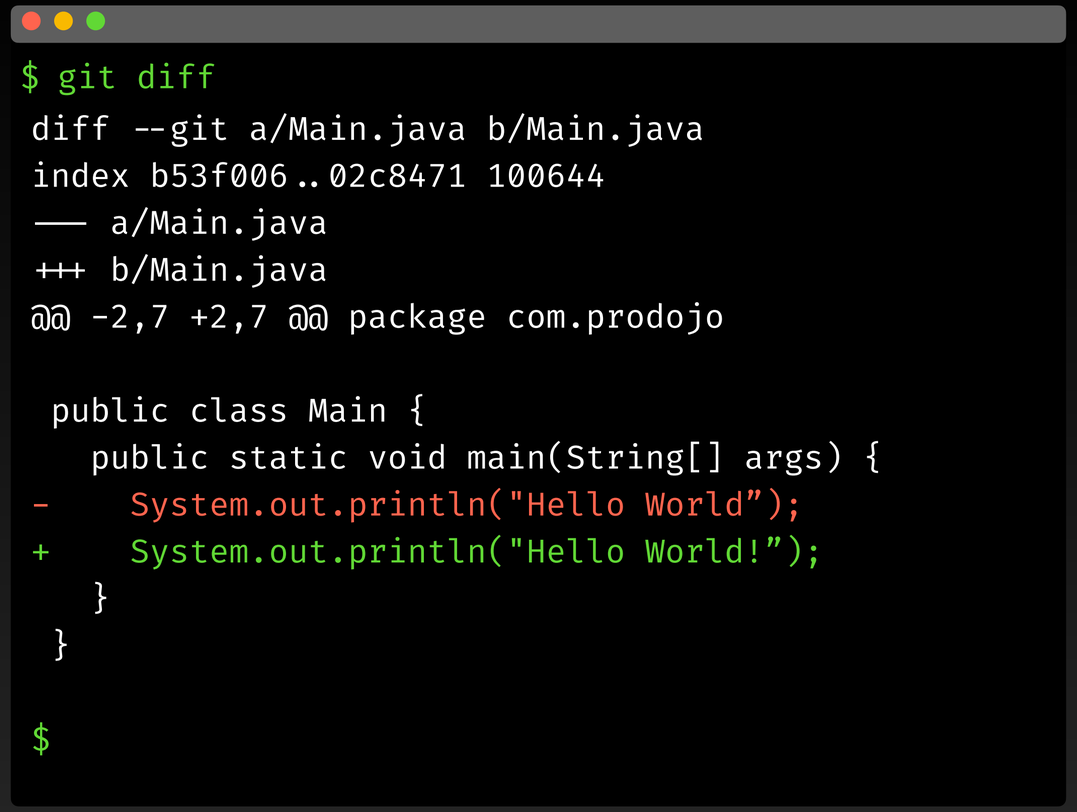

$ git status
On branch master
No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: Main.java
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: Main.java

git diff

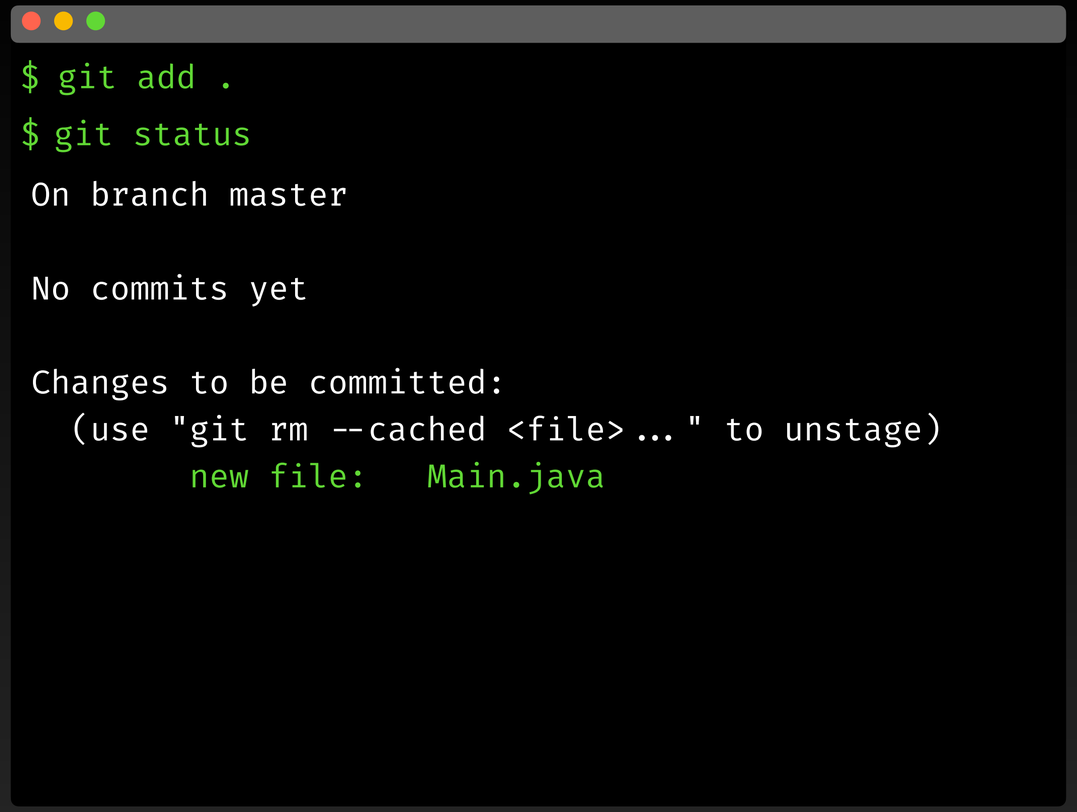
https://git-scm.com/docs/git-diff

- Show changes between the working tree and the index or a tree
- `git diff` - show changes in untracked file
- `git diff --staged` - show changes in staging area


.git/
Main.java
package com.artusacademy
public class Main {
public static void main(String[] args) {
System.out.println("Hello World!”);
}
}


$ git diff
diff --git a/Main.java b/Main.java
index b53f006..02c8471 100644
--- a/Main.java
+++ b/Main.java
@@ -2,7 +2,7 @@ package com.prodojo
public class Main {
public static void main(String[] args) {
- System.out.println("Hello World”);
+ System.out.println("Hello World!”);
}
}
$


.git/
Main.java
package org.prodojo
public class Main {
public static void main(String[] args) {
System.out.println("Hello World!”);
}
}

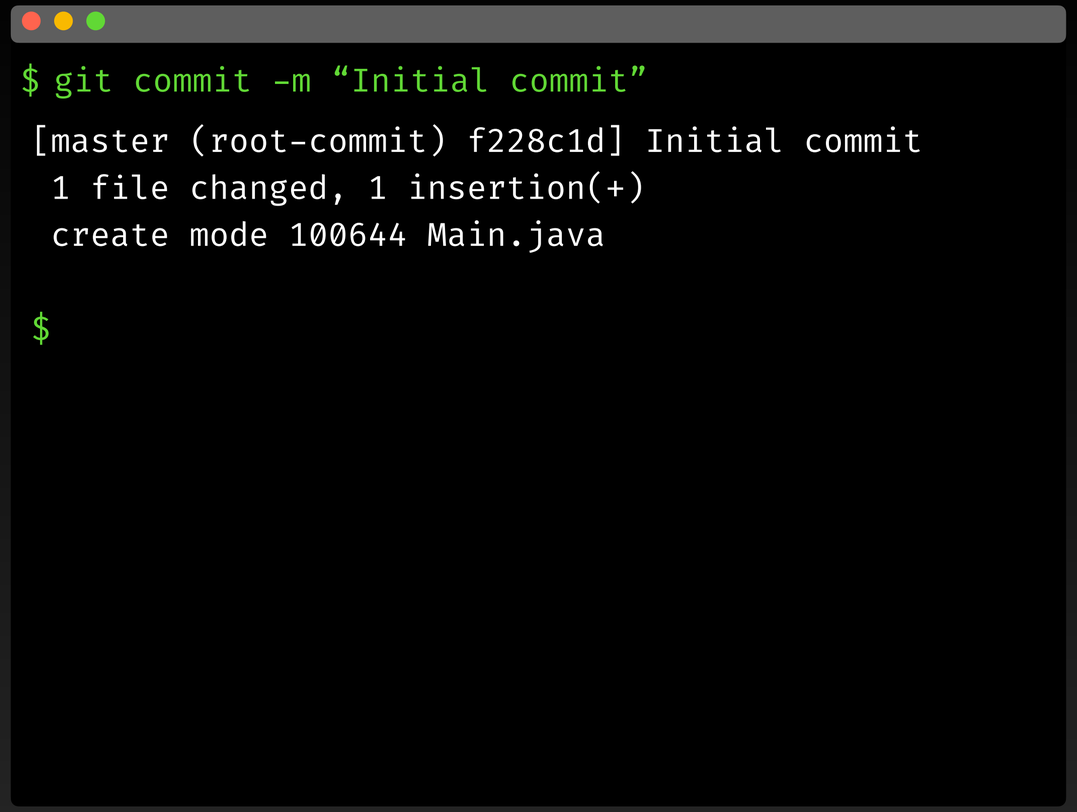

$ git add .
$ git status
On branch master
No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
new file: Main.java

git commit

https://git-scm.com/docs/git-commit

- Create a new commit containing the current contents of the index and the given log message describing the changes
- `git commit -m “<commit message>”`


.git/
Main.java
package com.artusacademy
public class Main {
public static void main(String[] args) {
System.out.println("Hello World!”);
}
}


$ git commit -m “Initial commit”
[master (root-commit) f228c1d] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 Main.java
$

Good commit message

- Use an imperative tone like, giving an order or request
- Do not end in punctuation
- Do not exceed 50 characters for first line, be precise
- More information can be added in description and lines should not exceed 72 characters

Good commit message: Examples

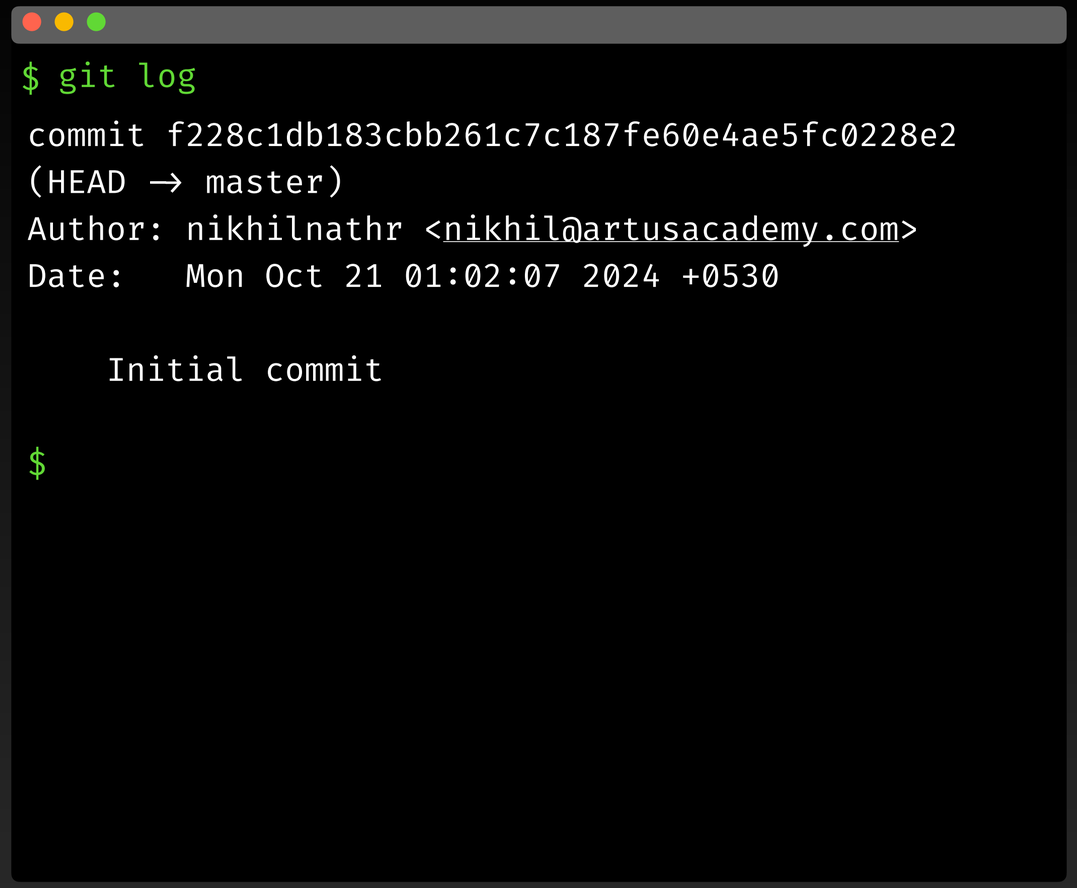
- Add fix for dark mode toggle state
- Update incorrect contact number in footer

- Changed style
- oops
- Fixed a bug which prevents users from logging into the application with a phone number

git log

https://git-scm.com/docs/git-log

- Show the history of commits existing in the repository.
- The output is given in reverse chronological order by default.


$ git log
commit f228c1db183cbb261c7c187fe60e4ae5fc0228e2
(HEAD → master)
Author: nikhilnathr <nikhil@artusacademy.com>
Date: Mon Oct 21 01:02:07 2024 +0530
Initial commit
$

Review

.git/

Main.java

```
package com.artusacademy

public class Main {
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

```
$ git init
Initialized empty Git repository in /app/.git/

$ git add .

$ git add .

$ git commit -m “Initial commit”
[master (root-commit) f228c1d] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 Main.java
```

untracked

“Hello World!”
Main.java

staged

git repository

f228c1d - Initial commit (HEAD → master)

“Hello World!”
Main.java

Create another commit

- Add a integer variable `sum` whose value is `10 + 10`.
- Display the variable `sum` after `Hello World!`.
- Commit the code with message `Print sum`.

.git/

Main.java

```
package com.artusacademy

public class Main {
  public static void main(String[] args) {
    int sum = 10 + 10;
    System.out.println("Hello World!");
  } System.out.println("Hello World!" + sum);
} }
}
```

```
$ git add .
$ git commit -m “Print sum”
[master 6c1735d] Print sum
 1 file changed, 2 insertions(+), 1 deletion(-)
```

untracked

“Hello World! 20”
Main.java

staged

git repository

6c1735d - Print sum

f228c1d - Initial commit (HEAD → master)

“Hello World! 20”
Main.java

“Hello World!”
Main.java

Branching

Commands

- `git branch`
- `git switch`
- `git merge`

git branch

https://git-scm.com/docs/git-branch

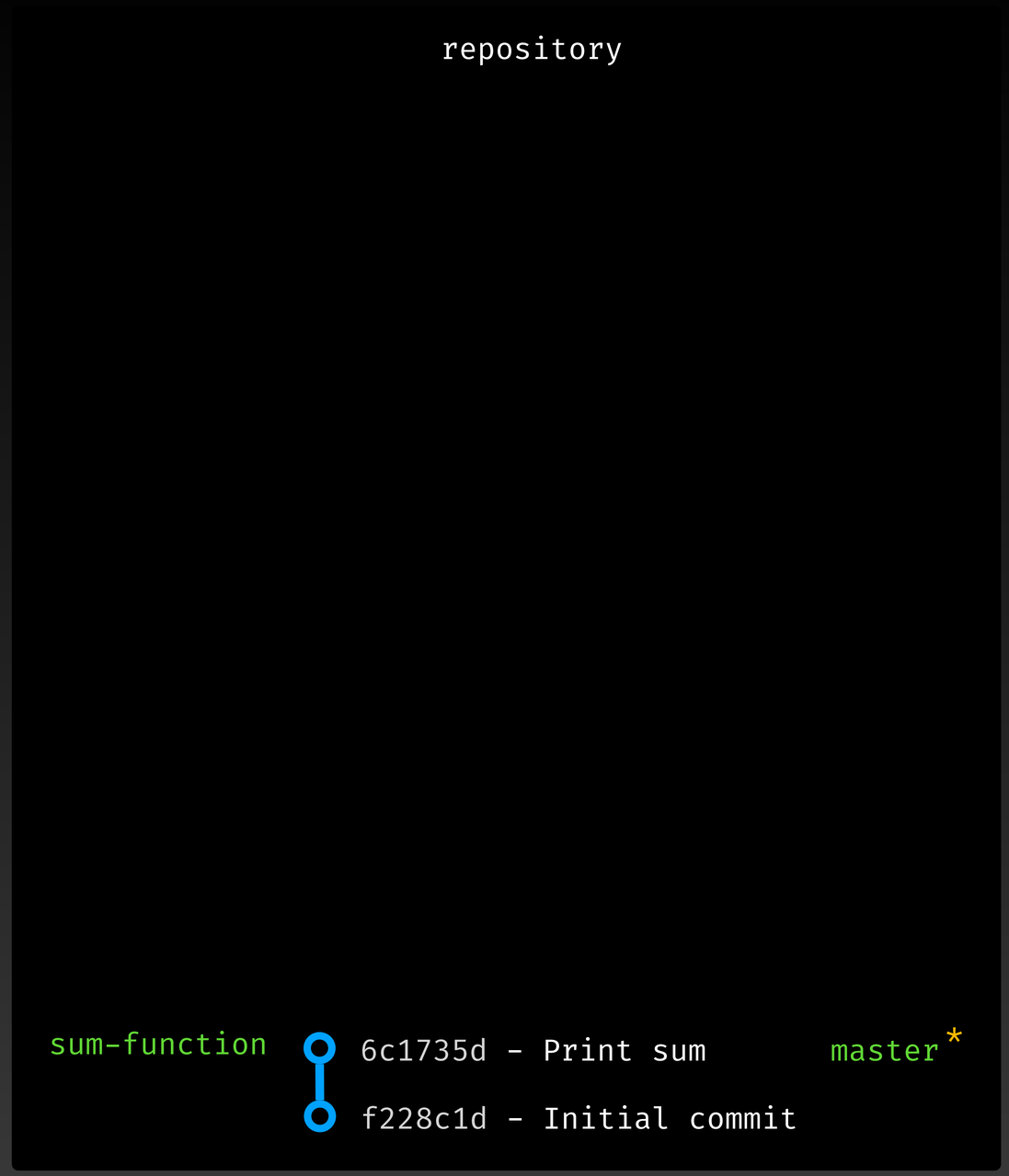
- Show the list of branches
- The current branch will be highlighted in green and marked with an asterisk.
- Show branches with,

```
git branch
```

- Create a branch with,

```
git branch <branch_name>
```

```
$ git branch
* master

$ git branch sum-function
$ git branch
  sum-function
* master

$
```



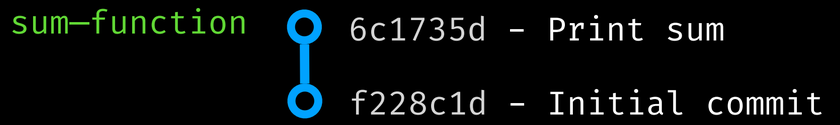
git switch

https://git-scm.com/docs/git-switch

- Switch to a specified branch.
- The working tree and the index are updated to match the branch.
- All new commits will be added to the tip of this branch.

```
$ git switch sum-function
Switched to branch ‘sum-function’

$ git branch
* sum-function
  master

$
```

repository



Commit in branch

- Add a function add in code for a function to calculate a sum of two numbers
- Use the function to calculate the sum.
- Commit using the message “Add sum function”

Main.java

```
public class Main {
  public static void main(String[] args) {
    int sum = add(10, 10);
    System.out.println("Hello World");
  }

  private int add(int a, int b) {
    return a + b;
  }
}
```

Do another commit

- Switch to master branch
- Create a file called README.md
- Commit with message “Add README file”

README.md

```
# Hello World

This is a hello world program written by nikhilnathr.
```

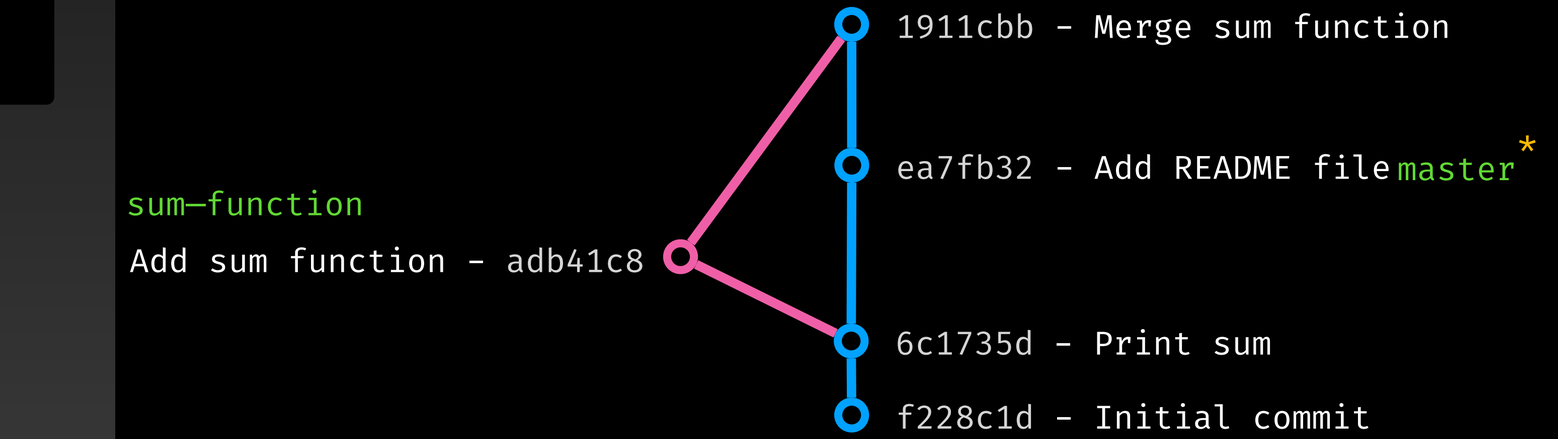
git merge

https://git-scm.com/docs/git-merge

- Join two or more development branches together
- To consolidate development work done in different branches.
- To do a merge
  - `git switch master`
  - `git merge <branch-name>`
  - (Or with message) `git merge <branch> -m “message”`

```
$ git merge sum-function -m “Merge sum function”
Merge made by the 'ort' strategy.
 hello.c | 6 +++++-
 1 file changed, 5 insertions(+), 1 deletion(-)

$
```

repository



Connecting to remote

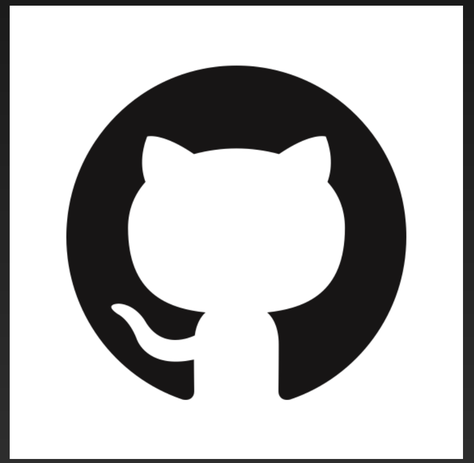
Commands

- `git remote`
- `git push`
- `git pull`

Github

https://www.github.com

- Largest web-based git hosting service
- Adds a UI for functionalities of git
- Founded in 2008

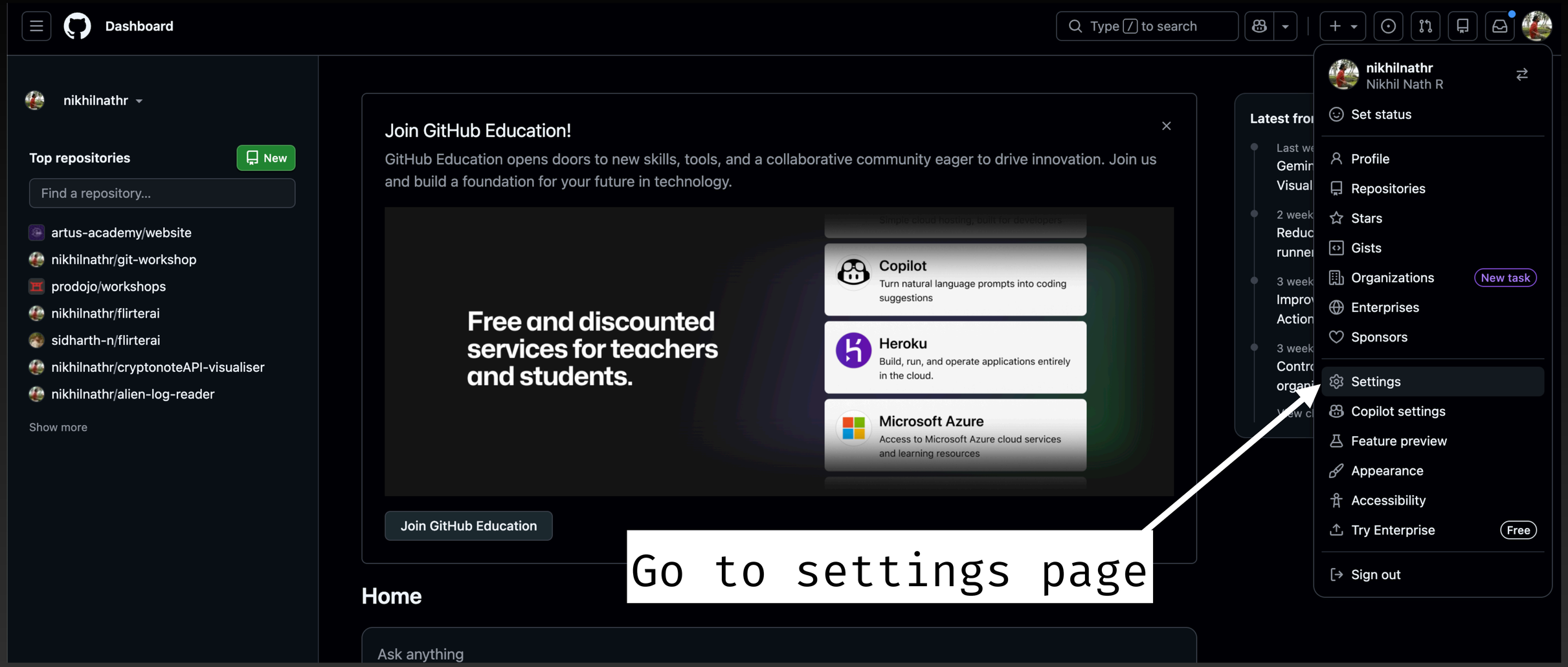


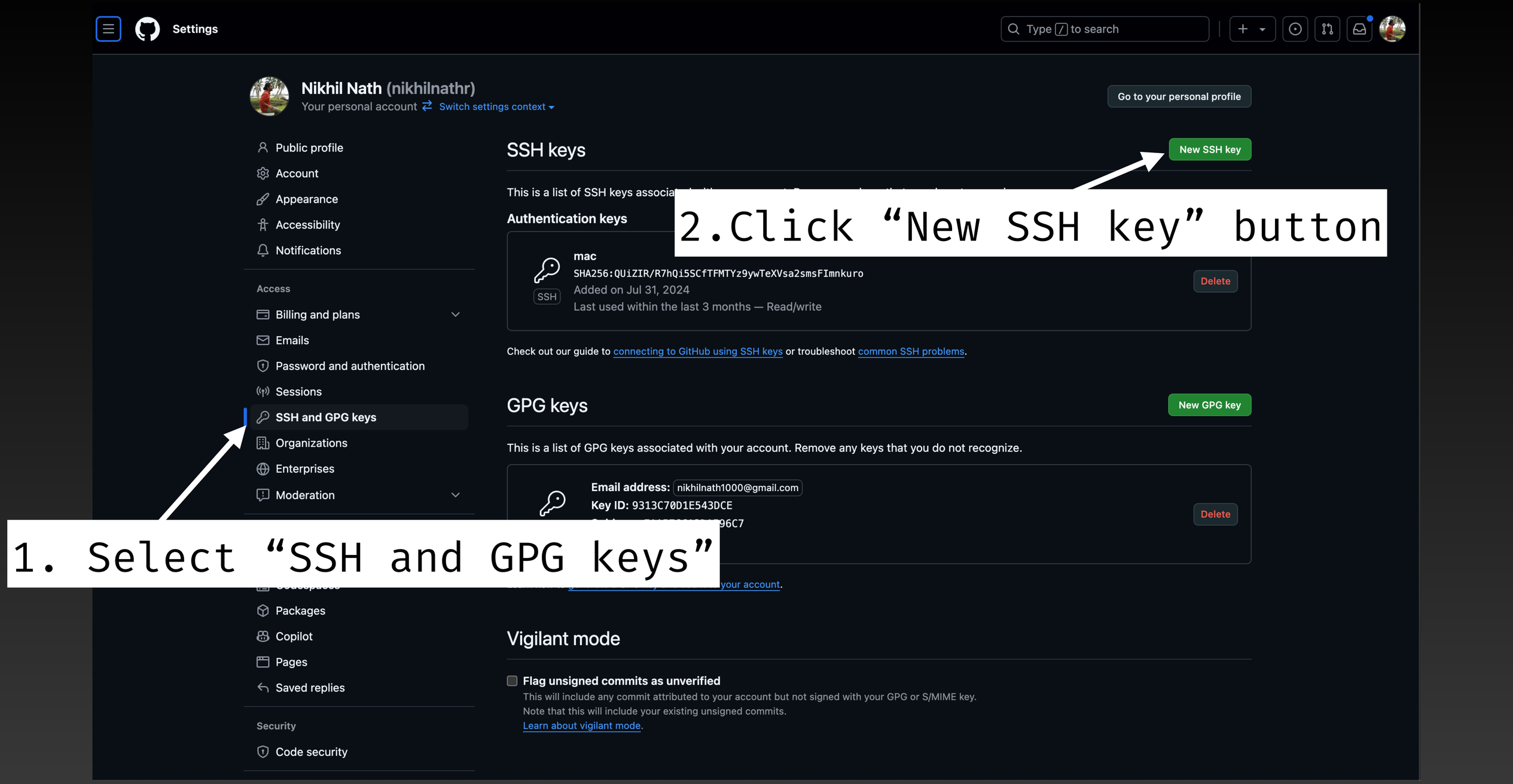
Create a GitHub account

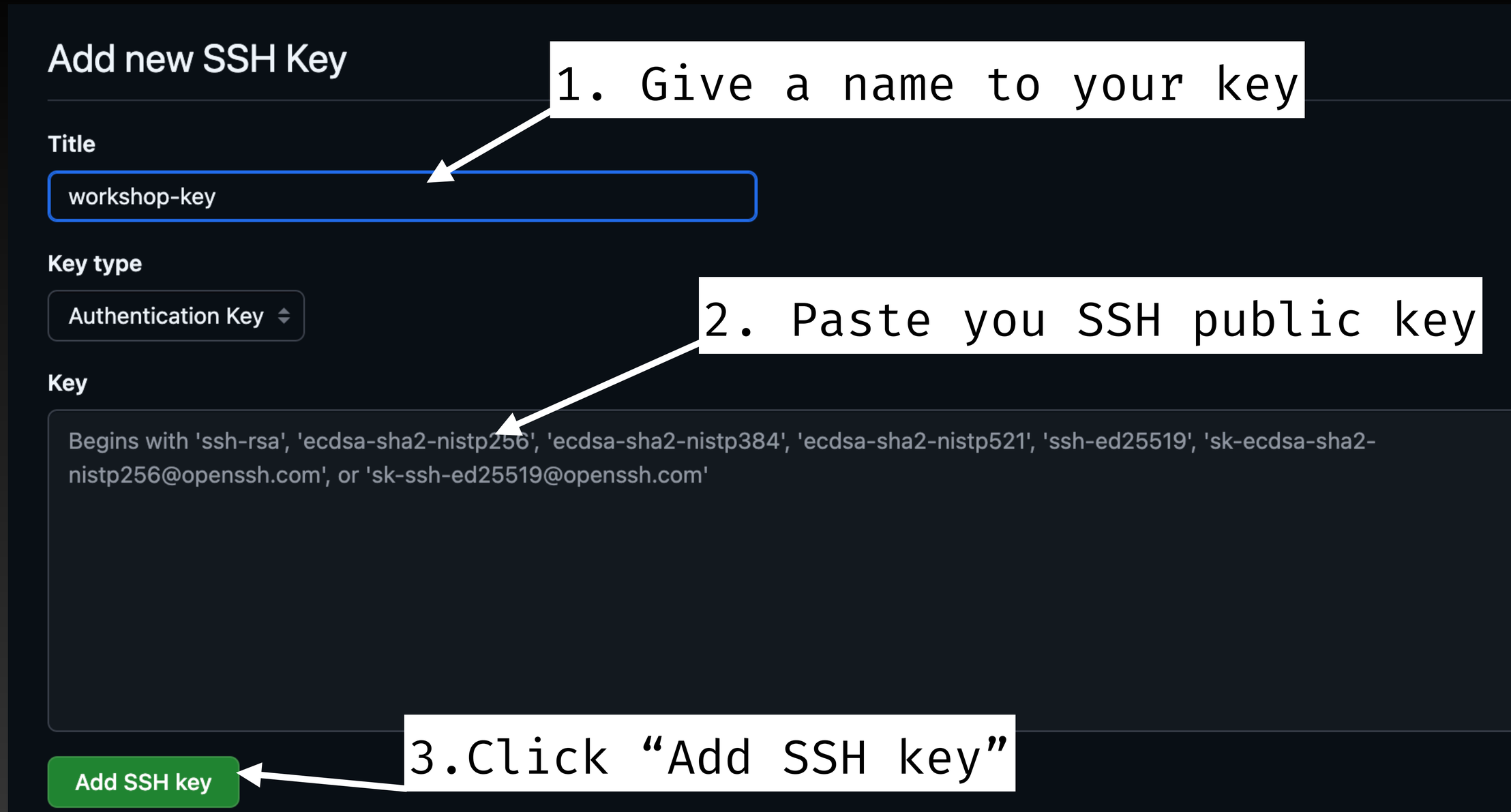
(or login)

github.com

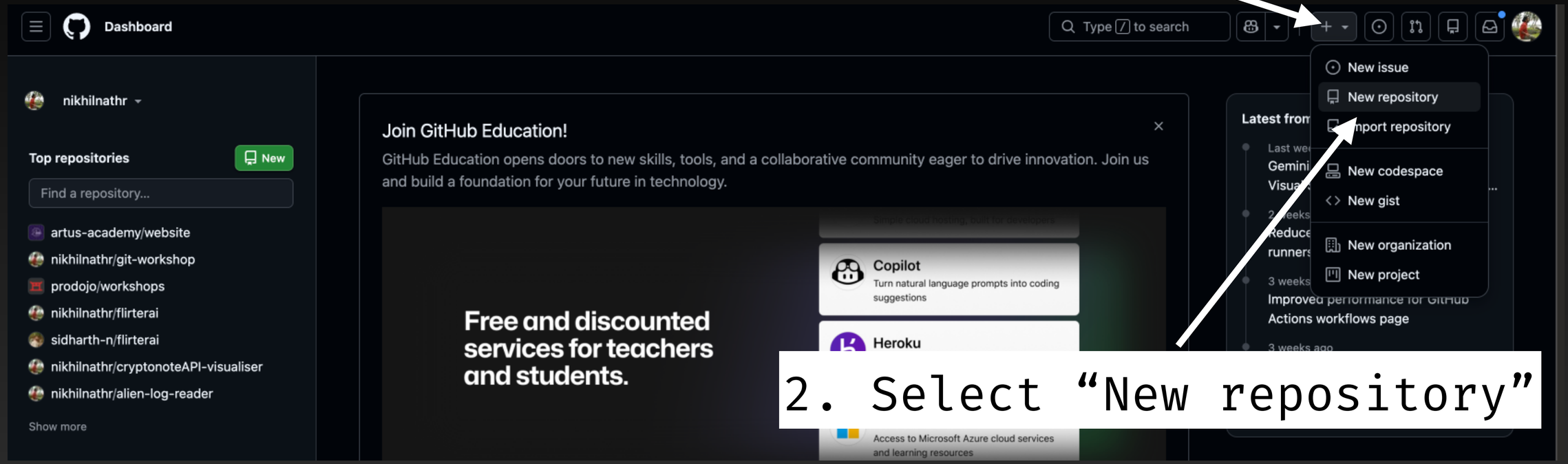
```
$ ssh-keygen -t ed25519 -f ~/.ssh/github
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /users/home/john/.ssh/github
Your public key has been saved in /users/home/john/.ssh/github.pub
The key fingerprint is:
SHA256:43+iLVviM8DhrX4vIJqzJjjHVa3ZDQC+H7Yg3neAy1Y machine@localhost.localdomain
The key's randomart image is:
+--[ED25519 256]--+
|     ..          |
|    .  .         |
|     .  o        |
|      oo o       |
|   . ++E*So      |
|  . +o*B=o..     |
| ... +=.+++ .    |
|+ B.  ..o*=. .   |
| =.o  ...+O+o    |
+——[SHA256]————+

$ echo -e "Host github.com\n  IdentityFile ~/.ssh/git" > ~/.ssh/config

$ cat ~/.ssh/github.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIF+XICi83RWt2HGkWuwhRhDVjKKMocha2A8SpR7iqxdE machine@localhost.localdomain
```


Dashboard
Type / to search
nikhilnathr
Top repositories
New
Find a repository...
artus-academy/website
nikhilnathr/git-workshop
prodojo/workshops
nikhilnathr/flirterai
sidharth-n/flirterai
nikhilnathr/cryptonoteAPI-visualiser
nikhilnathr/alien-log-reader
Show more
Join GitHub Education!
GitHub Education opens doors to new skills, tools, and a collaborative community eager to drive innovation. Join us and build a foundation for your future in technology.
Free and discounted services for teachers and students.
Copilot
Turn natural language prompts into coding suggestions
Heroku
Build, run, and operate applications entirely in the cloud.
Microsoft Azure
Access to Microsoft Azure cloud services and learning resources
Join GitHub Education
Home
Ask anything
nikhilnathr
Nikhil Nath R
Set status
Profile
Repositories
Stars
Gists
Organizations
New task
Enterprises
Sponsors
Settings
Copilot settings
Feature preview
Appearance
Accessibility
Try Enterprise
Free
Sign out
Go to settings page

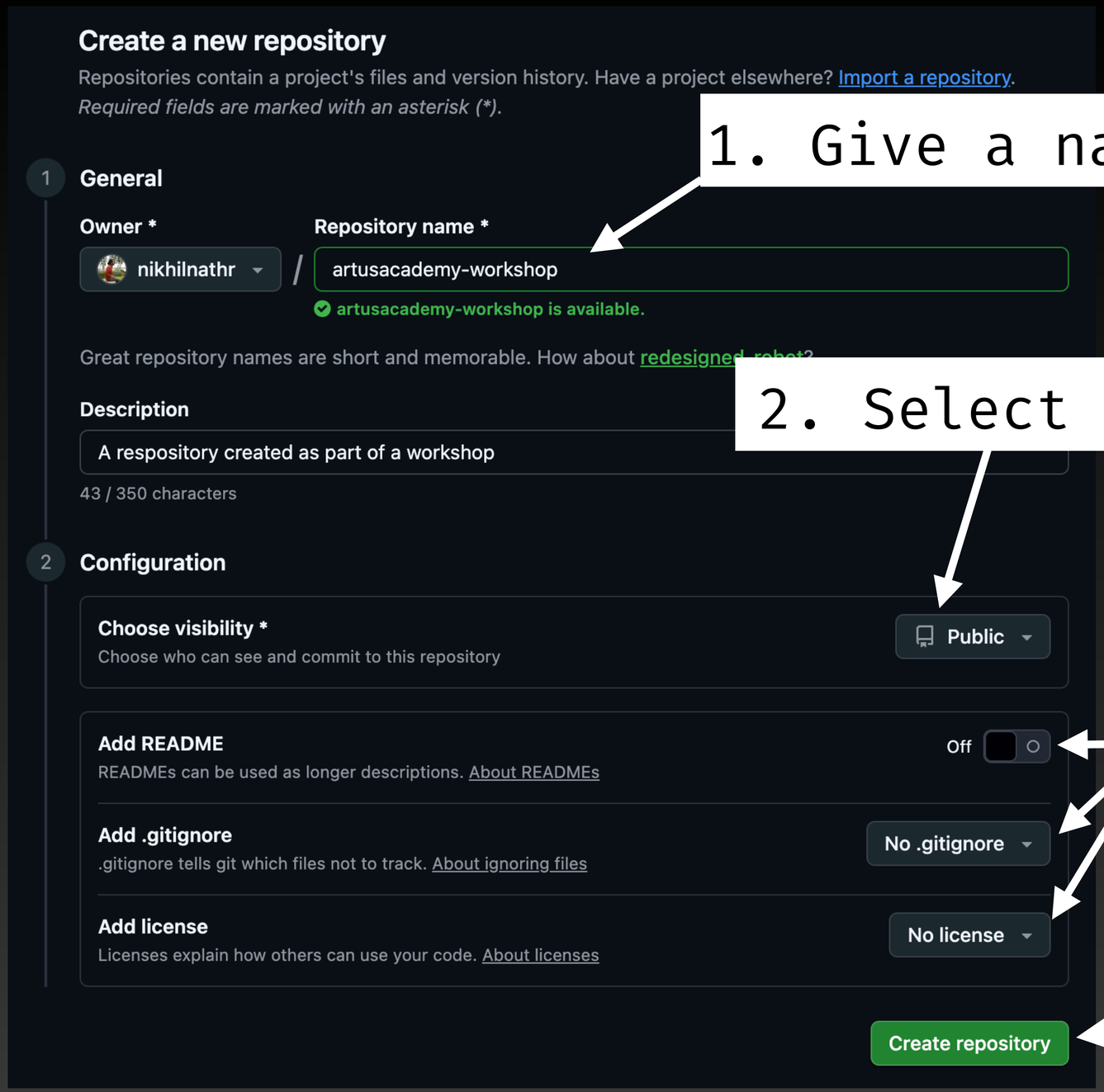

Settings
Type / to search
Nikhil Nath (nikhilnathr)
Your personal account
Switch settings context
Go to your personal profile
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation
Packages
Copilot
Pages
Saved replies
Security
Code security
SSH keys
New SSH key
This is a list of SSH keys associa
Authentication keys
mac
SHA256:QUiZIR/R7hQi5SCfTFMTYz9ywTeXVsa2smsFImnkuro
SSH
Added on Jul 31, 2024
Last used within the last 3 months — Read/write
Delete
Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.
GPG keys
New GPG key
This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.
Email address: nikhilnath1000@gmail.com
Key ID: 9313C70D1E543DCE
Delete
your account.
Vigilant mode
Flag unsigned commits as unverified
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
Learn about vigilant mode.
1. Select “SSH and GPG keys”
2.Click “New SSH key” button


Add new SSH Key
1. Give a name to your key
Title
workshop-key
Key type
Authentication Key
2. Paste you SSH public key
Key
Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'
3.Click “Add SSH key”
Add SSH key

1. Click the “+” dropdown



2. Select “New repository”

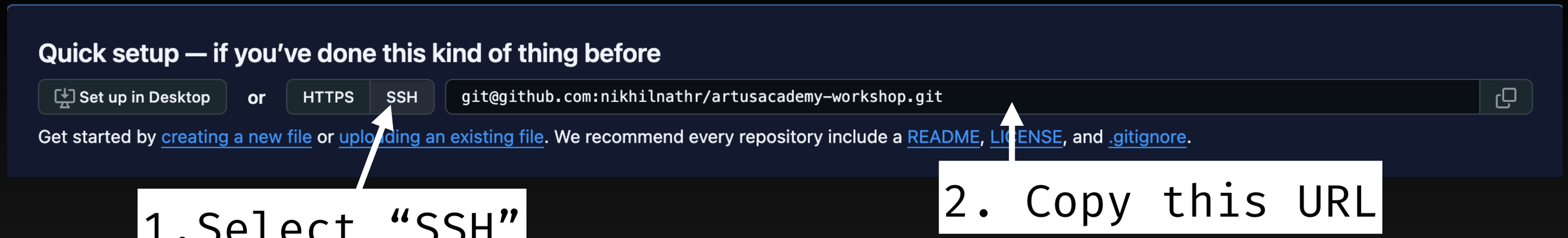


1. Give a name to your repository

2. Select public (open-source ❤️)

3. Keep these options unchecked
Or as “none”

4. Create repository

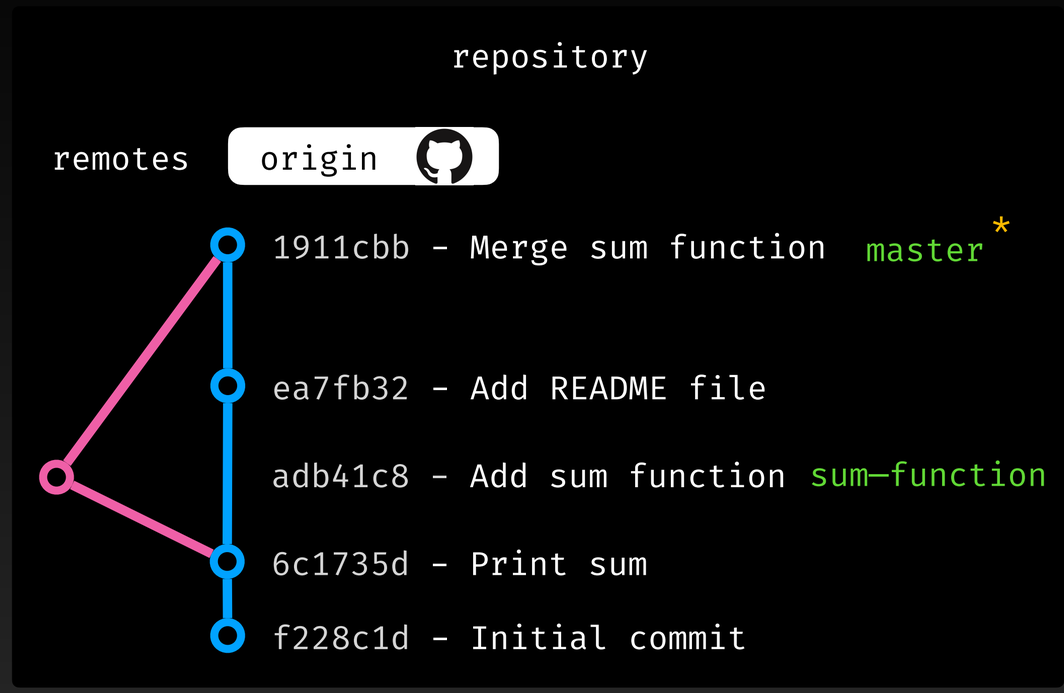

Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
git@github.com:nikhilnathr/artusacademy-workshop.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
1.Select "SSH"
2. Copy this URL

git remote

https://git-scm.com/docs/git-remote

- Manage the set of repositories, also called **remotes**
- `git remote add origin <remote-url>`
- Here **origin** is the name of the remote, there can be other remotes also.

```
$ git remote add origin git@github.com:nikhilnathr/
  git-workshop.git
$
```

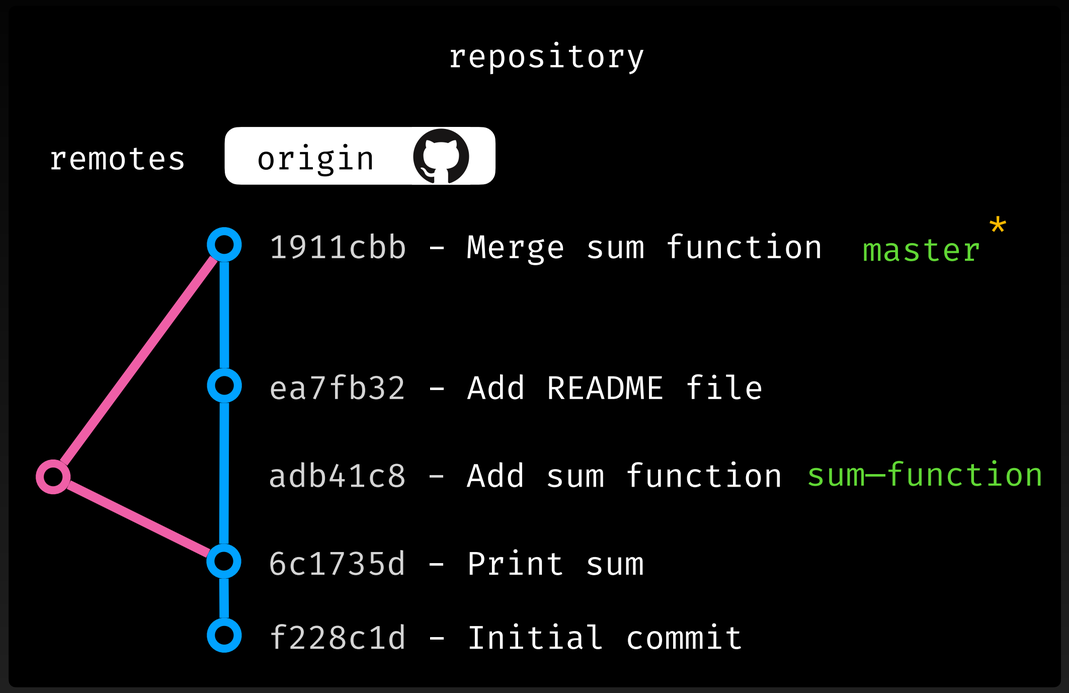


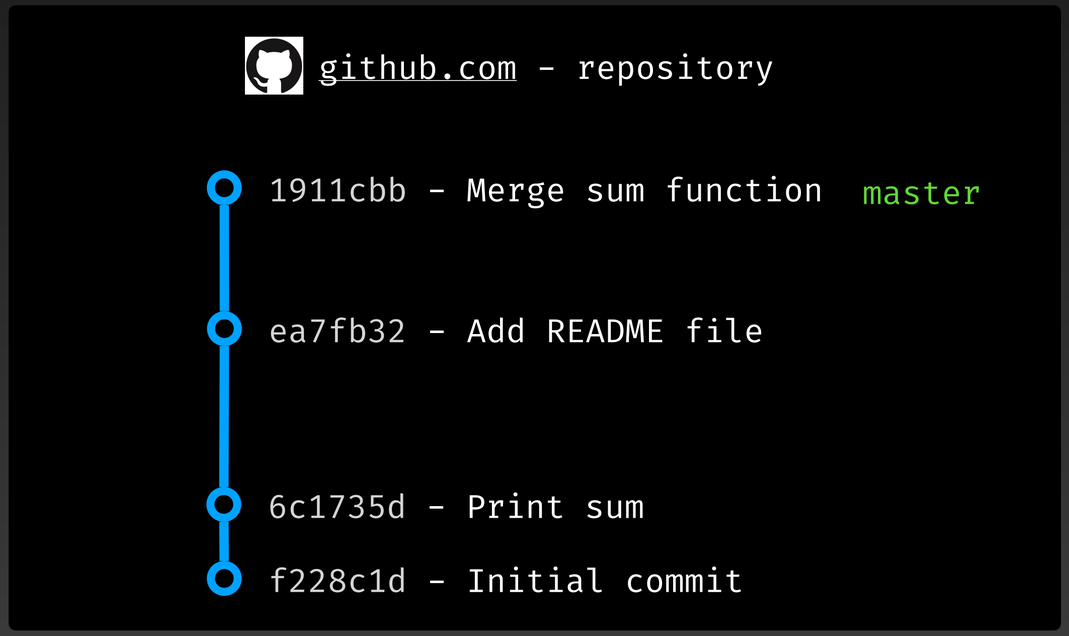
git push

https://git-scm.com/docs/git-push

- Updates remote using local, by sending objects necessary to complete the given refs.
- `git push -u origin <branch_name>` - when pushing a branch for first time
- `git push` - for subsequent updates

```
$ git push -u origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (14/14), 1.43 KiB | 1.43 MiB/s, done.
Total 14 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:johndoe/git-workshop.git
 + 8a4bdf6 ... 1911cbb master → master
branch 'master' set up to track ‘origin/master'.

$
```





git pull

https://git-scm.com/docs/git-pull

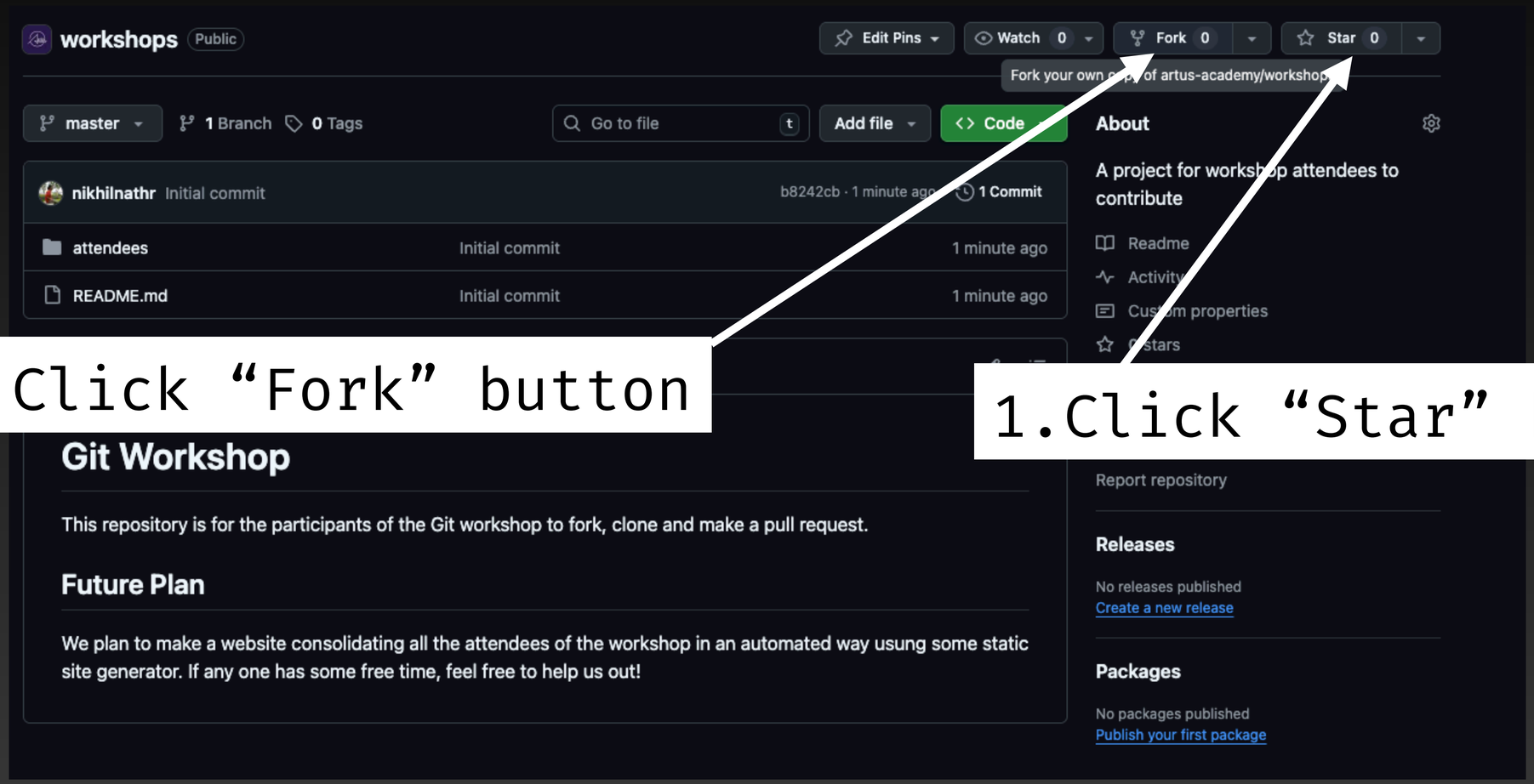
- Incorporates changes from a remote repository into the current branch.

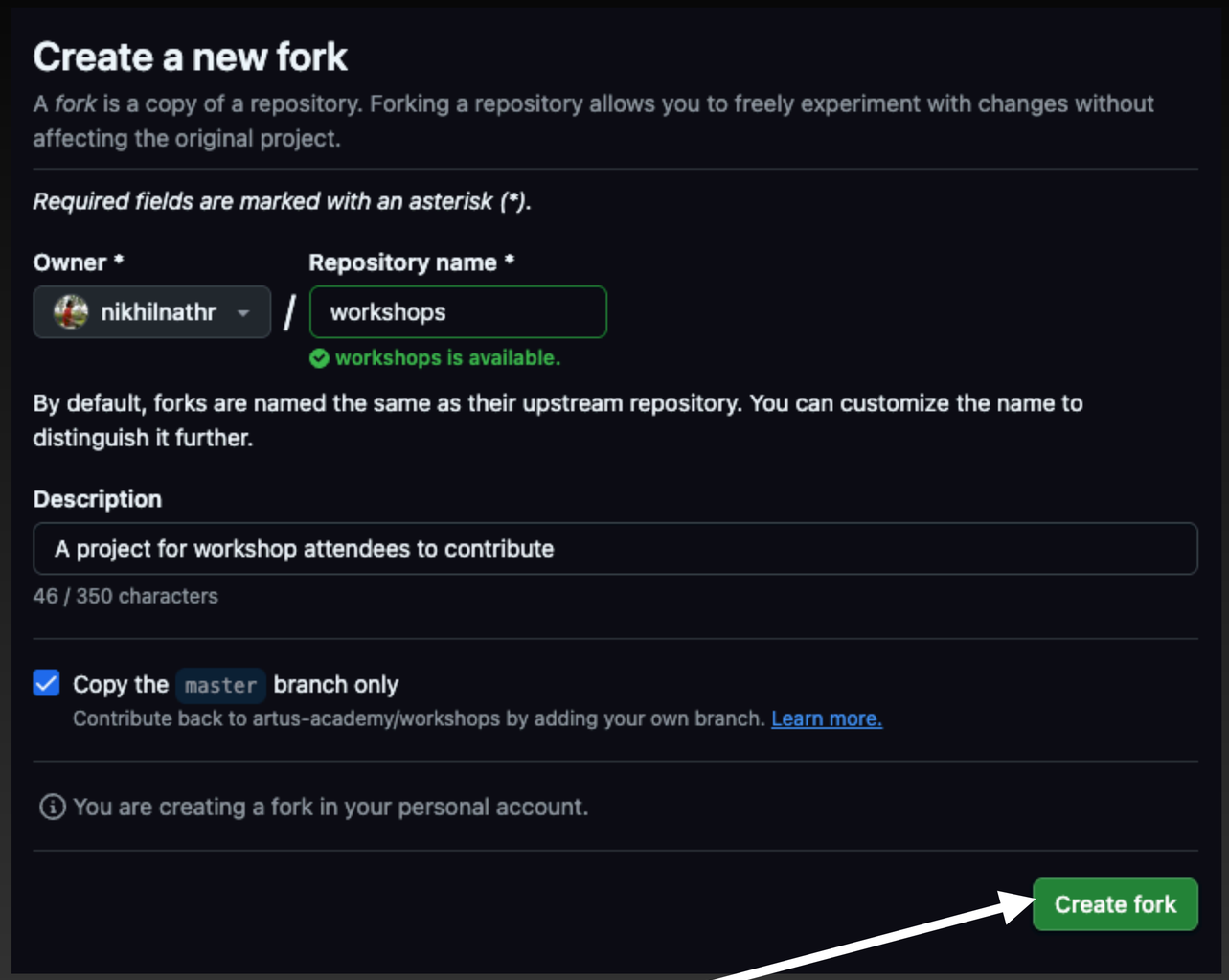
Contribute to project

forking

- A **fork** generally refers to creating a copy of a repository.
- This copy allows you to make changes independently from the original repository.
- On GitHub, when you fork a repository, it creates a personal copy of the repository under your own account.

https://github.com/artus-academy/workshops





Click “create fork” button

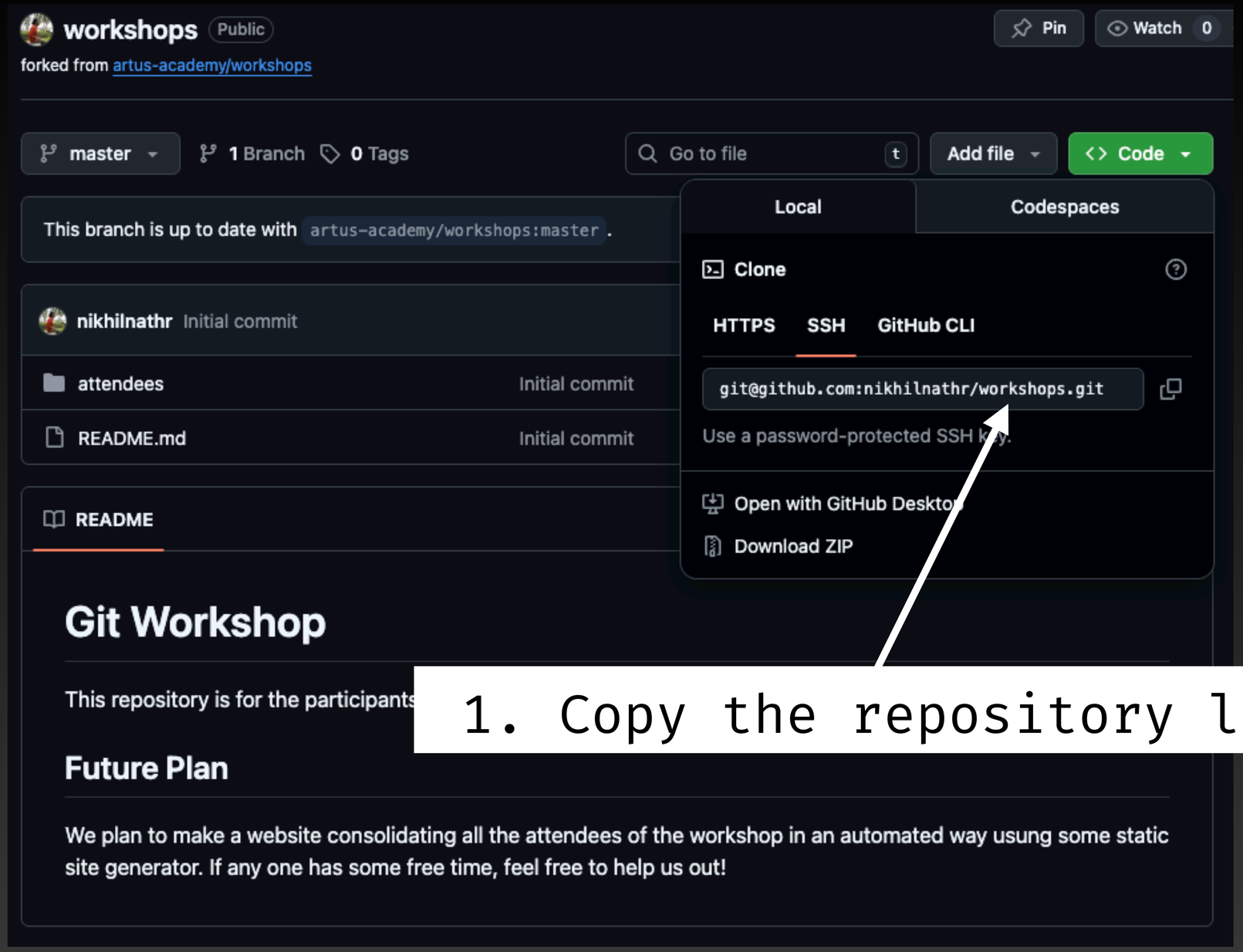
Commands

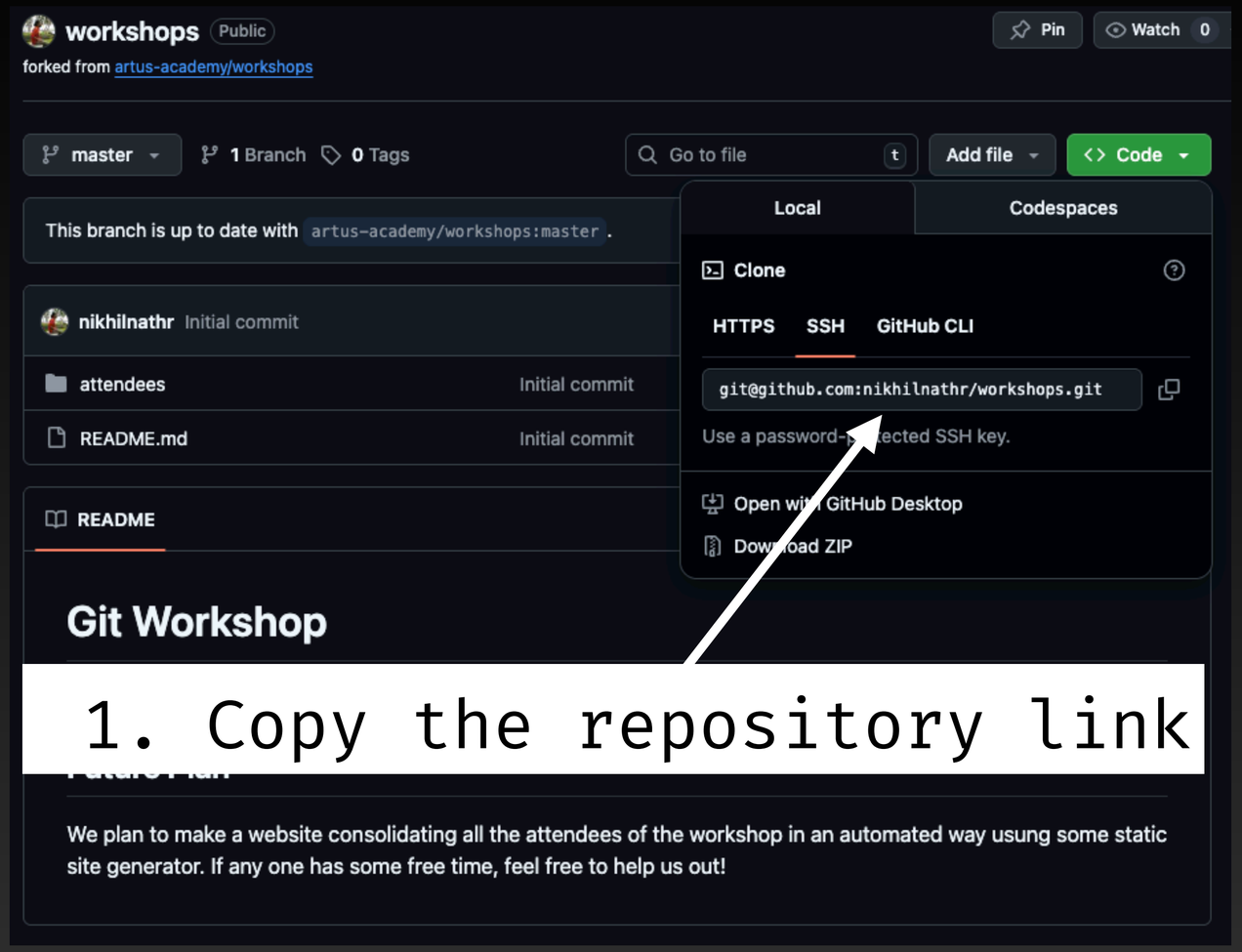
- `git clone`

git clone

https://git-scm.com/docs/git-clone

- Clones a repository into a newly created directory.
- Creates remote-tracking branches for each branch in the cloned repository


workshops
Public
forked from artus-academy/workshops
Pin
Watch 0
master
1 Branch
0 Tags
Go to file
t
Add file
Code
This branch is up to date with artus-academy/workshops:master .
Local
Codespaces
Clone
HTTPS
SSH
GitHub CLI
git@github.com:nikhilnathr/workshops.git
Use a password-protected SSH key.
Open with GitHub Desktop
Download ZIP
nikhilnathr Initial commit
attendees Initial commit
README.md Initial commit
README
Git Workshop
This repository is for the participants
Future Plan
We plan to make a website consolidating all the attendees of the workshop in an automated way usung some static site generator. If any one has some free time, feel free to help us out!
1. Copy the repository link



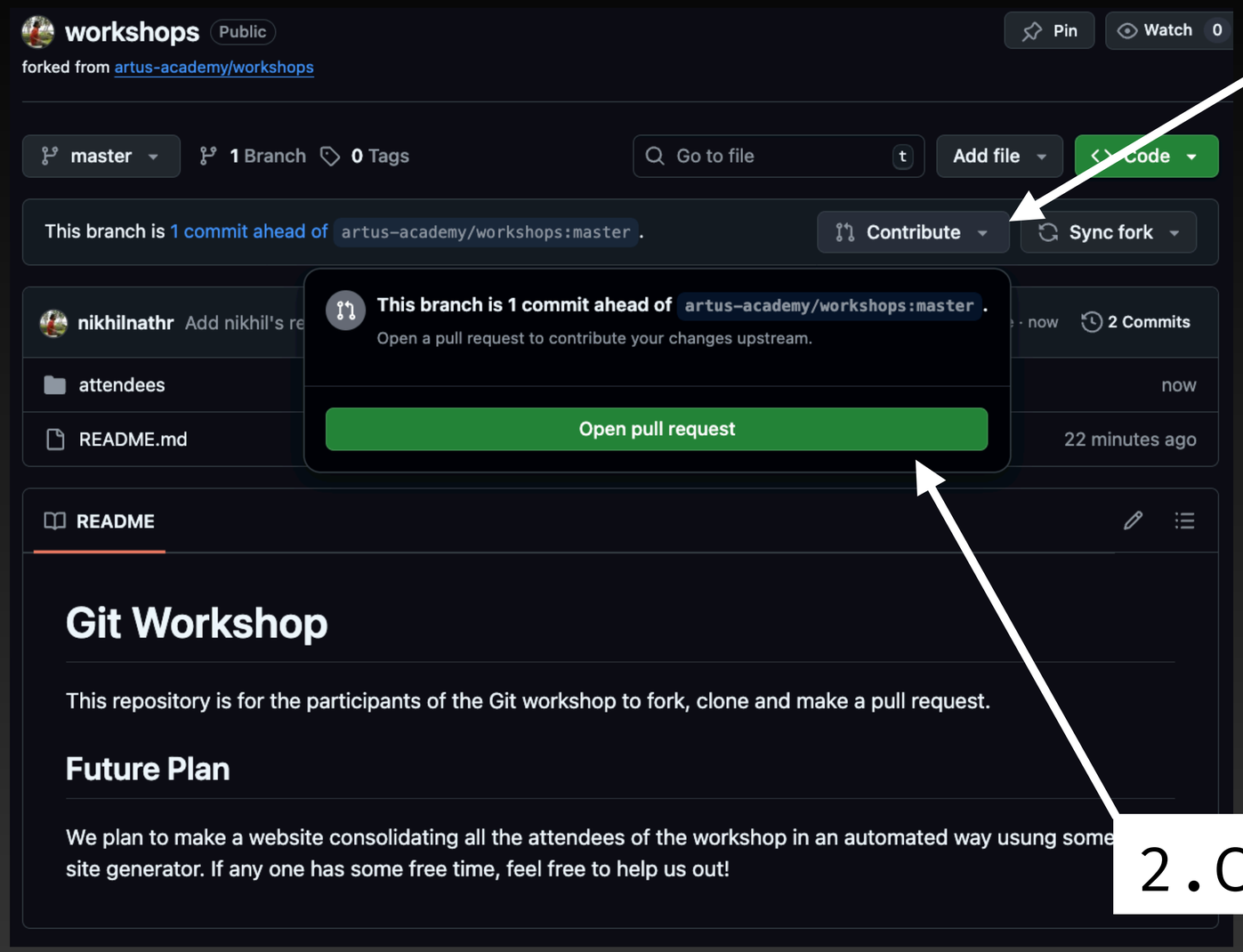

```
$ git clone git@github.com:nikhilnathr/workshops.git
Cloning into 'workshops' ...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 5 (delta 0),
pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.

$ cd workshops
$
```

Task

- Go to attendees folder
- Copy sample.md to <your_username>.md
- Edit <your_username>.md with your details
- Make a commit
- Push to your repository

merge / pull request

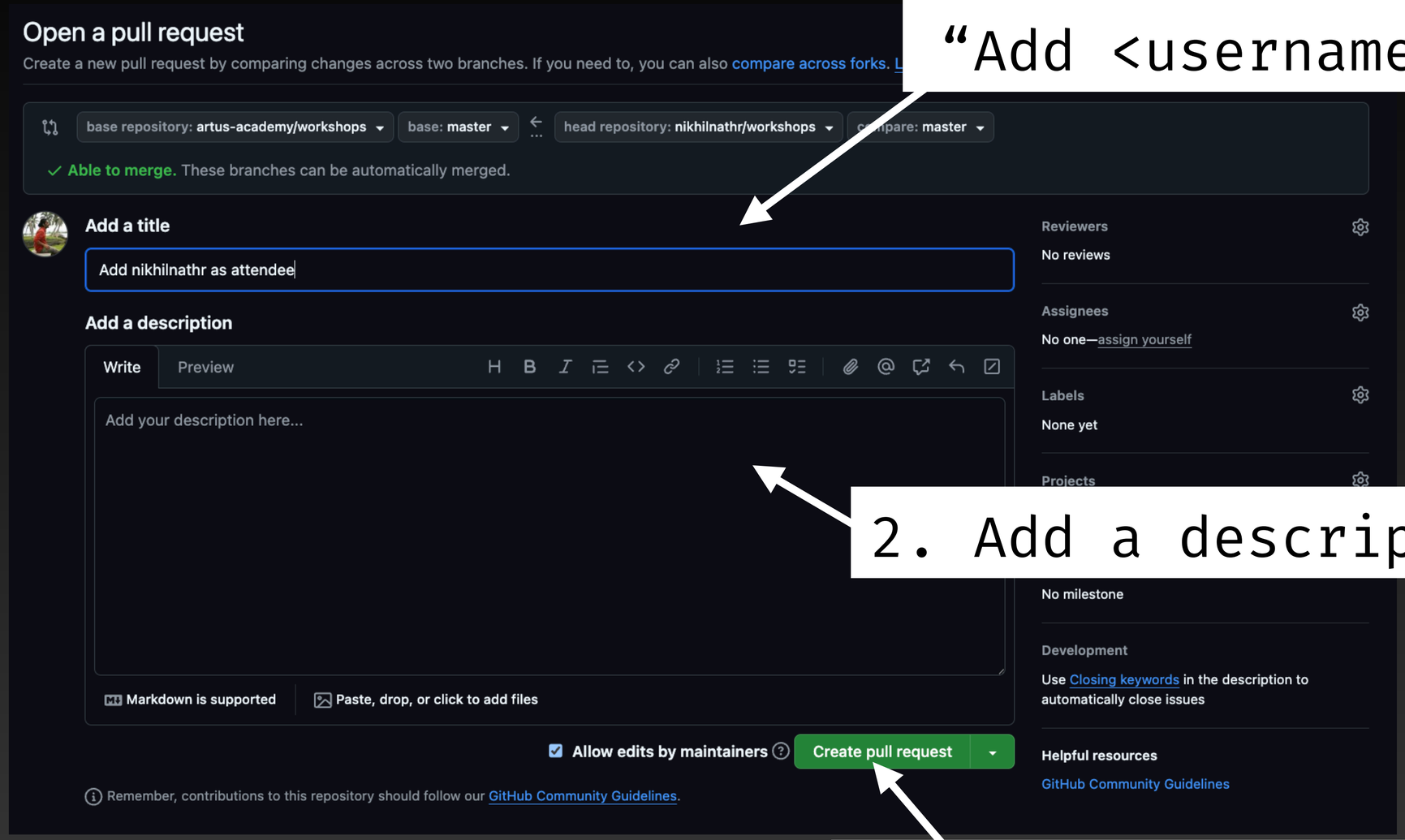
- A pull request is a way to propose changes you've made in a branch or fork to the original repository.
- It allows collaborators to review, discuss, and suggest improvements on the proposed changes before merging them.



1.Click “contribute”

2.Click “Open pull request”

1. Add a title
“Add <username> as attendee”



2. Add a description if required

3. Click “Create pull request”

Clean the files

- `rm ~/.gitconfig` - remove git configuration file with your name and email
- `rm ~/.ssh/github.pub ~/.ssh/github` - remove your public key
- `rm ~/workshop` - Delete the project

Thank You

Nikhil Nath R

nikhil@artusacademy.com
+91 9946624506
artusacademy.com