

Git & Github

Hope Townsend,
Aug 2023



I believe
in you!

Why should I care?

- Version control
 - Group work
 - Helps keep track of your own code and helps force better documentation
- Many tools are only accessible via Github
- Will be used in Software-Engineering for Scientists



<https://storage.googleapis.com/gweb-uniblog-publish-prod/images/Google-Docs-logo-transparent.max-2800x2800.png>

Git

-
- ```

if(parameters.contains("age")){
 hql += " and p.age = :age";
}

TypedQuery<Person> query = em.createQuery(hql);

if(parameters.contains("name")){
 query.setParameter("name", values[0]);
}

if(parameters.contains("age")){
 query.setParameter("age", Integer.valueOf(values[1]));
}

```

# Github

- Navigation to YMH Advanced Window

Dashboard

Home

Search

Hope2925

Top Repositories

Find a repository...

Hope2925/TranscriptFinderPro

Hope2925/LRproj

cu-swe4s-fall-2022/assignment-3-parallel-arrays-Hope2925

Dowell-Lab/sr2023

cu-swe4s-fall-2022/assignment-7-using-libraries-Hope2925

fanzhanglab/SCRNA-GWAS-Benchmarking

cu-csci-2270-fall-2022/final-project-Hope2925

Show more

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

For you

Following

Send feedback

Filter

Welcome to the new feed!

We're updating the cards and ranking all the time, so check back regularly. At first, you might need to follow some people or star some repositories to get started.

Send feedback

jtsanley

 created a repository · yesterday

jtsanley / branching\_demo

Star

A repo to demonstrate the use of branches

jtsanley

 created a repository · last week

jtsanley / another\_repo

Star

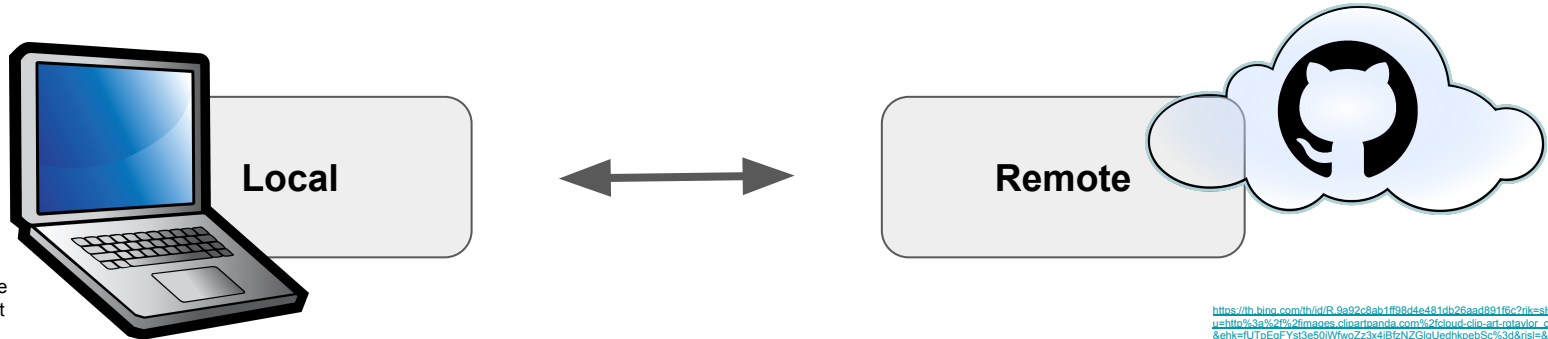
This is a repo created to demonstrate the use of GitHub

Jupyter Notebook

5

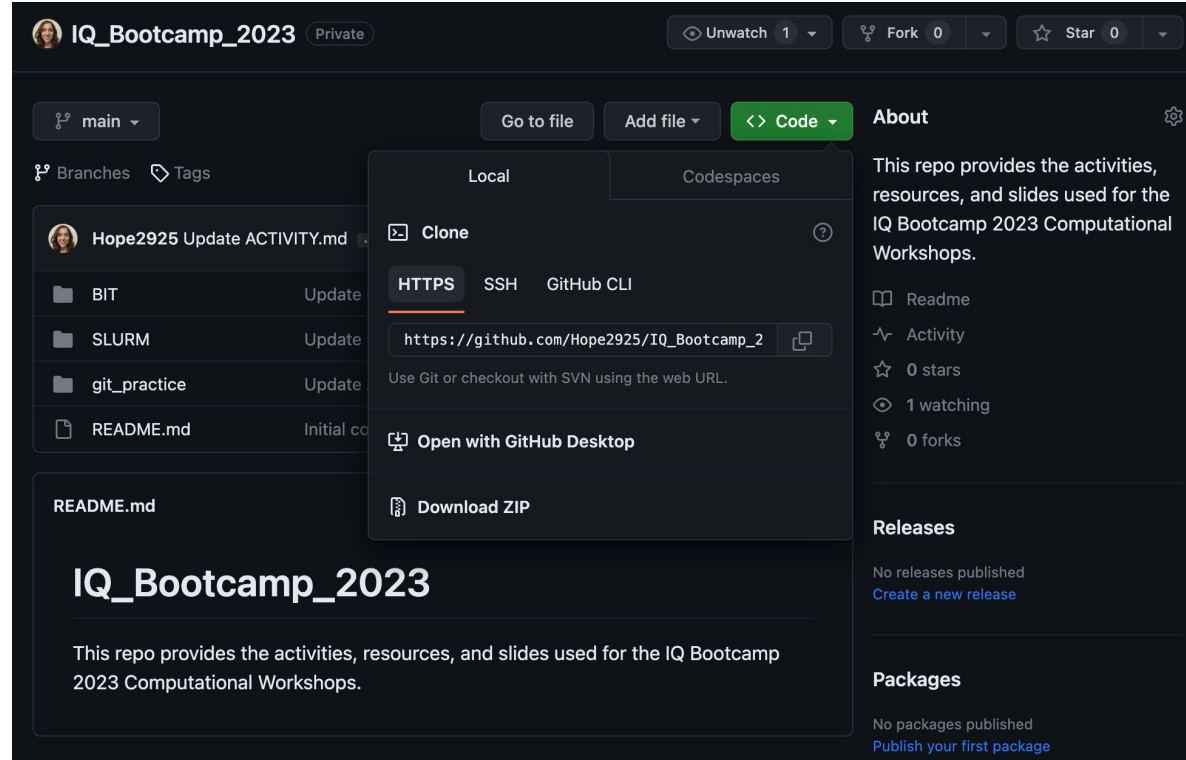
# Git Lingo

- **Repository**= folder with the version controlled content
  - Can contain folders & files within it
  - Usually one per project
- **README**= file with important information about repository (e.g. how to use it if tool, dependencies, other important notes)
- **Remote**= Github's version of a repository
- **Local**= Your personal machine's version of a repository



# How do we stay connected? (steps 1 & 2 of ACTIVITY.md)

1. Find Github repository (can create one or use another's)
2. Clone the repository:
  - a. Copy the https link



# Why is there https:// and ssh://?

- How can the remote system ensure only certain people can read/write the repositories?
  - It must be able to recognize the User securely
- 2/3 ways: Personal Access Tokens (PATs) (https) & SSH keys (ssh)
  - We will be using PATs today as they have some advantages over SSH. Interested in more?  
Click [here](#) for a short layman description and [here](#) for Github's description.
- But for now, let's see what happens if you don't have one.

Demo: Continue by  
following  
ACTIVITY.md Step 3  
on

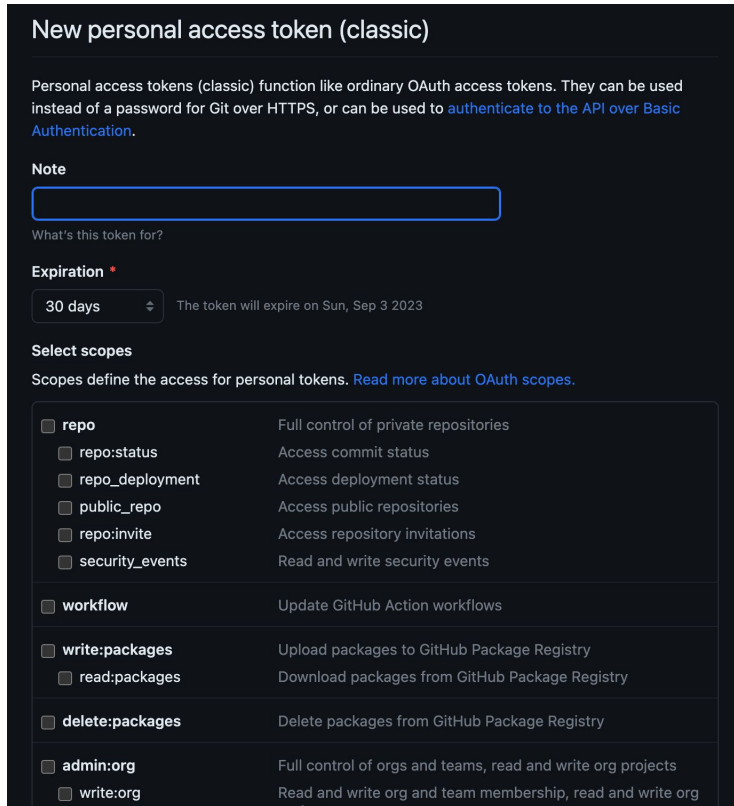
# Create a personal access token

Full instructions on [ACTIVITY.md](#)

1. Create a PAT
2. Type a name/note
3. Select the access you want
4. Copy the token password and save it somewhere on your computer

*This is what you'll type in as your password*

Alternatives?



The screenshot shows the GitHub interface for creating a new personal access token (classic). The form is titled 'New personal access token (classic)' and includes a description of what these tokens are used for. It features a 'Note' field, an 'Expiration' dropdown set to '30 days', and a 'Select scopes' section with various permissions like 'repo', 'workflow', 'write:packages', 'delete:packages', and 'admin:org'.

**New personal access token (classic)**

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git 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 Sun, Sep 3 2023

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

|                                                 |                                                             |
|-------------------------------------------------|-------------------------------------------------------------|
| <input type="checkbox"/> <b>repo</b>            | Full control of private repositories                        |
| <input type="checkbox"/> repo:status            | Access commit status                                        |
| <input type="checkbox"/> repo_deployment        | Access deployment status                                    |
| <input type="checkbox"/> public_repo            | Access public repositories                                  |
| <input type="checkbox"/> repo:invite            | Access repository invitations                               |
| <input type="checkbox"/> security_events        | Read and write security events                              |
| <input type="checkbox"/> <b>workflow</b>        | Update GitHub Action workflows                              |
| <input type="checkbox"/> <b>write:packages</b>  | Upload packages to GitHub Package Registry                  |
| <input type="checkbox"/> read:packages          | Download packages from GitHub Package Registry              |
| <input type="checkbox"/> <b>delete:packages</b> | Delete packages from GitHub Package Registry                |
| <input type="checkbox"/> <b>admin:org</b>       | Full control of orgs and teams, read and write org projects |
| <input type="checkbox"/> write:org              | Read and write org and team membership, read and write org  |

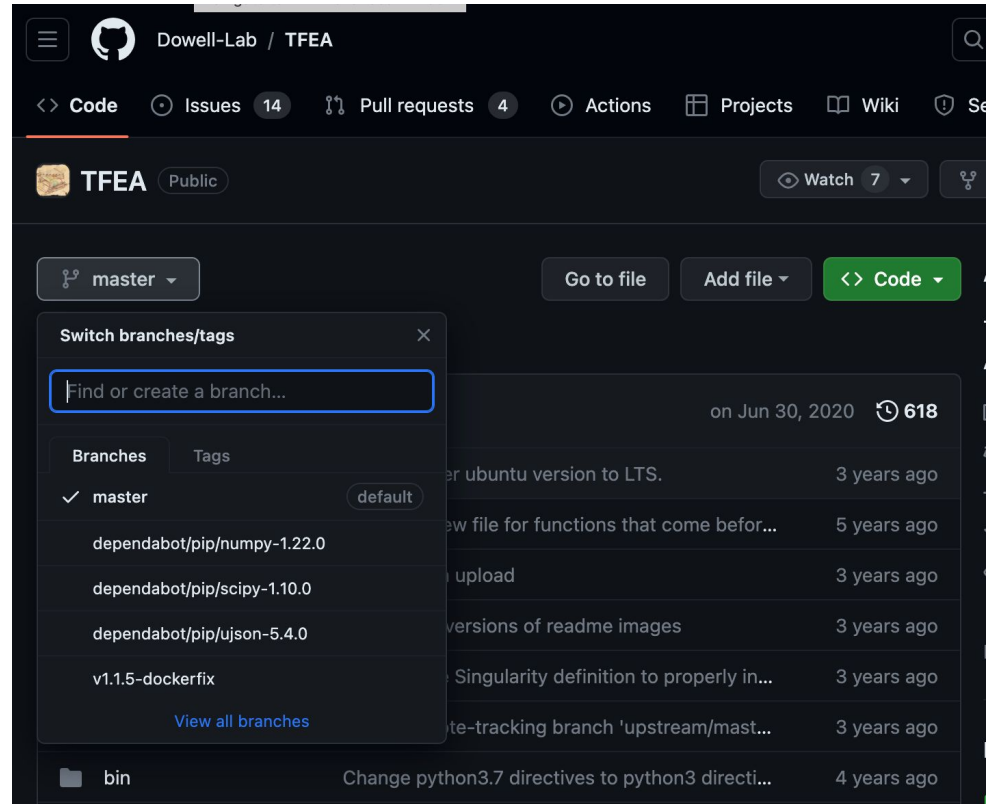


# If time... branches

How does a tool posted on Github make updates without messing up the current version?

Branches!!

Code for using branches can be found on the README



You made it!!

