

Graduate Training Programme 2020: Collaborating between Grads and Trainers using GitHub

Sep 2020
HSBC Technology Academy

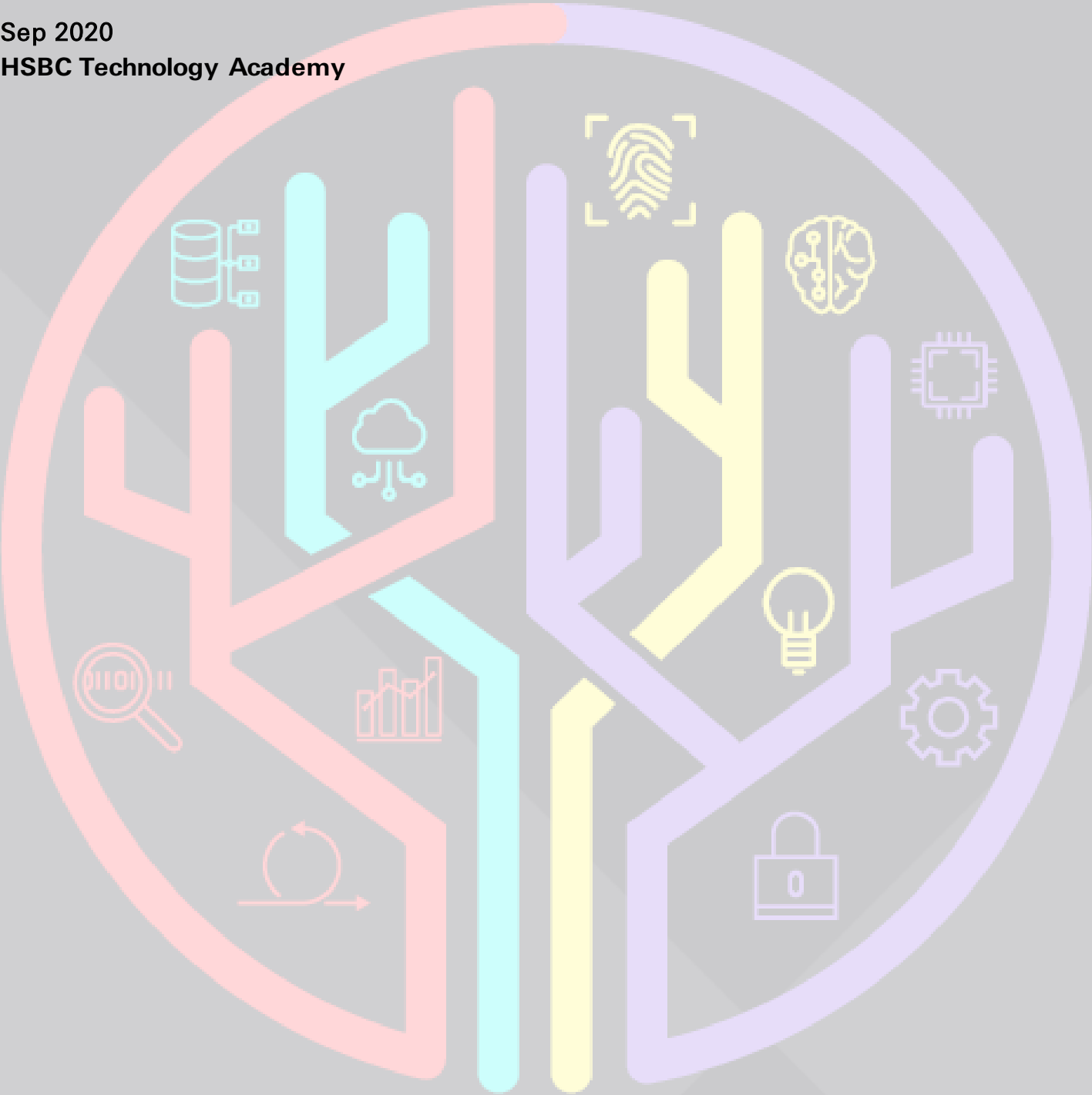
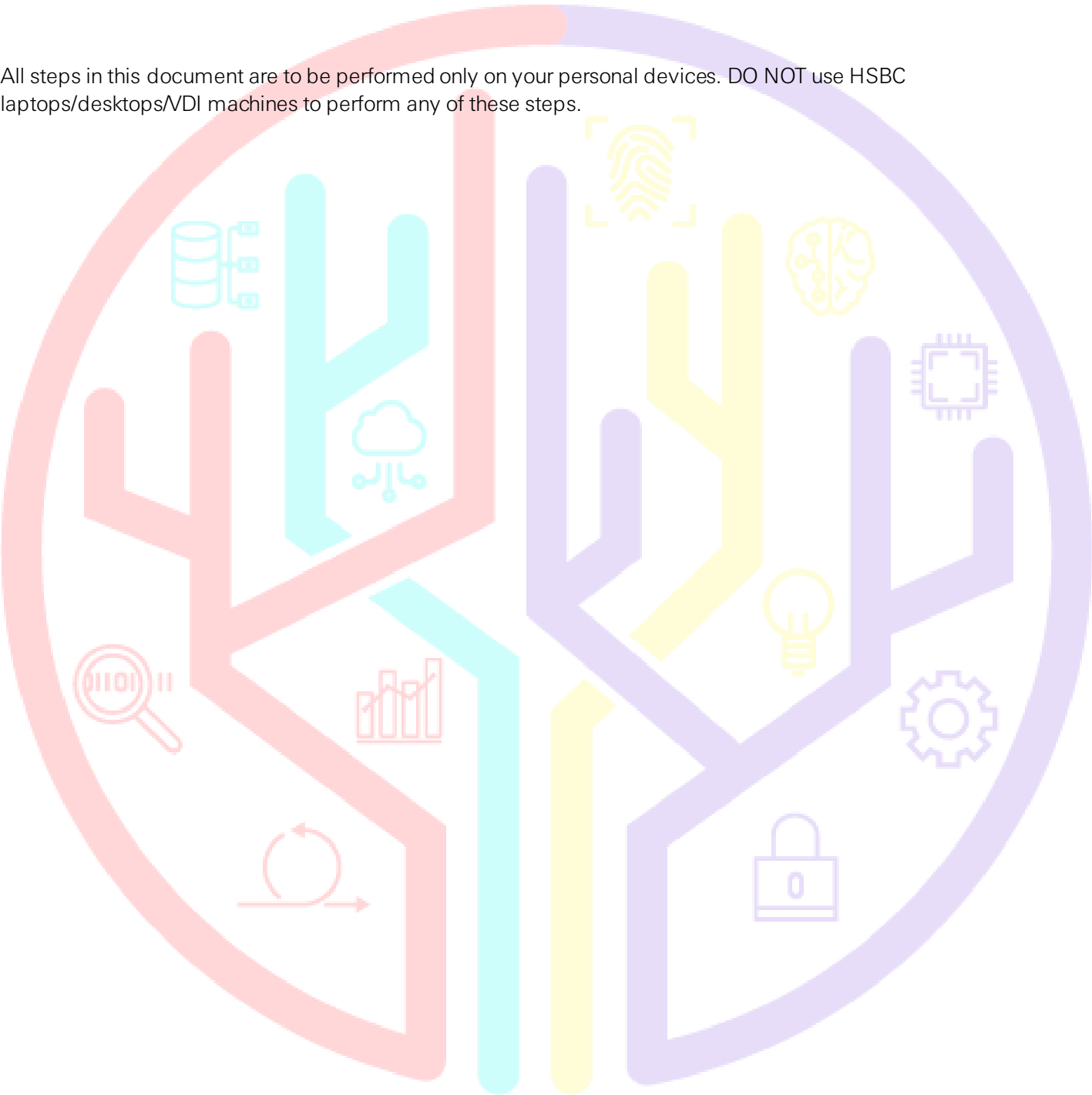


Table of contents

| | |
|--|----|
| IMPORTANT NOTICE | 2 |
| Setting up the training related repository | 3 |
| Accessing GitHub | 3 |
| Fork the trainer’s Git repository | 4 |
| Summary: Setting up the training related repository | 6 |
| Making changes to the repository | 7 |
| Using GitHub on web browser | 7 |
| Using GitHub Desktop | 8 |
| Other ways to work with Git | 12 |
| Summary: Making changes to the repository | 12 |
| Synchronizing between original (upstream) and forked (remote) repositories | 13 |
| Updating remote repository (of graduate) to reflect changes made to upstream repository (of trainer) | 13 |
| Updating upstream repository (of trainer) to reflect changes made to remote repository (of graduate) | 14 |
| Summary: Synchronizing between original (upstream) and forked (remote) repositories | 15 |

IMPORTANT NOTICE

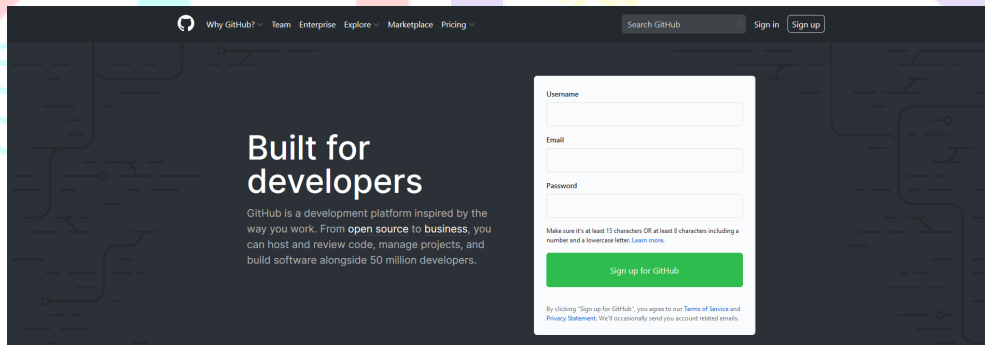
All steps in this document are to be performed only on your personal devices. DO NOT use HSBC laptops/desktops/VDI machines to perform any of these steps.



Setting up the training related repository

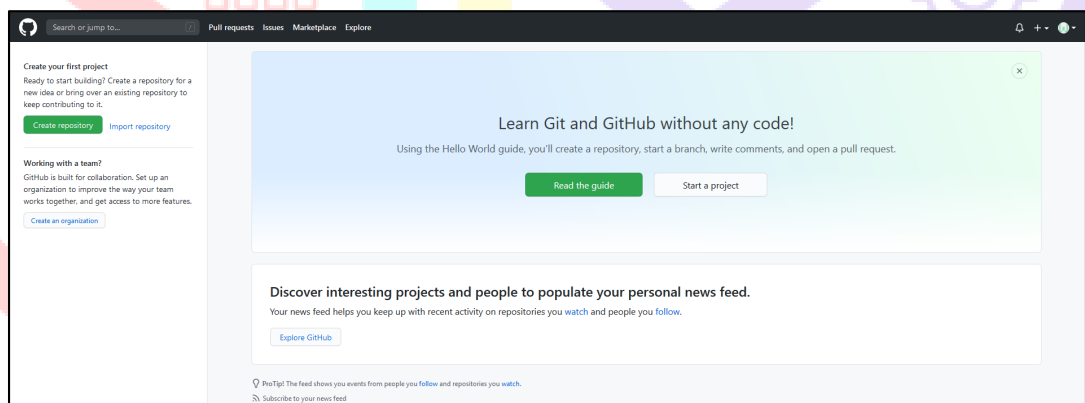
Accessing GitHub

- ◆ Navigate to <https://github.com/>
- ◆ If you already have an account, click **Sign in**, Otherwise **Sign up**

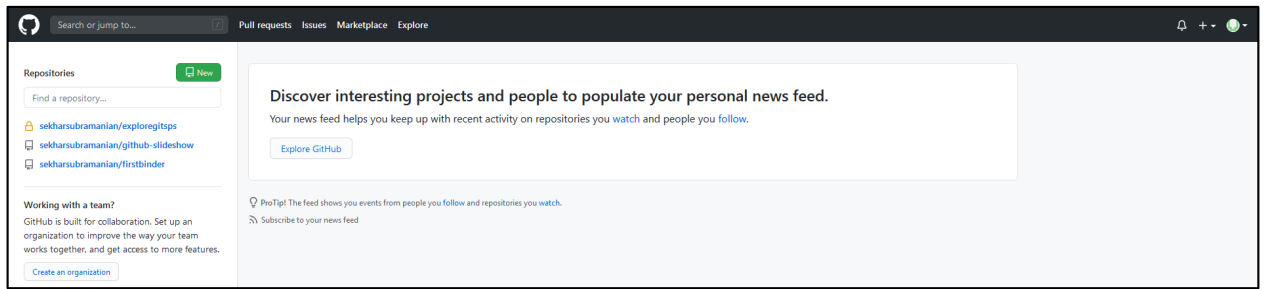


Homepage of GitHub

- ◆ When signing up for GitHub:
 - Create a **unique username**, a **strong password** and provide a **valid email address** (use your personal email address)
 - After creating the account, verify your email address by logging into your email and follow the instructions in the email sent by GitHub
- ◆ Log in to your GitHub account after verifying your email address



Home screen of a new GitHub User



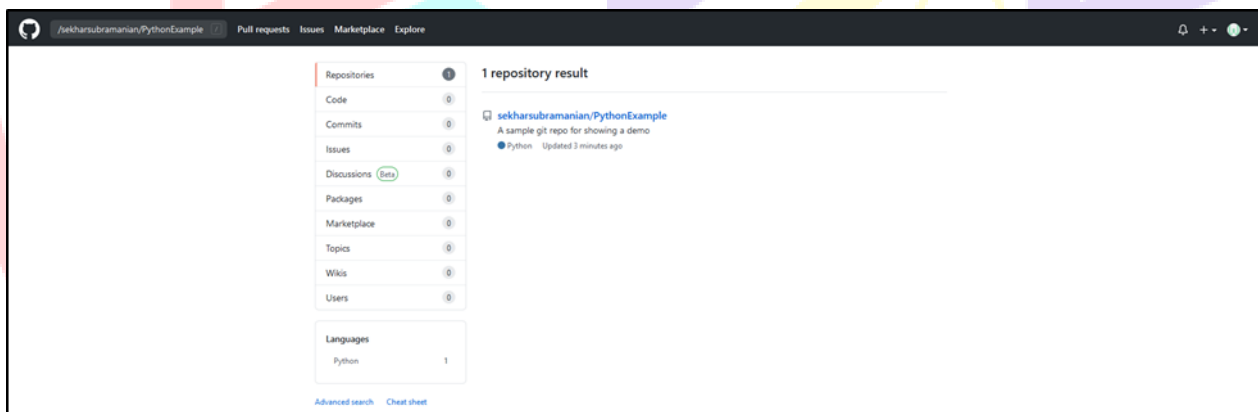
Home screen of a GitHub user with repositories

Fork the trainer's Git repository

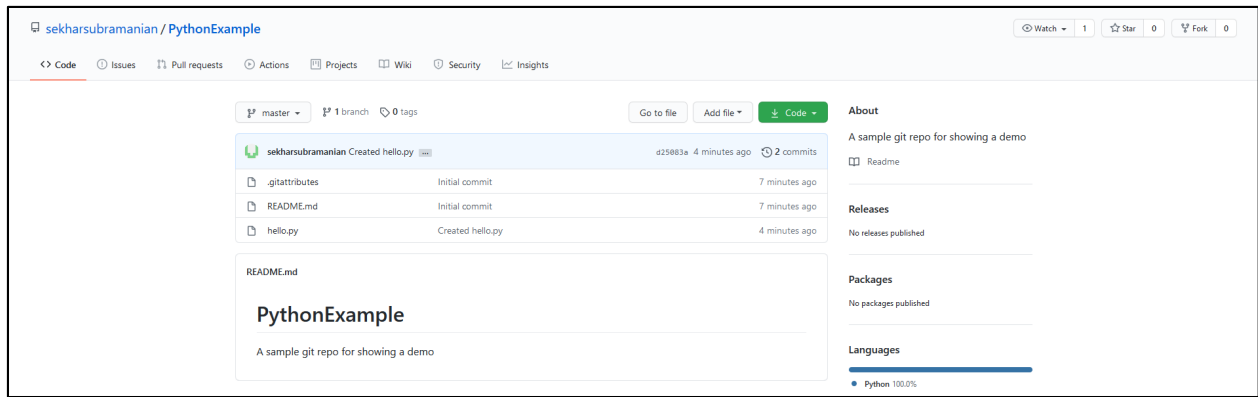
The trainer will share learning content such as demo programs, assignment documents, presentations etc. by uploading them onto their GitHub repository. Forking their repository ensures that you (the graduate) have a copy of the material in a new repository that you are in control on your GitHub profile. There are other options such as cloning or branching the repository as well, however for the purpose of the training we shall use the fork option.

Forking a repository means that you own a copy of the repository and the changes that you make in your version do not affect the main repository that you forked from. In order to update the main repository you will need to create a pull request (covered later in this document) which is not very common thing in the context of this training.


- ◆ Find the trainer's repository either using the search bar or by navigating to it directly if the link is shared with you.
- ◆ In this document, we are using a simple repository for demo purposes. The link to access it is: <https://github.com/sekharsubramanian/PythonExample>

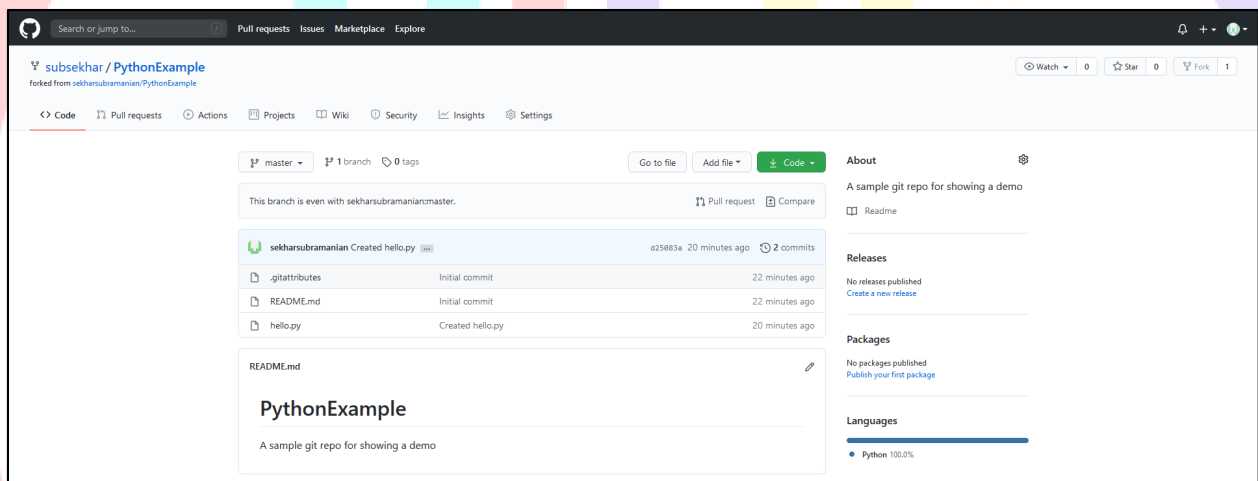


Searching for the GitHub repository (Use search bar on top-left of screen).



Viewing the repository on GitHub

- ◆ Click **Fork** on  **Fork** **0** (fork button) present near the top-right corner of the repository page.
- ◆ Observe that a copy of the repository gets created in your profile

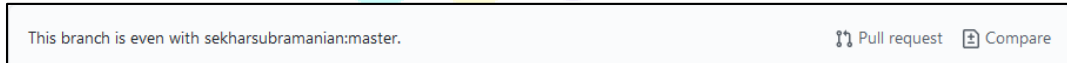


Screenshot of a forked GitHub repository

- ◆ Make note of the following key points visible on the page
 - Repository name and forking information - The repository name remains same as original but it is now associated with your username. You can also observe that it shows which repository it has been forked from.



- Status of your repository in comparison with the original repository that it was forked from. Here since we have just forked the repository, both repositories are even. As we go along the document we will notice how this changes.

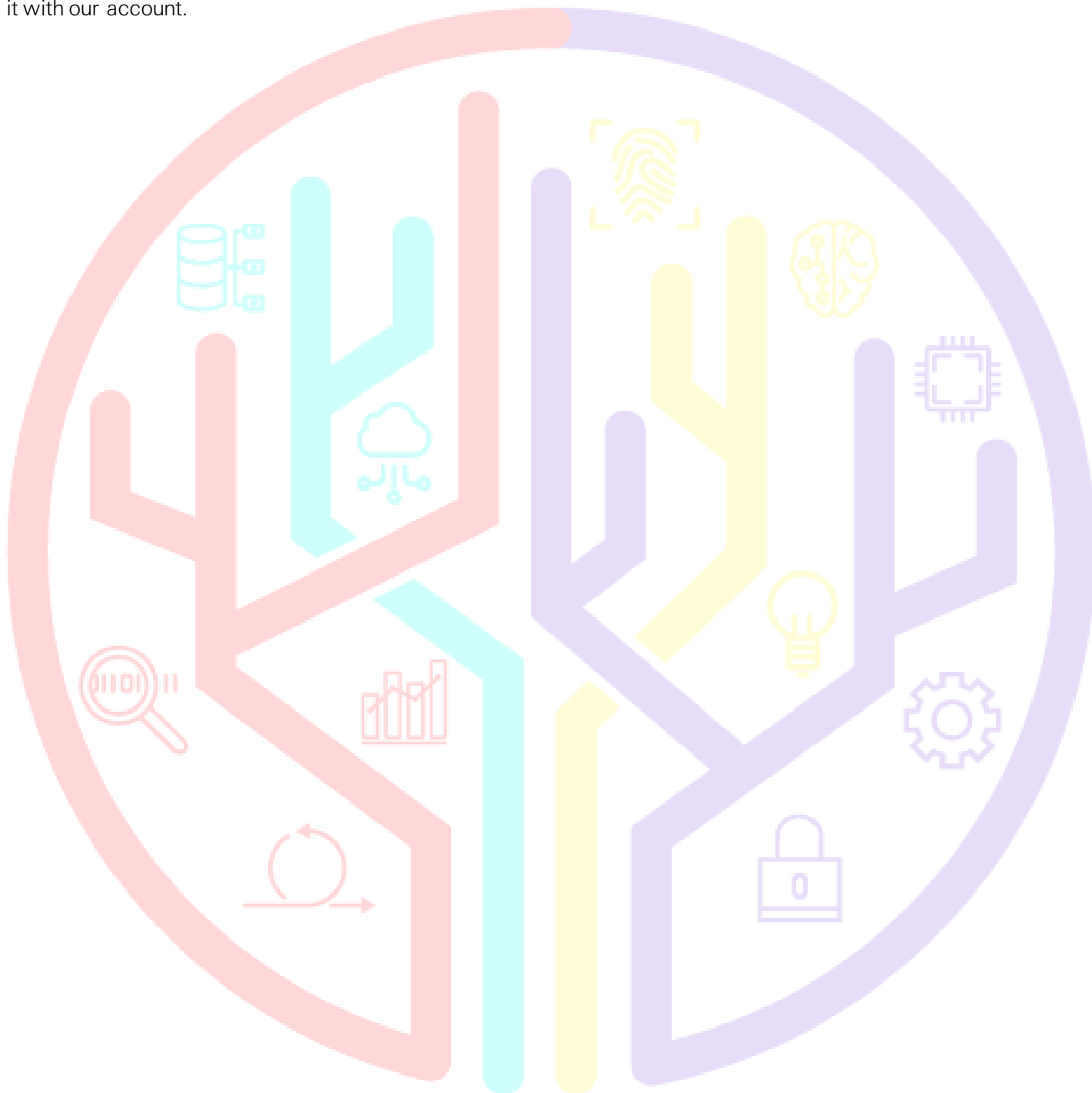


- ◆ Note that after the fork, you have the files associated with your username. However so far these are all on the remote GitHub server and you do not have these files on your local hard disk. The repository that you see on

GitHub is called the **'remote'** repository. Also, the original repository that you forked from is called **'upstream'** repository.

Summary: Setting up the training related repository

So far we have setup a GitHub account and forked the trainer's repository there by making a copy of it and associated it with our account.



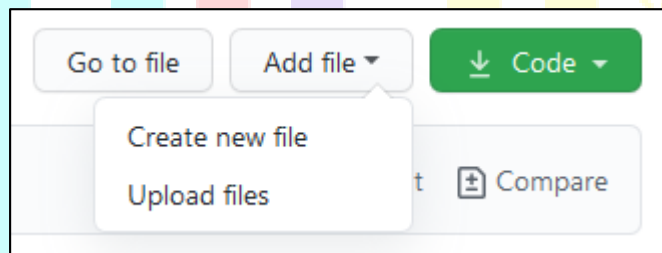
Making changes to the repository

Using GitHub on web browser

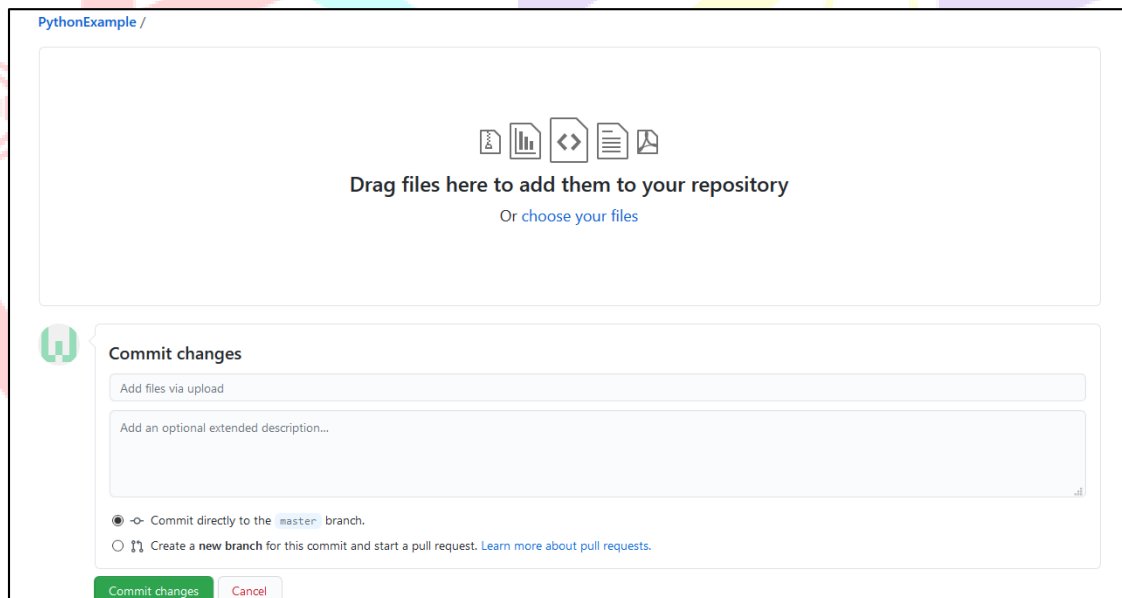
We can use the repository page on GitHub website to make changes to our repository. For example, we can upload files and create new files using the web interface. We will also be able to delete any files.

This might seem like a good approach when we want to work with just a few files. This is however NOT the most popular/recommended approaches to managing files in the repository and is not a practical solution for projects

- ◆ To add files into a repository from the web interface, click Add File → Upload files

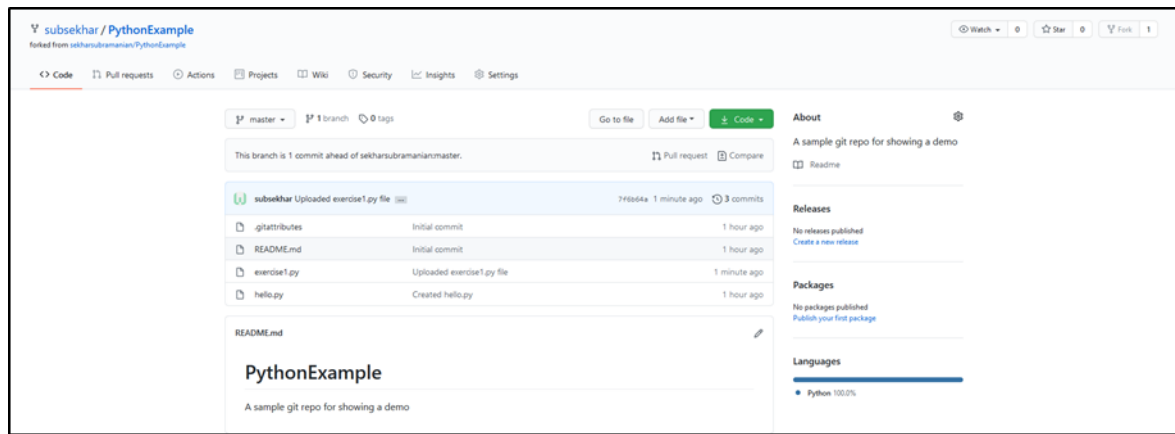


- ◆ Select the files to upload using the interface and provide appropriate title and description in the 'Commit changes' section. Leave the default option 'Commit directly to the master branch' selected and click 'Commit changes' button.

A screenshot of the GitHub 'Commit changes' form. The form is titled 'Commit changes' and has a 'Commit changes' button at the bottom. It includes a section for 'Add files via upload' with a text input field for an optional extended description. Below this, there are two radio button options: 'Commit directly to the master branch.' (selected) and 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' The form is set against a background of a repository page for 'PythonExample'.

Upload files via web interface

- ◆ View the updated repository on GitHub and observe that the files are uploaded, the commit history has changed and the comparison with the upstream repository (trianer's repository) has been updated.

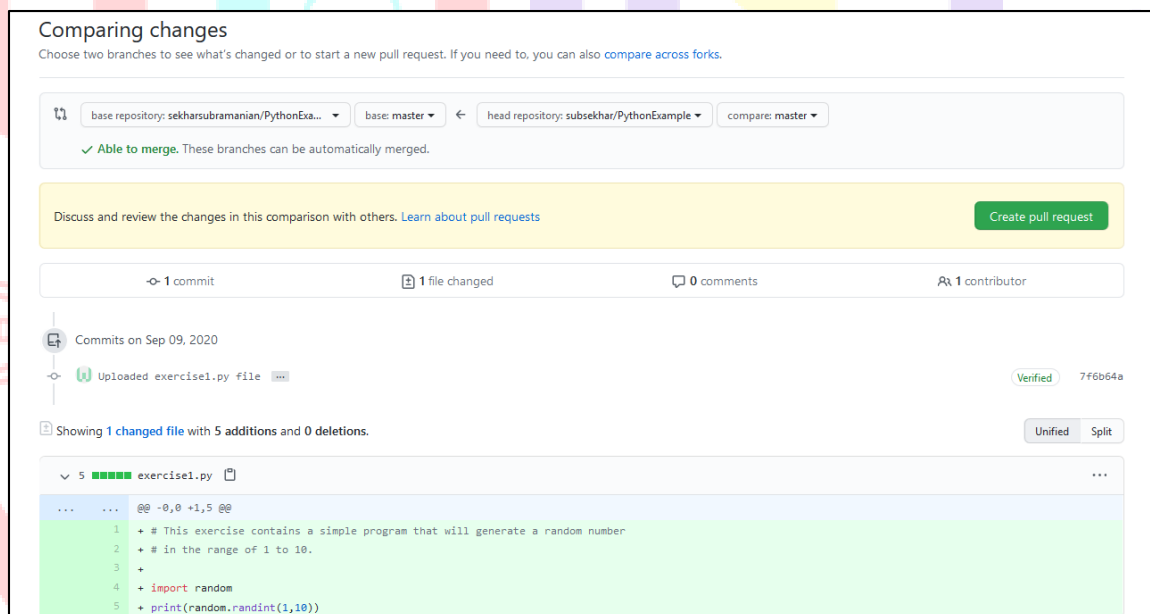


Repository updated after uploading files



Status of repository compared to the upstream repository

- ◆ Click compare and view the difference between current remote repository and upstream repository. Note that clicking compare actually navigates to the web page of the upstream repository. GitHub also analyzes if the two repositories can be merged without any conflicts.

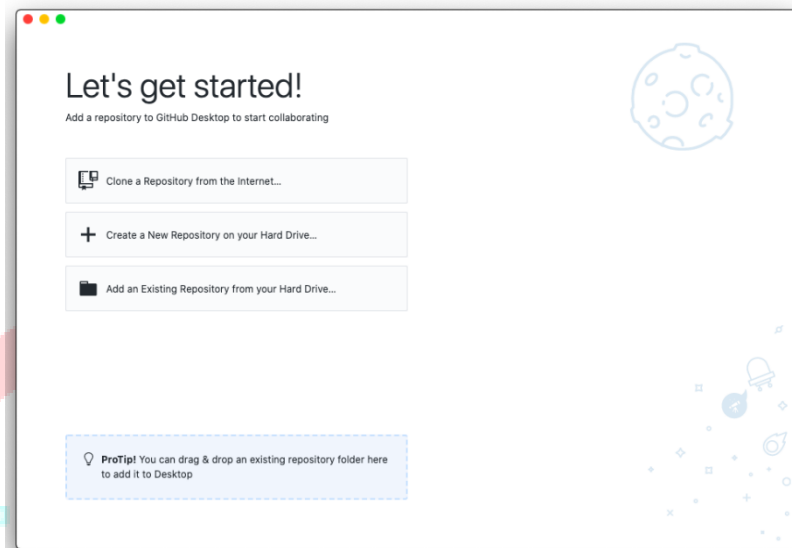


Comparing changes between upstream and remote repository

Using GitHub Desktop

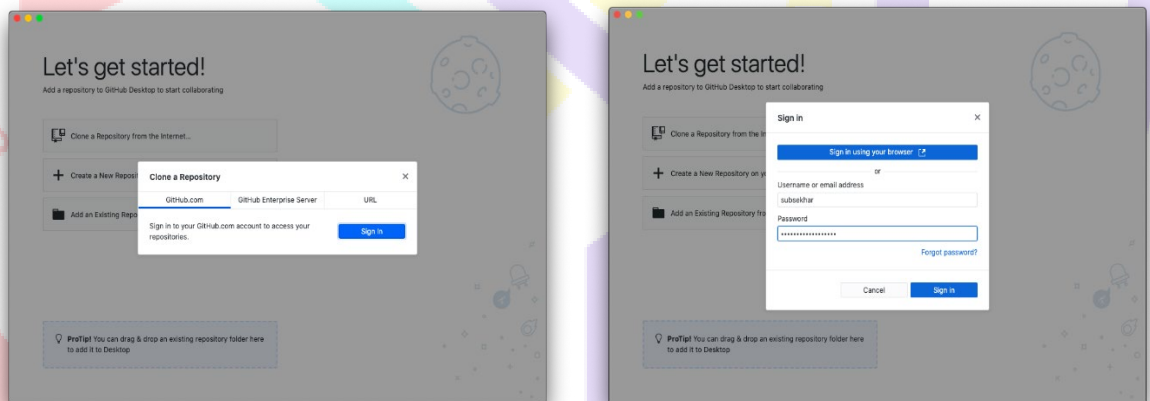
GitHub Desktop is an intuitive desktop application that can be installed on your computer and helps you with managing your repositories. It can be downloaded from: <https://desktop.github.com/>

- ◆ Launch GitHub Desktop after successfully installing it on your computer



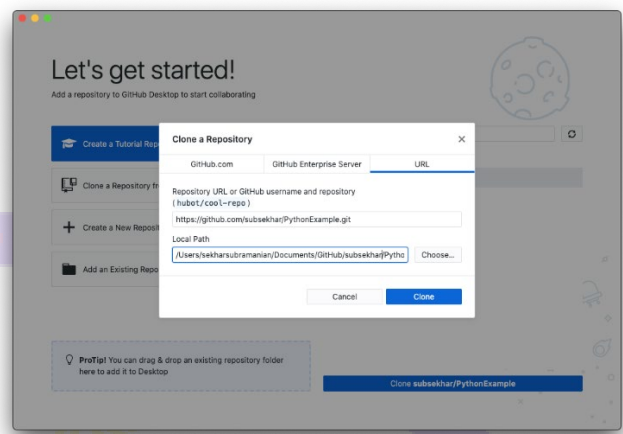
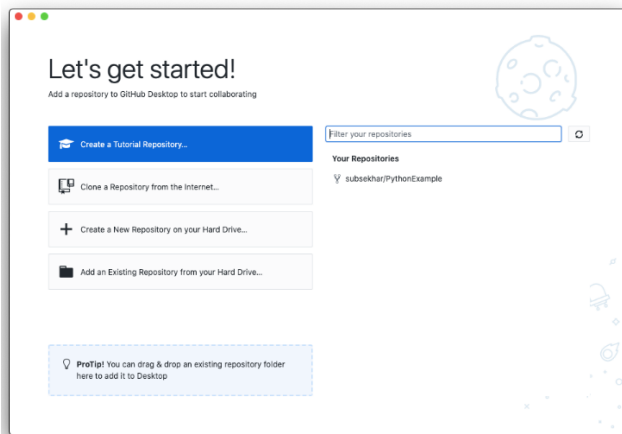
Main screen of GitHub Desktop

- ◆ The various options presented in the main screen are as follows:
 - Clone a repository from the Internet: This option downloads files from a 'remote' repository onto the computer's local disk and creates a 'local' repository.
 - Create a new repository on your hard drive: This option creates a new 'local' repository on the computer's local disk
 - Add an existing repository from your hard drive: Can be used to configure the GitHub desktop client with repositories already created/cloned on the local disk
- ◆ Choose **Clone a Repository from the Internet** and provide the authentication credentials to the GitHub account setup earlier.



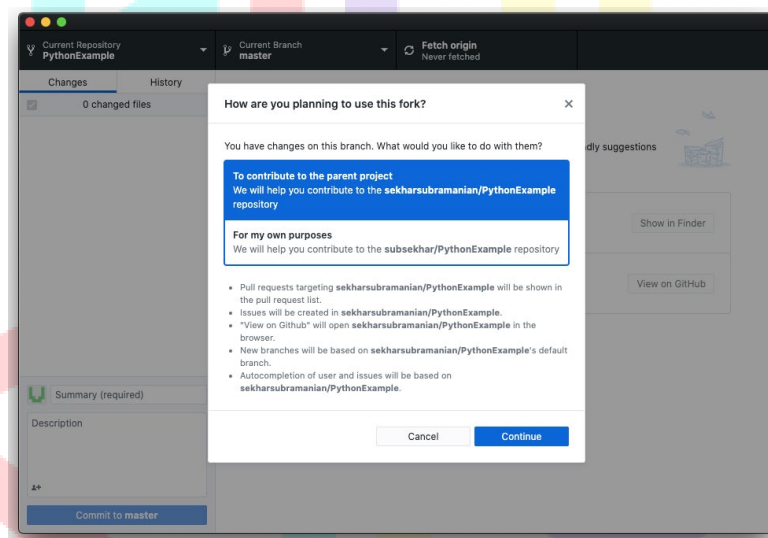
Authenticate on GitHub desktop using GitHub credentials

- ◆ After authentication, all repositories affiliated with the account will be displayed. Click the forked repository to clone it into the local disk and provide the file path where the repository will be cloned



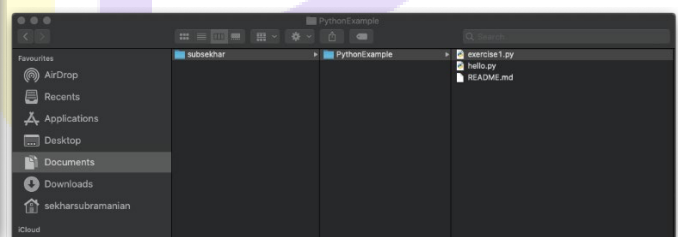
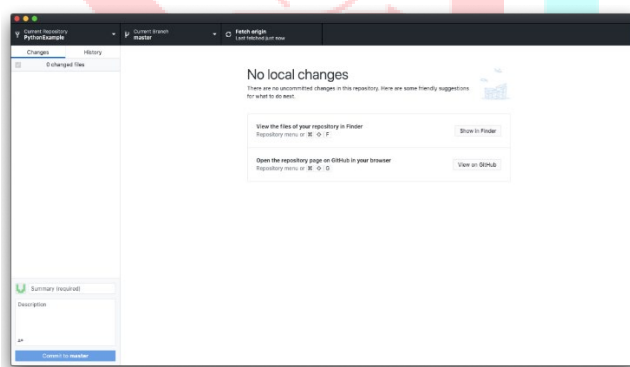
Choose the forked repository and provide the local path to clone into

- ◆ Select **For my own purposes** in the prompt that comes up when cloning the repository. This indicates that any changes made will be finally pushed to the remote repository and not the upstream repository.



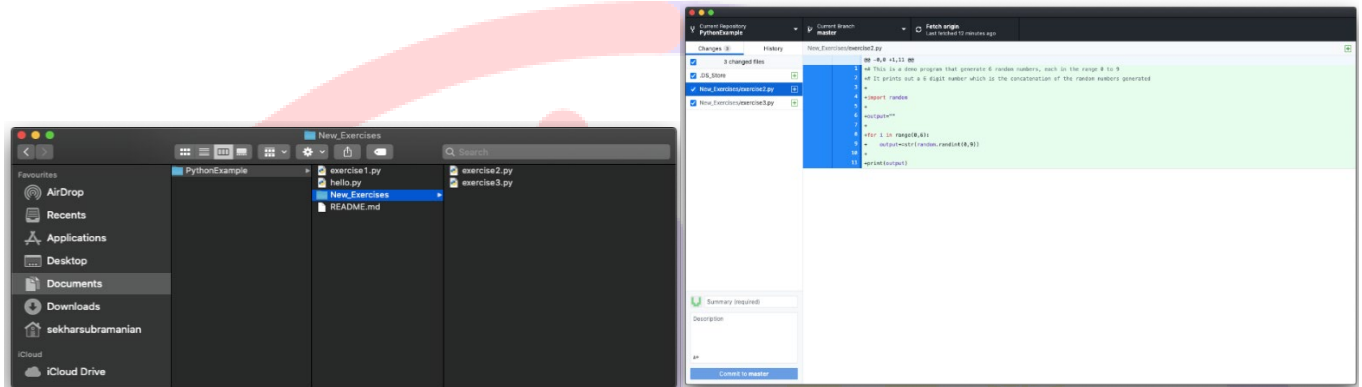
Select how the fork will be used

- ◆ Observe the details of **current repository**, **branches** and **sync status** with the remote repository on GitHub desktop. Note that the software doesn't show the files in the repository. To view that select **Show in Finder / Windows Explorer**.



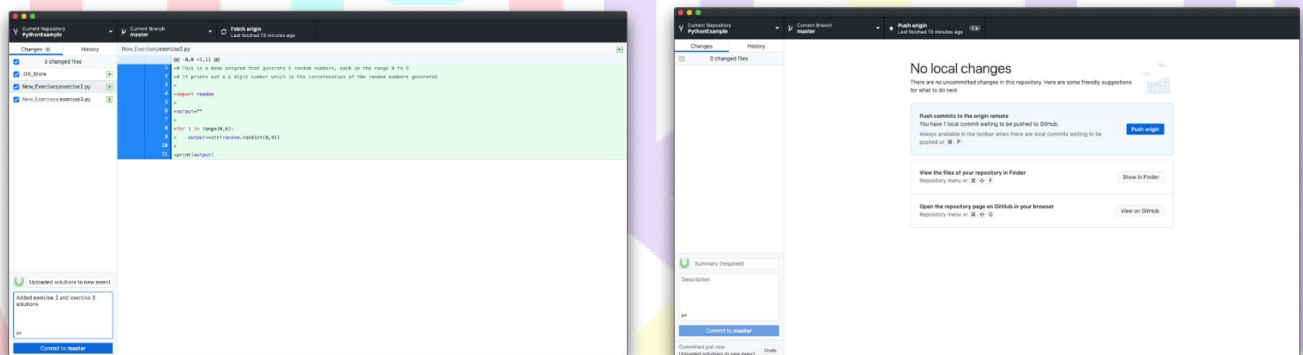
GitHub desktop after cloning a remote repository and File explorer showing files from remote repository

- ◆ The files in the local disk of the computer will be the same version as those in the remote repository. This is called as the local repository. Any changes made to the folders and files on the local disk implies making changes to the local repository.
- ◆ Observe that when new files are added to the workspace, GitHub Desktop displays these changes and allows that the changes can be committed to the repository.



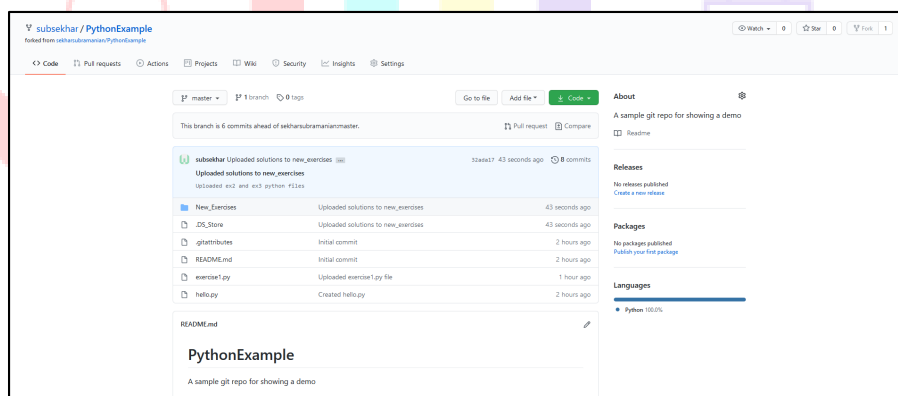
Updated folder in local disk and changes displayed in GitHub Desktop

- ◆ Commit the changes to the local repository using the form available in the bottom-left corner of GitHub Desktop by clicking **Commit to master**. After the commit happens, observe that there are no changes in the local repository and changes can be pushed to the remote using **Push origin**



Commit changes to local repository and observe that changes can be pushed to remote repository using Push Origin

- ◆ Click **Push Origin** and observe that the changes are pushed from local repository to remote repository. The GitHub page displays the updated repository.



Synchronized remote repository with all files uploaded

Other ways to work with Git

In addition to GitHub web interface or the GitHub Desktop application, here are some popular approaches to managing the repository:

◆ Using Git with command line/terminal

- This is perhaps the most popular and common way that git is used by developers across the globe.
- While there is extensive documentation on the usage of the commands, refer [this cheatsheet](#) to get a quick overview of the key commands that can be used to manage a repository.

◆ Using Git with IDEs

- Most modern IDEs like Spring Tool Suite, Visual Studio Code, Atome, etc. ship with plugins for git.
- The installation, configuration and usage of these tools is fairly straightforward and offers an advantage of integrating the development and version control environment
- Refer the documentation of the IDE that is of interest to get more information.

Summary: Making changes to the repository

In this section, we have added files to the remote repository using the web interface and also explored how to clone the remote repository into the local disk to create a local repository.


We have also made changes to the files on local disk, committed the changes to the local repository, and pushed the updated local repository to the remote repository there by publishing the update onto GitHub.

Synchronizing between original (upstream) and forked (remote) repositories


Updating remote repository (of graduate) to reflect changes made to upstream repository (of trainer)

In the event that the trainer updates their repository to include new files such as demo programs, assignments, presentations etc, the forked repository will not automatically be in sync. The owner of the repository (graduate) has to explicitly synchronize to get the updated files into their repository.

◆ Observe for changes in the upstream repository using the following techniques:

- Watch the upstream repository by clicking the Watch icon  in the web interface. This populates your GitHub main feed with any updates that are made to the repository that you are watching
- Observe the comparison status on the main page of the forked repository.

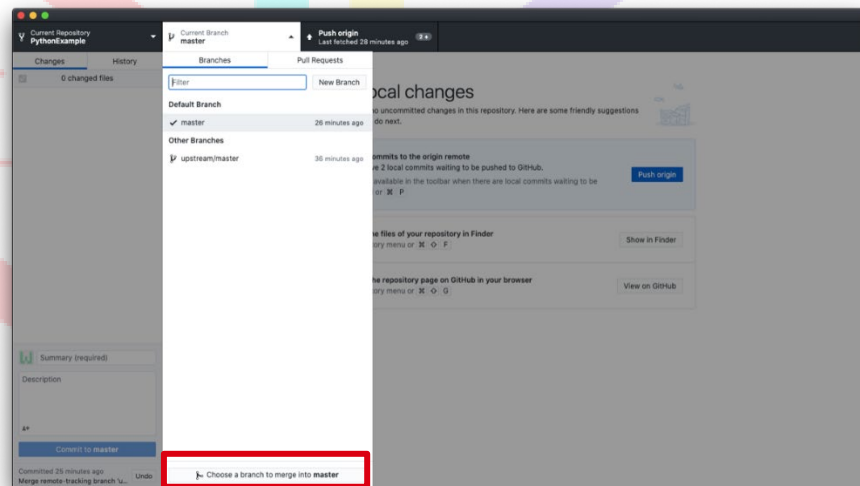
This branch is 7 commits ahead, 2 commits behind sekharsubramanian:master.

 Pull request  Compare

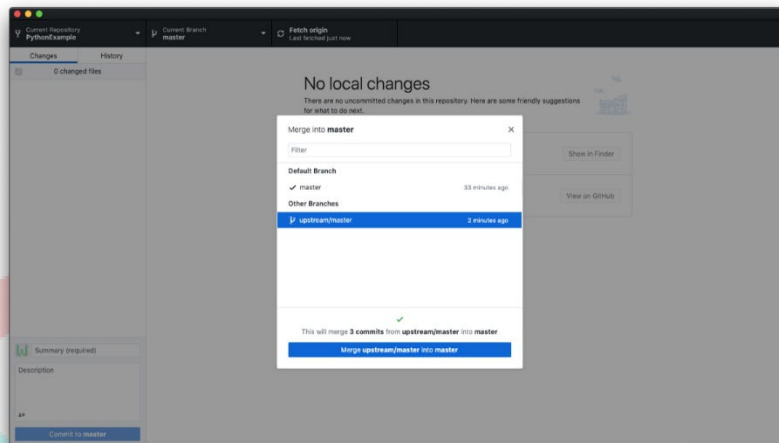
Here 7 commits ahead means that the forked repository has made 7 new commits that are not part of the upstream repository; likewise 2 commits behind implies that there are 2 commits that happened on the upstream repository that are not part of the forked repository

◆ Use GitHub Desktop to merge commits from the upstream repository to the remote repository

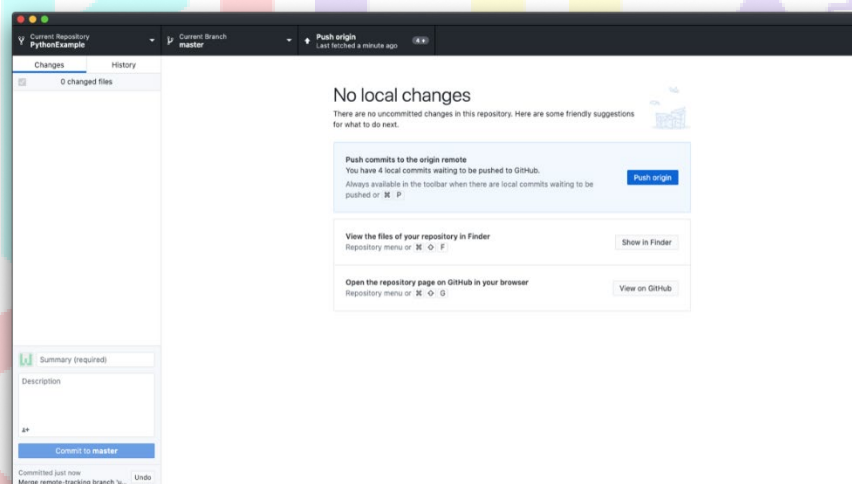
- In the **Current Branch** pane on GitHub desktop, click **Choose a branch to merge into master** button which appears at the bottom.



- Select **upstream/master** in the popup dialog and **Click Merge upstream/master into master**. This downloads all the new files from the upstream repository into the local repository on the computer



- After the merge is completed, the local repository is up-to-date with the upstream repository. To push the changes to the remote repository select **Push origin**. Once the push is completed observe the GitHub web for the new files inherited from the upstream repository



- To do this using command line follow the steps mentioned [here](#)

Updating upstream repository (of trainer) to reflect changes made to remote repository (of graduate)

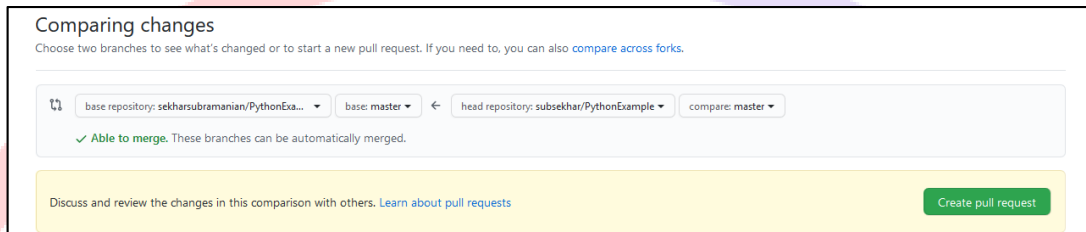
In the event where a trainer or the Technology Academy team sets up an upstream repository and asks the graduates to share their assignment/assessment solutions then the following steps can be followed:

- ◆ Fork the upstream repository provided to create a remote repository
- ◆ Clone the remote repository onto your local hard disk thus creating a local repository
- ◆ Follow any instruction pertinent to changes that need to be made to the local repository. For example, graduate may need to create a folder that can uniquely identify their work. They may need to make changes only in that folder such as pasting the assignment/assessment files into it.
- ◆ Commit the changes to the local repository

- ◆ Push the changes to the remote repository
- ◆ Create a pull request to merge the remote repository with the upstream using the following steps:
 - Click on Pull Requests in the Web UI



- Check if there are any merge conflicts as GitHub compares and displays them in the web interface



- Click **Create pull request**, fill out the form presented and click on the '**Create pull request**' button to submit the request.
- The upstream repository owner can view pull requests, share feedback, and start discussions. They can accept the pull request if it passes all checks. When the pull request is accepted, then your code becomes a part of their repository and your name gets added to contributors of the repository.

Summary: Synchronizing between original (upstream) and forked (remote) repositories

In this section, we have learnt how to synchronize between the upstream repository and the remote forked repository. We have seen how to download changes made to the original repository and sync it to our version and also seen how to upload changes we make to the original version.

Overall, we have explored the specific use case of how we can use Git for the Graduate training. Git is a very powerful version control management tool that is used across the industry. Go through [this link](#) to interactively understand the various concepts and commands of Git.