In this video, you will learn how to create and

merge a branch using the GitHub web interface.

A branch is a snapshot of your repository to which you can make changes.

It is a copy of the master branch and can be used to develop and test changes to the

workflow before merging it back to the master branch. In Git and GitHub, there is a main

branch.

The main branch which is called Master, is the one with deployable code

and the official working version of your project.

It is meant to be stable and it is always advisable never to push any code that is not

tested to master.

Many times, we want to make changes to the code and workflow in the master branch.

That is when we create a copy of the Master branch.

Let’s call it Child Branch.

We will then create a copy of the workflow to the child branch in the child branch, changes

and experiments are done.

We will build and make edits, test the changes and when we are satisfied with the changes

we will merge it back to the master branch where we prepare the model for deployment.

We can see that all of this is done outside of the main branch and until we merge, changes

will not be made to the workflow before we branched.

To ensure that changes done by one member, does not impede or affect the flow of work

of other members, multiple branches can be created and merged appropriately to master

after the workflow is properly tested and approved.

To create branches in GitHub, let’s look at this repository.

There is currently one branch in the repository.

I want to make some changes, but I don’t want to alter the master in case something

goes wrong.

We will create a branch.

To do that, we will click the drop-down arrow and create a new branch.

Let's name it - child branch and then we will click enter.

The repository now has two branches, the Master and the Child branch.

You can check this by selecting Child branch in the Branch selector drop-down list.

Whatever was in the Master branch was copied to the child branch.

But we can add files in the child branch without adding any files to the master branch.

To add a file, make sure Child branch is selected in the branch selector drop-down list.

Click on create new file. In the space provided, name the file - we will name it

testchild.py and then we will add a few lines of code.

We will print the statement – Inside child branch.

At the bottom of the screen, we will see a section called Commit new file.

Commit messages are very important as it helps to keep track of the changes that were made.

It is important to add a descriptive commit message so that other team

members can understand it.

Here we will add a commit message,

Create testchild.py, then we will commit the new file.

The file gets added to only the child branch.

We can check this by going to the master branch by clicking ‘master’ from the Branch selector

menu and here we can see that the new file is not added to the master branch.

After we have created the new file, tested and made sure that is up to standards.

We then want to merge the changes in the child branch to reflect in the master branch.

To merge the changes, we will first have to create a pull request, also known as a PR.

A pull request in simple terms is a way to notify other team members of your changes

and edits and ask them for review so they can be pulled or merged into the master branch.

Pull requests are the heart of collaboration on GitHub.

When you open a pull request, you’re proposing your changes and requesting that someone review

and pull in your contribution and merge them into the target branch.

Pull requests show the differences of the content from both branches.

To open a pull request and see the differences between the branches, click on the Compare

and pull request button.

If you scroll down to the bottom of the screen, you will see something like this that shows

you the difference between both branches.

As you can see on the screen it shows that one file has changed and the file has two

additions, which are the two lines we added to the file and 0 deletions.

We will now create the pull request.

Add the title and an optional comment for the pull request.

Click Create Pull request to create the pull request.

You can assign team members to review and approve pull requests.

On the next page you will see this image. If you are okay with the changes, click on Merge

pull request and click confirm. You will get a confirmation that the pull request has been

successfully merged.

You can now delete the branch if you no longer need to make any edits or add new information.

Now, the child branch has completely merged with the Master branch.

You can check the Master branch and we can now see it contains the

testchild.py file.

You should now be familiar with how to create and merge branches using the web interface.