Remote vs Local

In the pre-internet era, saving project files to different machines for

backup and transfer was a tedious process.

It required manually copying files between machines, one at a time.

Making things slow for teams.

Nowadays the cloud has enabled a more efficient way to do this.

And in this video I'll explain the differences between remote and

local on Git hub.

You have previously learned about the flows modified, staged and

committed in a Git workflow.

Now you will learn about pushing your changes from your

local to a remote repository.

Remote refers to any other remote repository to

which developers can push changes.

This can be a centralized repository, such as one provided by Git hub or

repositories on other developer devices.

In this lesson, you will be hearing some new terms such as clone,

push, pull and repo.

Don't worry.

These will all be explained soon.

The remote code is accessed through a URI which is unique and

only accessible to those who have permission local.

On the other hand refers to your machine which can be a laptop, desktop or

even a mobile device and

is only accessible to you to demonstrate both of these in action.

Let's say we have a project called coding project one which is located on

Git hub with a unique URL.

In other words, this project is stored on a remote server.

When a user wants to copy this project to their local device, they need to either

perform a clone if it's the first time or pull it to get the latest changes.

To clone a project a user must first choose a folder on their local machine.

Coding project one is then cloned from the server and copied into the chosen folder.

The user can then make changes to the project and

push those changes back to the server.

Other users working on the code base won't see those changes on their local machines

unless they pull the latest changes from the server.

One of the advantages of it is that you can work offline and

then commit your changes when you are ready.

Now let's go through an example of how exactly you would do this and Git hub.

In this video I'm going to explain what local and remote mean in Git hub and

help you to understand the differences between the two.

First off, I'm going to create a new local repository using the Git in IT command.

I'll start by inputting M K D I R to create A new directory and

then I'll set the name as my second repo.

Next I'll use the change directory command which is CD followed by the name

that I just said.

Finally, I'll perform the Git in IT command to create my new repository.

This will return a line telling me that an empty

repository has been initialized under route projects my second repo.

If I execute another command called git remote, it comes back as blank.

The reason for that is that I've only initialized a local repository which has

no connection to a central repository that sits either on Git hub or another server.

Right now it's only available locally on my machine.

Now I'll step back out from this directory and

go into my first repo with a CD command.

Again, this is a repository that I created earlier and

does have a connection to Git hub.

Using the remote URI I'll then check it by using the get remote minus V command.

Git tells me that it's set to git tutorials 101 my first repo.Git.

Next I'm going to set this against her second repository.

I'll step into the new directory once more using the CD command.

In this case we're going to add this URL to the remote settings by using

the command.

Git remote add specifying a name and then passing in URI,

the name that is typically used here is origin.

So I'll stick with that.

So again the full command with the URL should read Git remote at origin.

Git at Github.com:

Git tutorials 101/

My first repo.Git.

This time when I execute the Git remote minus V command.

I have this set up against the get tutorials 101 which sits up on Git hub.

What I'm going to do next is use the Git pull command which will connect with

the GIT hub server, and pull down all the changes from the repository.

So on my local I now have all the changes, but when I check the directory it's blank.

The reason for this is that I haven't set up a branch that matches with what I

have on the server repository.

Fortunately I can change that by performing the command.

Git check out main.

Which will set up a branch main on my local that tracks the branch main from

the remote.

And now when I check my folder using the LS command,

it confirms that I have the read me test and test two files available on my local.

In this video, you learned about the differences between local and

remote and Git hub.

This will help set you up to exchange data more efficiently within your

development team.

See you next time.

You have created a new local repository called “test-repo”. Which command would you use to set the remote repository in git?



Git clone



Git pull



Git push



Git remote