By now, you should know how to use git add,

and git commit to add

new changes to your local repository.

Put them into the staging area,

and get them ready for a commit.

Now, let me guide you through the next step in uploading

these changes to the remote repository using git push.

I'll also demonstrate how to retrieve changes from

the remote servers and apply them

to your local repository with git pull.

Before we begin, let's go over

the command line and perform the command git status.

Git tells me that I'm on the branch, main,

but also the my branch is ahead

of the origin main branch by one commit.

What this means is that all the changes that I

have on my local repository are currently

ahead of what is stored in

the remote repository on GitHub that ties into

Git's distributed workflow in which you can

work in an offline state and then only ever

communicate with a remote repository when you

use the commands of git push or git pull.

Now I'll guide you on pushing

your changes to the remote repository,

and then I'll demonstrate how to use

the pull command to get the latest changes.

It's always good practice to

check which branch you're currently on.

You can use git status or git branch to do this.

This is important because if you do

make changes in a different branch,

you need to specify where you're pushing up to.

Let's push up the changes using the git push command.

I'll specify the origin branch to be the main branch.

As in, I'm pushing my changes to

the origin as the remote repository,

and then I want to push it to this branch as the main.

I'll be prompted for my login information

as I am pushing using HTTPS.

Once I enter my login information,

you'll notice that the commit is pushed

from the local main to the remote main,

on the remote repository.

Let's refresh the page on the GitHub website.

You can see that my test.txt file now appears there.

That's taken the commit snapshots that I have in

my local repository and pushed

it up to the remote repository.

Git has then compared those files with what's on

the remote repository to find any conflicts or problems.

If none are found,

it'll just merge them straight through,

which is classed as an auto merge.

If there are any conflicts,

my push will fail.

Before doing a push,

it's also good practice to

perform a git pull in order to get

the latest changes from

the remote repository and reduce

the odds of encountering a conflict.

Because I only have a single file

and this will be a new repository,

I'm not going to run into any conflict or any issues.

Now let's move on and I will

guide you through how you can use git pull.

Normally when you're working on a project,

you could have several developers all

submitting with different branches,

different code, and different features.

In order for you to get those changes,

you need to use the git pull command.

To demonstrate this, I will add

a single line to the test.txt file,

using the GitHub UI and then add the climate change,

updated the test.txt file.

I'll then commit it directly to the main branch,

by clicking on Commit Changes.

The changes now appear on the UI.

But because I haven't used the git pull

command on my local machine yet,

I should have no content of the test.txt file.

Let's verify by using the cut command on test.txt.

Sure enough, the file is empty,

which is what you'd expect.

As I mentioned before,

I need to run the command git pull.

This will get the latest changes

from the remote repository.

If any new changes were added,

it'll be reflected in the shell output.

I run the command,

and in this case,

Git tells me that one file

has changed with one insertion.

If I run the cut command on test.txt once more,

it shows that the line "this is my

change" is now available in my local directory.

With git pull,

you're taking all that data from the remote server.

Git then merges the snapshot from the

remote with the snapshot that's on your local.

It will auto merge them if there is no conflicts.

Once that's complete, I'll have

the latest changes on my local machine.

You have now learned how to push to

and pull from your GitHub repository.

When you perform a Git Push command, it copies content from your local repository and then uses it to merge all of the content in the remote repository.



False



True

Incorrect

Not quite. When you use Git Push, Git compares a snapshot of your local repository with the remote one and only replaces the files that have been changed.

Correct

That's correct! When you use Git Push, Git compares a snapshot of your local repository with the remote one and only replaces the files that have been changed.