So I've opened up my terminal window.

I'll have a look at what directory I'm currently located in.

I can do this by running the pwd command which is short for

print working directory.

Notice that I'm in the directory my-first-repo.

Now I can check inside that directory by running the ls- la command.

I can see that I have two items, a README.md file and

a hidden folder called .git.

Before I add any files or make any changes,

it's always good practice to check if any changes or commits are currently there.

I can do this by using the git status command.

Git status also displays what branch I'm on.

In this instance, I'm prompted that I'm on the branch called main and

that my branch is up to date with the origin main.

This means that all the latest files on my local machine are exactly the same as

what is displayed on the GitHub UI,

which represents the server that everyone commits to.

Git status also tells me that I have nothing to commit and

that my working tree is clean.

Now, let me show you how to add a simple text file.

I'll add a file called test.txt by using the command touch test.txt.

Then I'll run the command gitt status again.

Now git is telling me that I have an untracked file which is the test.txt

file that I just added.

It's also telling me that I have nothing added to the commit but

that untracked files are present and that I should use git add to track them.

The purpose of the git add command is that I'm essentially prompting git and

letting it know that I want to track this file, and

that it will be included as part of my commit.

The first phase of this process is just to run the command git add test.txt.

Now I'm going to run git status again to check that file is now being tracked.

Notice again, that I'm notified that my branches are up to date.

But it's also telling me now that there are staged changes to be committed,

which is this new file called test.txt.

It prompts me asking if I want to revert those changes.

For this, I can use the git restore command with the flag-- stage and

the file name test.txt.

Running the command will unstage the file from the commit.

I then run git status one more time to see the file is back to an untracked status.

So once again I'll add the file using git add test.txt.

Run git status, and now notice that the file is back in a tracked state.

Okay, let me clear my screen before moving on.

To do this, I use the clear command.

Now any changes that I make from here on will be tracked and

then at the end I will use the git commit command.

The staged area is really important because you're essentially preparing to

get all of the files and

changes that you want as part of whatever feature you're working on.

Basically, you are getting all of that content ready for commit.

You also have to remember that this is only on your local machine.

The distributed manner of git means it will only push to

the server using the actual push command itself.

But any change you make here is only specific to you and your local machine.

Anyone else who pulls down the project from GitHub will only get what's available

on the remote server.

Okay, now I want to explain to you how to run the git commit command.

First, type in git commit, you can pass in a flag of -m which stands for

message, allowing you to type in a message which will be attached to the commit.

In this example, the message is adding a new file for testing.

Next, press Return on the keyboard and now notice that the response

states 11 file change 0 insertions, 0 deletions.

There is also a create mode statement with the name of the file test.txt.

Finally, if I run the git status command,

the response says that there is nothing to commit and the working tree is clean.

However, I want to be aware of the message at the top of my screen.

This message tells me to use git pushed to publish my local commits and

this ties back in with what I mentioned earlier.

All of these changes are on my local machine,

and they will only be uploaded to the remote server when I run the push command.

You'll learn more about the push and pull commands in a future lesson.

Command

Git status is used for looking in which branch we have

cd my-first-repo-ai

pwd

ls -la

git status

touch test.txt

git status : to stage area

git add test.txt

git restore --stage test. :to unsstatus

git commit -m "Adding a new Files for Testing "

-m : mean message

[main 283049b] Adding a new Files for Testing

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 test.txt

You've learned that Git works on the principles of the edit, stage, and commit pattern. But when you edit a file in your Git working directory it will not be tracked until it's added. You can do this by using the Git commit command.



False



True

Incorrect

Not quite. You use the Git add command to tell Git that the file is staged to be committed and that any further changes to the file will be tracked going forward.

That's correct! You use the Git add command to tell Git that the file is staged to be committed and that any further changes to the file will be tracked going forwar