1) Create the Repository
You should have already created a GitHub account in the Setting up Git lesson.
If you haven't done that yet, you can sign up here.

2) Create a new repository by clicking the button shown in the screenshot below.

3) Give your repository the name "git_test" in the repository name input field, and
create the repository by clicking the green "Create repository" button at the bottom of the page.

4)His will redirect you to your new repository on GitHub. To copy this repository onto your local machine,
select the SSH option and copy the line next to it.

5)
In the command line on your local machine,
navigate to where you want to store this project, and then clone your repository on GitHub onto your
computer with git clone followed by the URL you copied in the last step. The full command should look similar to
git clone git@github.com:USER-NAME/REPOSITORY-NAME.git.

6)
That's it! You have successfully connected the repository you created on
GitHub to your local machine. To test this, you can cd into the new git_test folder
that was downloaded and then enter git remote -v in your command line. This will display the URL of the repositor
y you created in GitHub,
which is the remote for your local copy. You may have also noticed the word origin at the start of the git remote -v
output,
which is the name of your remote connection. The name "origin" is both the default and the convention for the rem
ote
repository, but it could have just as easily been named "party-parrot" or "dancing-banana".
(Don't worry about the details of origin for now; it will come up again near the end of this tutorial.)

1)

Use the Git Workflow
Create a new file in the git_test folder called "README.md" with the command touch README.md.

2)

Type git status in your terminal.
In the output, notice that your README.md file is shown in red, which means that this file is not staged.

3)

Type git add README.md. This command adds your README.md file to the staging area in Git.
Now, type git status again. In the output, notice that your file is now shown in green, which means that this file is n
ow in the staging area.

4)

Type git commit -m "Add README.md" and then type git status once more. The output should now say, “nothing to commit”, indicating that your changes have been committed.

5)

Type git log and look at the output.
You should see an entry for your “Add README.md” commit.
You will also see details on the author who made the commit and the date and time for when the commit was made.

1)

Add Another File
Create a new file in the git_test folder called hello_world.txt. In the terminal, type git status, and notice hello_world.txt is not staged.

2)

Open README.md in your text editor of choice and add the text “This is (YourUsername)’s first git project!” and then save the file.

3)

Back in your terminal, type git status, and notice that README.md is now shown as modified, and not staged or committed.
This is because you made a change to it, and it is already a tracked file.

4)

Add README.md to the staging area with git add README.md.

5)

Can you guess what git status will output now?
README.md will be displayed in green text, while hello_world.txt will still be in red.
This means that only README.md has been added to the staging area.

6)

Now, add hello_world.txt to the staging area with a slightly different command: git add .,
where the full stop means to add all files in the current directory that are not staged.
Then, type git status once more, and everything should now be in the staging area.
(Note: You can use git add -A to add ALL unstaged files to the staging area within the repository)

7)

Finally, let's commit all of the files that are in the staging area and add a descriptive commit message git commit -m "Add hello_world.txt and
edit README.md". Then, type git status once again, which will output "nothing to commit".

8)

Take one last look at your commit history by typing git log. You should now see two entries.

1)

Push Your Work to GitHub
Finally, let's upload your work to the GitHub repository you created at the start of this tutorial.
Type git push origin master.

2)

Type git status one final time. It should output "nothing to commit, working tree clean".

3)

When you reload the repository on GitHub, you should see the README.md and hello_world.txt files that you just pushed there from your local machine.

New topic on git

The seven rules of a great Git commit message
Keep in mind: This has all been said before.

1. Separate subject from body with a blank line
2. Limit the subject line to 50 characters
3. Capitalize the subject line
4. Do not end the subject line with a period
5. Use the imperative mood in the subject line
6. Wrap the body at 72 characters
7. Use the body to explain what and why vs. how