

How to Set Up Your Environment for a New or Existing Git Repository

A brand-new repo.

1. Set up the new repo directly in the appropriate GitHub account with Private, .gitignore, README, license etc., as you please.
2. Once finished, click Code -> Clone -> https and copy the URL string to your clipboard.
3. On your local system, in a command line processor, Git Bash, for example, navigate the current directory to the parent of where this new repo shall live.
4. Create a new directory same name as the repository you just created on GitHub with:

```
$ mkdir [new directory name]
```

5. Execute the following commands.

```
$ cd [new directory name]
```

```
$ git init
```

The 'git init' command turns an existing directory into a new Git repository inside the folder you are running this command. After using the git init command, link the local repository to an empty GitHub repository using:

```
$ git remote add origin [the URL you still have on your clipboard]
```

Now set the remote branch to your current branch. Assumes you haven't done any branching...

```
$ git config --global push.default current
```

Now, download all the files in the repo to your local working directory.

```
$ git pull [the URL you still have on your clipboard]
```

Alternatively, say you're new to the repo, and you want set up your local machine with the latest updates to download into your local git directory, execute the following commands:

1. On your local system, in a command line processor, Git Bash, for example, navigate the current directory to the parent of where this new repo shall live. Then:

```
$ git clone [the URL you still have on your clipboard] [your new directory name]
```

This will create a new directory named [your new directory name], and download the entire working repo, with history, into this directory. As you can see, this wins for brevity...

To be sure, as all things git, there are other ways to accomplish this.