

DMIT 1530

Week 3

Git

Other than an affectionate insult,
what is **Git**, anyway?

Git is a type of **version control system**.

... sounds fancy, right? All that means is it keeps track of **changes** that you make to files and folders.

This makes it possible to work on the same project with other people and not worry about file conflicts.

However, you have to tell Git:

1. to start **tracking changes**
2. **which files** you want it to keep track of
3. when you want to save a new **version** of your project.

But first, we have to make sure that Git's **installed**.

Installing Git

Sorry, Windows users.

Linux

(incl. Ubuntu, Fedora, Solaris, etc.)

If you're using a **Linux** distro, you most likely have some form of Git installed under **usr/local/bin**.

(The bloke who created Linux also created Git, so ...)

For the **latest version**, copy and paste the provided snippets into your command line.

<https://git-scm.com/download/linux>

... and that's it. You're good to go, champ.

macOS
(incl. OS X)

If you're using any version of macOS released in the past ten years, you already have Git installed.

If you are using an older version of macOS or OS X, you will need a package manager (a thing that helps you install and update stuff), like [Homebrew](#).

Open up **Terminal** (under Applications > Utilities) and follow the installation instructions on their homepage.

<https://brew.sh/>

After you have Homebrew installed, **run the commands** provided on Git's installation page.

<https://git-scm.com/download/mac>

Once you've completed your installation, you can check to see whether it was successful.

In Terminal, type `git --version` and hit enter.

If the prompt prints your version number, you successfully installed Git and can begin using it.

If you receive an error, or your version number does not match the current one, you will need to install Git again.

Windows
(incl. 32-bit & 64-bit systems)

If you use **Windows**, you will need to install Git.

Fortunately, there is an **installation wizard** available for you.

<https://git-scm.com/download/win>

As you work through the installer, make sure that you choose:

1. Override the default branch name for new repositories. Type `'main'` as your new default.
2. Git from the command line and also from 3rd-party software.

Once you've completed your installation, you can check to see whether it was successful.

Open the Command Prompt by typing

1. Ⓜ Win + R
2. cmd
3. Enter

Next, type `git --version` and hit enter.

If the prompt prints your version number, you successfully installed Git and can begin using it.

If you receive an error, you will need to install Git again.

Configuring Git

Not sorry, Windows users.

Before we start using Git, we need to change some of our **configuration settings**.

You can make these changes by using the three commands on the following slide in **Command Prompt** (Windows) or **Terminal** (macOS).

Note: make sure to change any green text to your own information.

```
git config --global user.name "First Last"
```

Sets your name.

```
git config --global user.email "hi@email.ca"
```

Sets your email address.

```
git config --global init.defaultBranch main
```

Changes the name of your default branch to 'main'. This means that 'main' will be used in all of the following commands.

GitHub Desktop

There's an app for that.

The rest of our resources will cover the overall workflow and how to use Git from the **command line**.

However, you may also use [GitHub Desktop](#), an app that executes Git commands through a more user-friendly GUI.

You can use it to control your entire project from start to finish, or in conjunction with your command line.

GitHub Desktop

<https://desktop.github.com/>

Further Reading & Resources

Git Documentation

<https://git-scm.com/doc>

GitHub

<https://github.com/>