## **INTRODUCTION TO GIT**

### **Introduction**

Git is like a really epic save button for your files and directories. Officially, Git is a version control system.

A *save* in a text editor records all of the words in a document as a single file. You are only ever given one record of the file, such as essay.doc, unless you make duplicate copies (which is difficult to remember to do and keep track of):

essay-draft1.doc, essay-draft2.doc, essay-final.doc

However, a *save* in Git records differences in the files and folders AND keeps a historical record of each save. This feature is a game changer. As an individual developer, Git enables you to review how your project grows and to easily look at or restore file states from the past. Once connected to a network, Git allows you to push your project to GitHub for sharing and collaborating with other developers.

While Git works on your *local* machine, GitHub is a *remote* storage facility on the web for all your coding projects. This means that by learning Git, you will get to showcase your portfolio on GitHub! This is really important because almost all software development companies consider using Git to be an essential skill for modern web developers. Having a GitHub portfolio will provide proof to future potential employers as to what you are capable of.

In this lesson, we will briefly explore the history of Git, what it is, and what it’s useful for.

In the next lesson, we will go over the basic workflow for using Git, which should enhance your understanding and demonstrate why Git is so useful.

Finally, you will set up a project with Git that will serve as a template for your future projects.

For now, let’s learn what Git is and why it’s so powerful!

### **Learning Outcomes:**

By the end of this lesson, you should be able to do the following:

* Explain what Git and GitHub are and the differences between the two.
* Describe the differences between Git and a text editor in terms of what they save and their record keeping.
* Describe why Git is useful for an individual developer and a team of developers.

### **Assignment**

1. Read Chapter 1.1 through 1.4 in [this book about version control](https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control) to learn the differences between local, centralized, and distributed version control systems.
2. Watch [this video](https://www.youtube.com/watch?v=8oRjP8yj2Wo) about how Git can improve the workflow of both an individual and a team of developers.
3. Watch [this video](https://www.youtube.com/watch?v=1h9_cB9mPT8&feature=youtu.be&t=13s) for some history on Git and GitHub, and make sure you know the difference between the two. Git is a technology used in the command line while GitHub is a [website](https://github.com/) you can visit.
4. If you haven’t yet installed Git, visit the Setting Up Git lesson.

### **Additional Resources**

This section contains helpful links to other content. It isn’t required, so consider it supplemental if you need to dive deeper into something.

* [Git and GitHub in plain English](https://blog.red-badger.com/blog/2016/11/29/gitgithub-in-plain-english)

### **Knowledge Check**

This section contains questions for you to check your understanding of this lesson. If you’re having trouble answering the questions below on your own, review the material above to find the answer.

* What kind of program is Git?
* What are the differences between Git and a text editor in terms of what they save and their record keeping?
* Does Git work at a local or remote level?
* Does GitHub work at a local or remote level?
* Why is Git useful for an individual developer?
* Why are Git and GitHub useful for a team of developers?