

Login

Search for



Apply

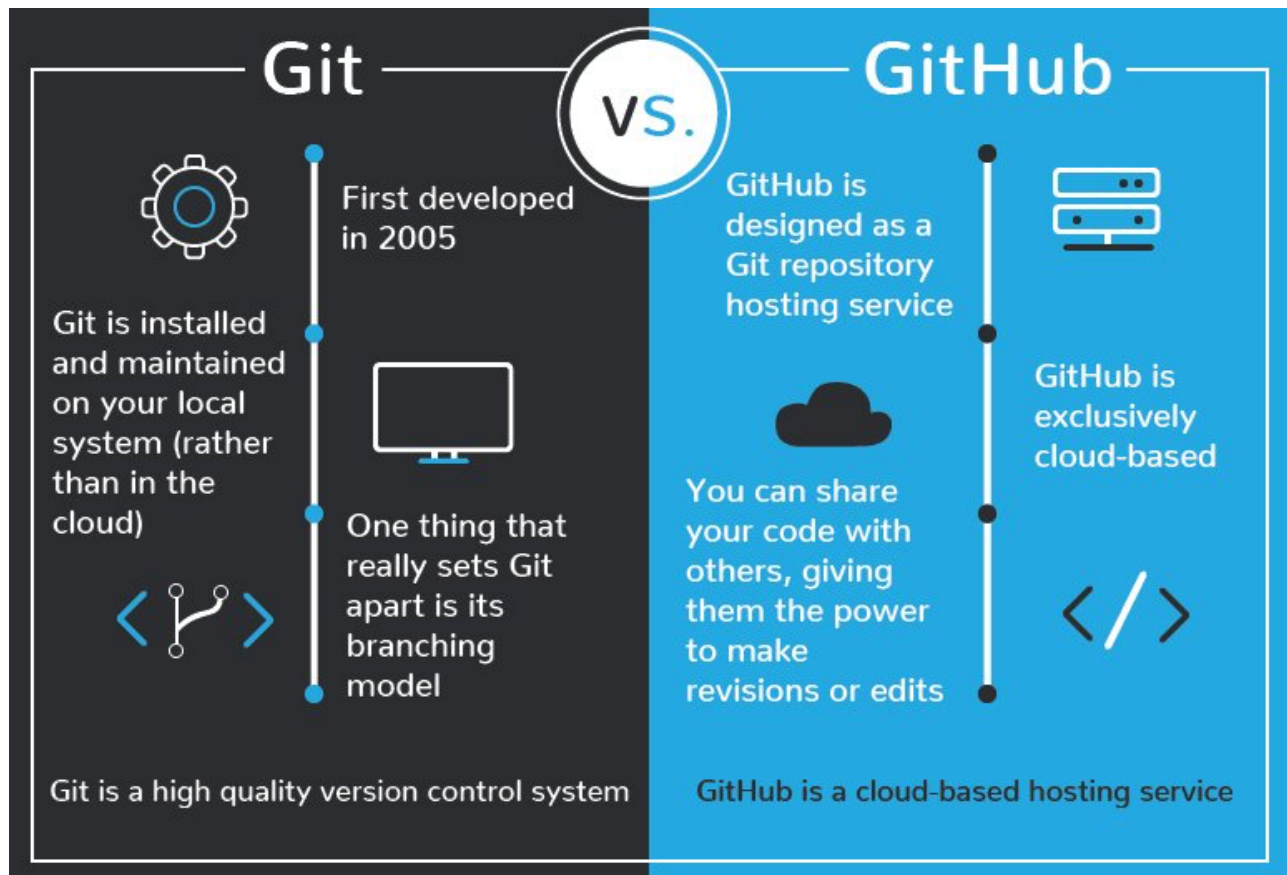
Git vs. GitHub: What's the Difference?

Git vs. GitHub: What's the Difference?

Given that coding relies so heavily on exact syntax, the naming convention surrounding programming languages and resources is anything but intuitive. Java and JavaScript are about as closely related as *ham* and *hamster*, and Python's logo may be an image of intertwined snakes, but it's actually named after the Monty Python sketch comedy group. HTML and CSS are acronyms that describe what the code actually is or does (*HyperText Markup Language* and *Cascading Style Sheets*, respectively), while C++ describes its origins. And that's only scratching the surface.

So, for someone first hearing about *Git* vs. *GitHub*, the apparent connection may not be quite so apparent.

Git vs. GitHub



Git or GitHub. Are they the same thing? If not, are they connected in some way?

Or, like Java and JavaScript, is the connection only superficial?

These questions are definitely worth asking. After all, Microsoft was willing to shell out [\\$7.5 billion dollars](#) to acquire GitHub back in 2018, so developers of all skill levels should be sitting up and taking notice. But the truth is that Git and GitHub are connected much more closely than Java and JavaScript — but with some key differences setting them apart.

What's the difference between Git and GitHub? Well, to answer that, we'll be taking a closer look at each one. But before we do that, let's first discuss the concept of *version control*.

Version Control

Version Control

is like a save program for your project. By tracking and logging the changes you make to your file or file sets over time, a version-control system gives you the power to review or even restore earlier versions.

Development projects don't come into existence *ex nihilo*; they're built code line upon code line from the ground up. And often, there's a lot of trial, error, and correction that goes into creating something that actually works the way it's supposed to. That's where version control comes in.

Version control is like a savings program for your project. By tracking and logging the changes you make to your file or file sets over time, a version-control system

gives you the power to review or even restore earlier versions. Version control takes snapshots of every revision to your project. You can then access these versions to compare or restore them as needed.

For example, let's say that you're working on a web development project, and through the course of your revisions, you suddenly notice all of your text has become misaligned. And because the first rule of coding is that it's *always* your fault, you can bet that some change you made somewhere along the way is the cause. But not to worry; instead of having to crawl back through every line of code, you can just use your version control system to reload earlier versions, until you pinpoint the offending change and correct it.

And with that in mind, let's move on to Git.

What Is Git?

First developed back in 2005, Git is an extremely popular version control system that is at the heart of a wide variety of high-profile projects. Git is installed and maintained on your local system (rather than in the cloud) and gives you a self-contained record of your ongoing programming versions. It can be used completely exclusive of any cloud-hosting service — you don't even need internet access, except to download it.

Compared to other version control systems, Git is responsive, easy to use, and inexpensive (*free*, actually). Git is also specially designed to work well with text files — which, if you think about it, is what code actually is. But one thing that really sets Git apart is its *branching* model. Branching allows you to create independent local branches in your code. This means you can try out new ideas, set aside branches for production work, jump back to earlier branches, and easily delete, merge, and recall branches at the click of a button.

And that's it. Git is a high-quality version control system. But what about GitHub?

What Is GitHub?

In the discussion of Git vs. GitHub, it's been said that GitHub is to Git what Facebook is to your *actual face*. What's that mean? Well, it means that while Facebook is kind of like an online face database (of sorts), GitHub is designed as a Git repository hosting service.

And what exactly is a Git repository hosting service? It's an online database that allows you to keep track of and share your Git version control projects outside of your local computer/server. Unlike Git, GitHub is exclusively cloud-based. Also unlike Git, GitHub is a for-profit service (although basic repository-hosting features are available at no cost to those who are willing to create a user profile, making GitHub a popular choice for open-source projects).

That's because, in addition to offering all of the features and advantages of Git, GitHub expands upon Git's basic functionality. It presents an extremely intuitive, graphically represented user interface, and provides programmers with built-in control and task-management tools. Additional features can be implemented via the GitHub Marketplace service. And because GitHub is cloud-based, an individual's Git repositories can be remotely accessed by any authorized person, from any computer, anywhere in the world (provided it has an internet connection).

Through GitHub, you can share your code with others, giving them the power to make revisions or edits on your various Git branches. This makes it possible for entire teams to coordinate together on single projects in real-time. As changes are introduced, new branches are created, allowing the team to continue to revise the code without overwriting each other's work. These branches are like copies, and changes made on them do not reflect in the main directories on

other users' machines unless users choose to push/pull the changes to incorporate them. There is also a [GitHub desktop](#) application available, which offers some additional functionality for experienced developers.

Other Git repository hosting services also exist; GitLab, BitBucket, and SourceForge are all viable GitHub alternatives, and GitLab even offers a built-in option which allows GitHub users to migrate their projects directly into GitLab.

Git vs. GitHub in Simple Terms

In Simple Terms

Git

is a version control system that lets you manage and keep track of your source code history

Git
Hub

is a cloud-based hosting service that lets you manage Git repositories

So, taken all together: Git vs. GitHub... what's the difference? Simply put, Git is a version control system that lets you manage and keep track of your source code history. GitHub is a cloud-based hosting service that lets you manage Git repositories. If you have open-source projects that use Git, then GitHub is designed to help you better manage them.

After all, in the world of programming, naming conventions aren't always intuitive. That's why it's worth recognizing the connections and the differences in the similarly named *Git* and *GitHub*. Both Git and GitHub give programmers valuable version-control functionality so that they can build ongoing coding projects without being afraid of messing everything up. GitHub just takes things a

little bit further than Git, offering more functionality and resources, as well as a place online to store and collaborate on projects.

And that's essentially it. Simple, right?

Is a Coding Bootcamp Right for You?



Grab Your Web Dev Course Outline

First name*

Last name*

Email*

Full-Time or Part-Time*

SEND ME THE OUTLINE

Ready to Apply to Devmountain?



Recommended Reading

[What Is GitHub and How Do You Use It?](#)

[CSS Cheat Sheet: Your Quick Reference for Cascading Style Sheets](#)

[What Is Serverless Architecture and What Does It Mean for Programmers?](#)

📌 Web Dev, Career Support

Read On

What Is GitHub and How Do You Use It?

What Is the Difference Between Graphic Design and UX Design?

Java vs Python: How Are They Different?

COMPANY

[Home](#)

[About](#)

[Contact](#)

[Work for Us](#)

[Policies](#)

[Press](#)

STUDENTS

[Approach Login](#)

[Apply](#)

[Blog](#)

[Beginners Guide](#)

[FAQ](#)

[Our Students](#)

COMMUNITY

[DM Hub](#)

[Strayer](#)

[Instructors](#)

[Events](#)

[Shop](#)

FOLLOW US

