

# What is GitHub And How To Use It?

GitHub is an increasingly popular programming resource used for code sharing. It's a social networking site for programmers that many companies and organizations use to facilitate project management and collaboration. According to [statistics collected in October 2020](#), it is the most prominent source code host, with over 60 million new repositories created in 2020 and boasting over 56 million total developers.

Before we take an in-depth look at what GitHub is, let's check out a use case and gain an appreciation for its relevance.

## A GitHub Use Case

Let's consider the case of Decathlon, the world's largest sporting goods retail brand. The company has over 1600 stores in 57 countries, with more than 87,000 employees.

Every company, no matter how large or small, inevitably experiences challenges and obstacles. We could best summarize Decathlon's problems as:

1. How would the company maintain workflow visibility and avoid redundancies in such a large workforce?
2. How would the company hire developers for so many diverse locations?

[GitHub](#) to the rescue! GitHub is not only an affordable resource but also features a great open-source community. Since it is a cloud-based tool, the code is conveniently visible across the entire client organization, facilitating every participant's contributions.

GitHub allows collaboration with developers from all over the world. Open-source solutions like GitHub enable potential developers to contribute and share their knowledge to benefit the global community.

The version control system, or VCS, is the element in [Git](#) that is best suited for tackling Decathlon's two problems. So, let's expand our knowledge of GitHub by taking a closer look at the Git version control system and see why it's such a game-changer.

# What is a Version Control System?

The Git version control system, as the name suggests, is a system that records all the modifications made to a file or set of data so that a specific version may be called up later if needed. The system makes sure that all the team members are working on the file's latest version, and everyone can work simultaneously on the same project.

Before we dig deeper into what GitHub is, we must examine first what the 'Git' part is all about.

## What is Git?

[Git](#) is a version control system used for tracking changes in computer files, making it a top-rated utility for programmers world-wide. Git can handle projects of any size.

Git is used to coordinate the workflow among project team members and track their progress over time. It also benefits both programmers and non-technical users by keeping track of their project files. Git allows multiple users to work together without disrupting each other's work.

Now that you've been introduced to Git, you have the foundation needed to understand what is GitHub better.

## What is GitHub?

GitHub is a Git repository hosting service that provides a web-based graphical interface. It is the world's largest coding community. Putting a code or a project into GitHub brings it increased, widespread exposure. Programmers can find source codes in many different languages and use the command-line interface, Git, to make and keep track of any changes.

GitHub helps every team member work together on a project from any location while facilitating collaboration. You can also review previous versions created at an earlier point in time.

So now we know what Git and GitHub are. Time to gain a better understanding of the importance and relevance of what is GitHub by exploring its features.

## What are GitHub's Features

### 1. Easy Project Management

GitHub is a place where project managers and developers come together to coordinate, track, and update their work so that projects are transparent and stay on schedule.

### 2. Increased Safety With Packages

Packages can be published privately, within the team, or publicly to the open-source community. The packages can be used or reused by downloading them from GitHub.

### 3. Effective Team Management

GitHub helps all the team members stay on the same page and organized. Moderation tools like Issue and Pull Request Locking help the team to focus on the code.

### 4. Improved Code Writing

[Pull requests](#) help the organizations to review, develop, and propose new code. Team members can discuss any implementations and proposals through these before changing the source code.

### 5. Increased Code Safety

GitHub uses dedicated tools to identify and analyze vulnerabilities to the code that other tools tend to miss. Development teams everywhere work together to secure the software supply chain, from start to finish.

## 6. Easy Code Hosting

All the code and documentation are in one place. There are millions of repositories on GitHub, and each repository has its own tools to help you host and release code.

Now that we've gained some familiarity with GitHub let's check out the competition.

## So How Do You Get Started With GitHub?

It's easy to get things going with GitHub. For starters, click onto the [GitHub site](#) and create an account. Then, consider installing [Git](#) on your system, especially if you plan on using your local computer. Then, go to your terminal and make yourself [known to Git](#) by setting up your user name in every repository. Use this command:

```
git config --global user.name "<your_name_here>"
```

Make sure the "your name here" parameter is your own name. Pick any name you'd like.

Next, share your email address with Git. It should be the same address you entered when you joined GitHub.

```
git config --global user.email "<your_email@email.com>"
```

You're now ready to use Git!

## How Do You Use Git and GitHub?

Here's a very broad overview of the steps you need to use both Git and GitHub. You can find more details regarding the specific commands and syntax here on [opensource.com](https://opensource.com).

1. Create your GitHub account, which you should have already done, thanks to the previous section!
2. Create a repository or "repo" for short. This is where you store your code.
3. Build a file.
4. Make a commit. Whenever you create a file or change it, you create a Git commit to store the new version.
5. Connect your repo with your computer system.

## GitHub's Competitors

The market provides many alternatives and competitors to GitHub. As of the end of 2020, the top ten competitors are:

1. Bitbucket
2. Google Cloud Source Repositories
3. Phabricator
4. GitLab
5. Gogs
6. Gitea
7. SourceForge
8. Apache Allura
9. Launchpad
10. AWS CodeCommit

Enroll in [DevOps Engineer Master's Program](#) and learn to work on tools like Docker, Git, Jenkins, Puppet & Nagios.

## Wrapping It Up

We hope this article has helped you understand GitHub's relevance. You learned the basics of a version control system and how Git works. We explored why GitHub's features make it such a vital resource, then wrapped up by listing GitHub's alternatives and competition.

For more information, check [this](#) video, which goes into more detail about what is GitHub and the many ways you can use it.

## Want to Learn More About Git?

Since Git is such a popular and valuable tool in modern DevOps environments, that means there are plenty of career opportunities for developers who know it well. If this introduction piqued your interest in Git, consider enrolling in Simplilearn's [Post Graduate Program in DevOps](#), created in collaboration with Caltech CTME today!