

# Introduction to Git: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2019

## Syntax

- Getting started with Git

```
git
```

- Initializing a repo

```
git init
```

- Check the state of each file

```
git status
```

- Add files to staging area

```
git add
```

- Configure identity in Git

- Configure email

```
git config --global user.email "your.email@domain.com"
```

- Configure name

```
git config --global user.name "Your name"
```

- Making a commit

```
git commit -m "Commit message here"
```

- Viewing the diff

- View the diff before staged

```
git diff
```

- View the diff after staged

```
git diff --staged
```

- View repo's commit history

```
git log
```

## Concepts

- Distributed version control systems will "merge" changes together intelligently, and exist to enable multiple developers to work on a project at the same time.
- A repository (or "repo") tracks multiple versions of the files in the folder, enabling collaboration.
- While there are multiple distributed version control systems, Git is the most popular.
- Files and folders with a period prefix are typically private.
- Commits are checkpoints that you store after adding files and/or making changes.
- A diff are the changes between commits.
- Files can have one of three states in Git:
  - **committed** : The current version of the file has been added to a commit, and Git has stored it.
  - **staged** : The file has been marked for inclusion in the next commit, but hasn't been committed yet (and Git hasn't stored it yet). You might stage one file before working on a second file, for example, then commit both files at the same time when you're done.
  - **modified** : The file has been modified since the last commit, but isn't staged yet.

## Resources

- [Git Documentation](#)

- [GitHub's Blog](#)



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2019