



# Intro to Git

# Intro to Git

---

So far, we have used GitHub as a sort of “drop box” to store our files.

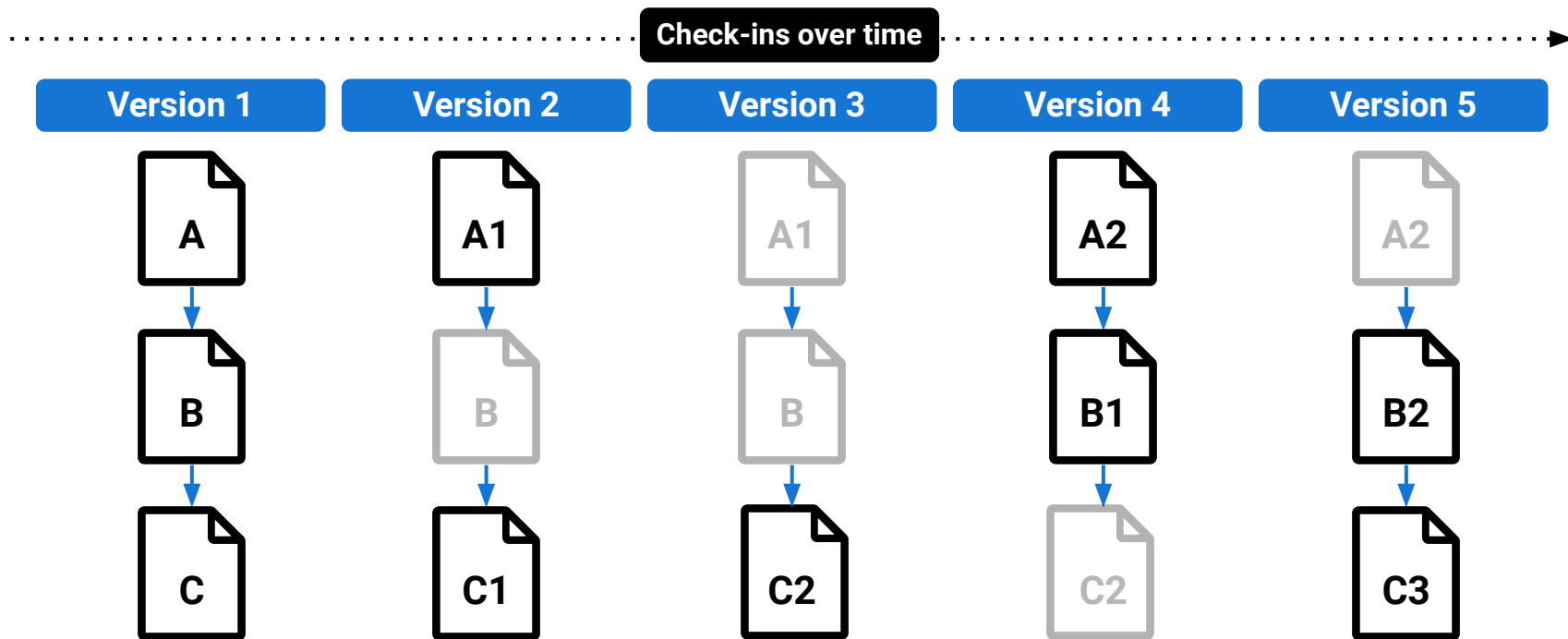
Although this is one way to use GitHub, it has much deeper capabilities.

Today, we will delve into Git and how to use it through the terminal to interact with GitHub.



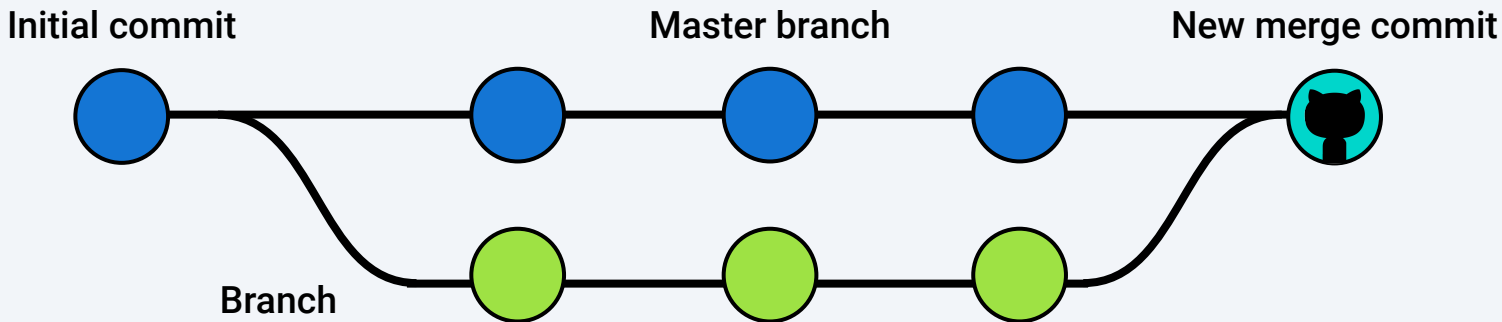
# Intro to Git

Git is essentially a way for us to keep track of our work over time. Whenever we get another piece of a project working, we can save the change with Git.



# Git Commit

A Git “save” is called a **commit**. It represents a checkpoint for our project where we save and describe our work.

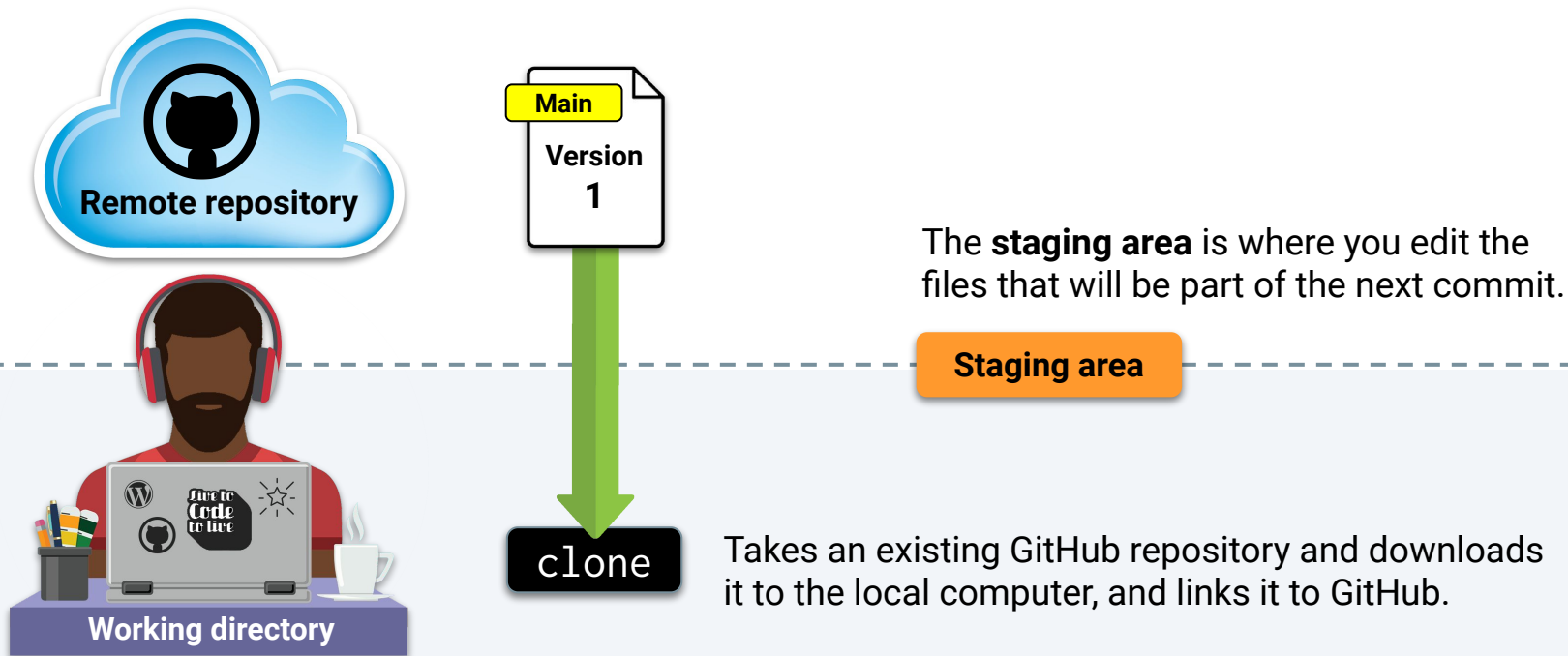


If we break something while working on our code, this system allows us to restore working code from earlier. Since Git remembers these checkpoints, we can work on several different concerns all at once.

# Git Version Control

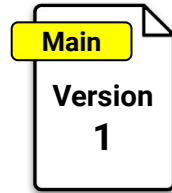
**Scenario:** Your group has been working with Uber's rider data, and you've decided to analyze the average age of the riders:

The root code for the project is called **main**.



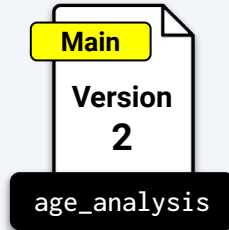
# Git Version Control

Git essentially allows us to write this code and save it with the name `age_analysis`.



The **staging area** is the where you edit the files that will be part of the next commit.

**Staging area**



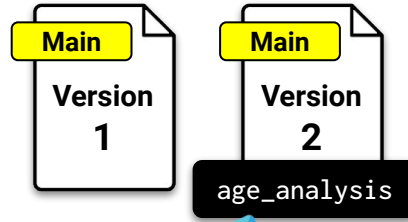
`git commit`



Your staged changes are saved once you commit.

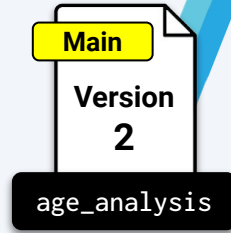
# Git Version Control

**age\_analysis** is a branch that originates from the main branch. It contains updates that will be added to the main branch when it's ready to **merge**.



The **staging area** is the where you edit the files that will be part of the next commit.

**Staging area**



# Popular Git CLI Commands

---

<code>git clone</code>	Clones a git repository onto the local file system.
<code>git add</code>	Adds changed files to the queue of tracked files ready to be committed.
<code>git commit</code>	Adds tracked files as a bulk checkpoint ready to be pushed to the remote git repository.
<code>git push</code>	Uploads changed files from the local git repository to the remote git repository and updates the remote files.
<code>git pull</code>	Downloads changed files from the remote git repository to the local git repository and updates the local files.



A commit in GitHub is like a snapshot of what your project or file looks like at a particular moment in time. If a file doesn't contain any changes, the file is not stored again; instead, Git provides a link to the identical file that it previously stored.





# Time to Code

## Adding Files from the Command Line

Suggested Time:

---

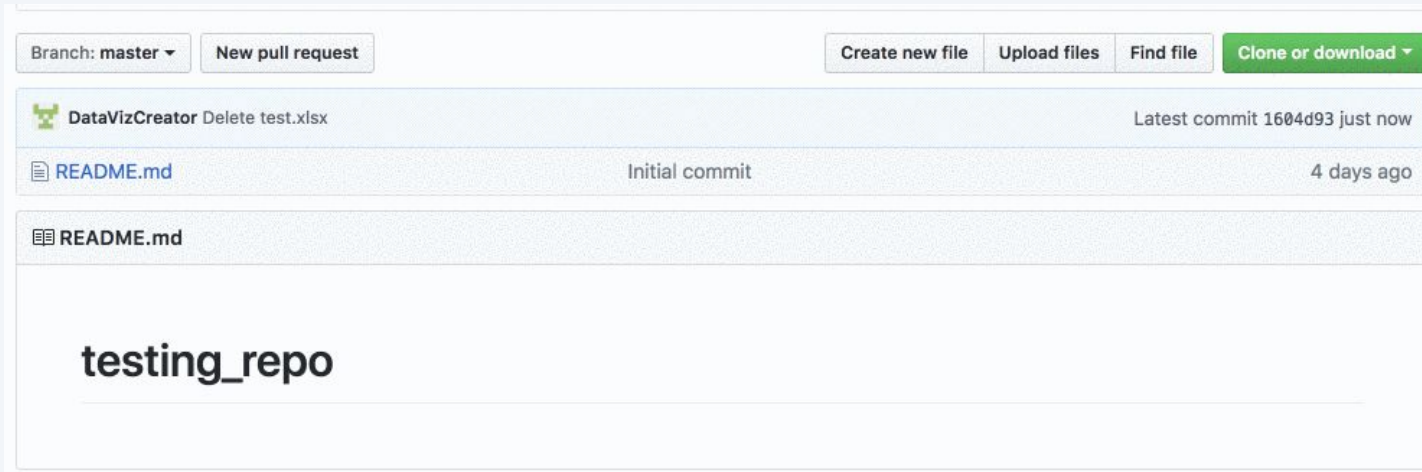
10 Minutes

# Activity: Adding Files from the Command Line

## Instructions

Create a new repo.

From the repo page, click the green box in the top-right labeled "Clone or download", select "Use SSH", and copy the link to the clipboard, as captured in the following GIF:



# Activity: Adding Files from the Command Line

---

## Instructions

Open Terminal (or Git Bash for Windows users), and navigate to the home folder using `cd ~`.

Type `git clone <repository link>` in the terminal to clone the repo to the current directory. Once this code has run, everyone should find a folder with the same name as the repo:

```
$ git clone git@github.com:DataVizCreator/testing_repo.git
```

Open the folder in VS Code, and create two python script files, named `script01.py` and `script02.py`.

# Activity: Adding Files from the Command Line

---

## Instructions

Then, open Terminal/Git Bash, and navigate to the repo folder. Run the following lines:

```
# Displays that status of files in the folder  
git status
```

```
# Adds all the files into a staging area  
git add .
```

```
# Check that the files were added correctly  
git status
```

```
# Commits all the files to your repo and adds a message  
git commit -m <add commit message here>
```

```
# Pushes the changes up to GitHub  
git push origin main
```

Navigate to the repo on [Github.com](https://github.com) to confirm that the changes have been pushed up.



# Activity: Adding More to the Repo

In this activity, you will make or add changes to the repo that we just created.

Suggested Time:

15 Minutes

# Activity: Adding More to the Repo

---

## Instructions

Using the repo that we just created, make or add the following changes:

- Add new lines of code to one of the Python files
- Create a new folder.
- Add a file to the newly created folder.
- Add, commit, and push the changes.
- Delete the new folder.
- Add, commit, and push the changes again.



Time's Up! Let's Review.



# Questions?



*The  
End*