

# Git & GitHub

# Git: History

# History of Git

- Made in 2005 by Linus Torvalds
- Before that, he made the Linux Kernel
  - Here is a [Ted talk](#)
  - Here is his [GitHub](#)
  - Here is the [source code for Git](#)

# **Why is it called Git?**

**"I'm an egotistical bastard, and I name all my projects after myself"**

**- *Linus Torvalds***

## Warning!

- "Git is infuriating" - Mandy Brown
- It takes a long time to feel comfortable with it
- Most explanations of it get very technical very quickly
  - Focus on the concepts

# What is it?

- **A version control system (or VCS)**
  - It takes snapshots of our projects
  - Gives us a project-wide undo button!
- A collaboration tool
  - It merges differences in our code for us
- A local development tool
- Supports non-linear development

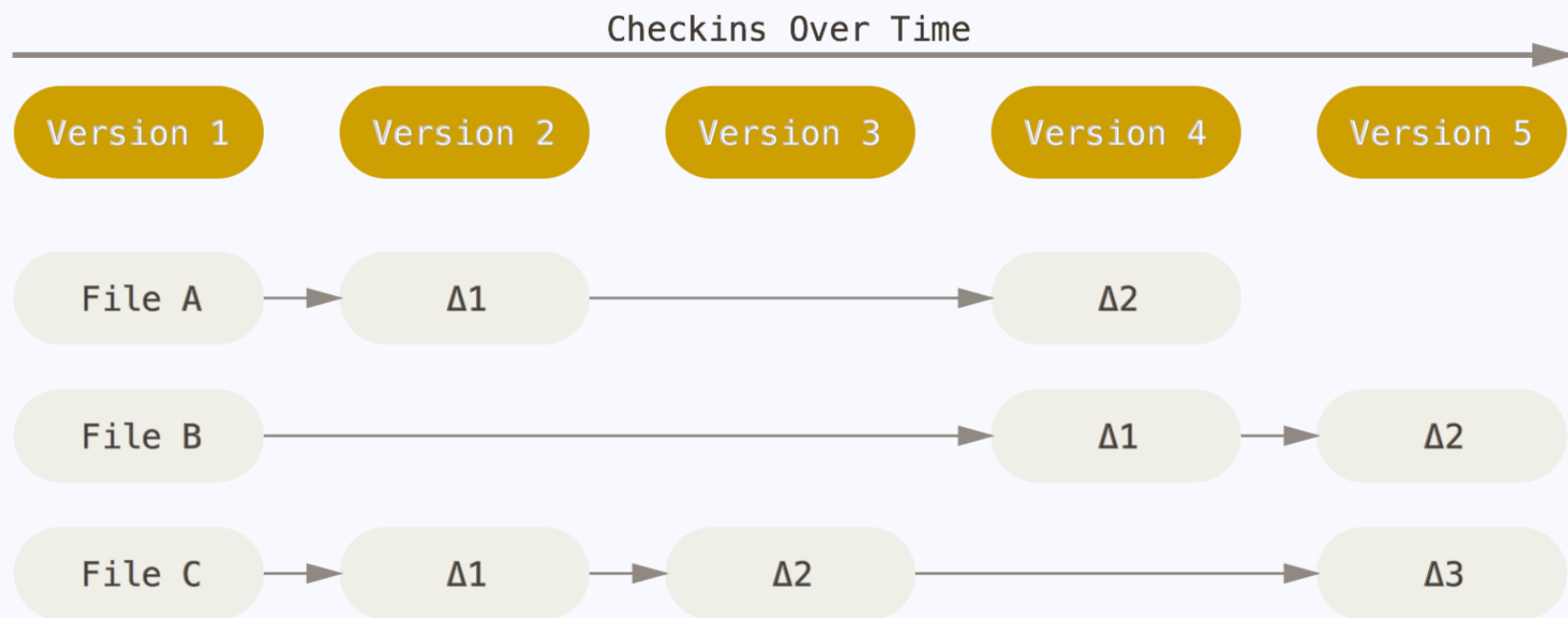
# What is it?

It's a tool for modern-day teamwork

For people who are working asynchronously, on a shared body of work

It saves us from moving floppy disks around, or saving lots of copies of the one file.

More people === more likely to use it





# Why use it?

- You make a change and realise it was a horrible mistake?
  - **Git can undo it**
- You want to figure out where everything went wrong?
  - **Git will show you**
- You want to try out a new innovative feature that will probably destroy everything?
  - **Git can protect you**
- You want to work with a bunch of people?
  - **Git will make that easier**

## **Remember the audience**

The ultimate audience of Git is you.

It takes a long time to get used to!

# Concepts

# Terminology

- **Repository**
  - A project
- **Add**
  - Tell Git to pay attention to a file(s)
- **Commit**
  - Tell Git to take a snapshot of a file(s)
- **Origin**
  - A place where your code is stored

# Terminology

- **Push**
  - Put all the code up on GitHub
- **Branch**
  - A version of your project
- **Clone**
  - When you copy a project from GitHub to your computer

# Terminology

- **Fork**
  - Your copy of someone else's GitHub repository
- **Merge Conflict**
  - What happens when two pieces of code can't be automatically merged
  - You need to decide what you want
- **Pull Request**
  - When you request to have a project include your code

**How do we use Git?**

# How do we use Git?

- The Command Line
- Applications
  - [GitHub Desktop](#)
  - [SourceTree](#)
  - [GitKraken](#)
  - Plus more...



## From here on...

- If you just installed Ubuntu
  - Run all commands in the Ubuntu terminal
- If you just installed Cygwin
  - Run all commands in the Cygwin terminal
- If you installed iTerm2
  - Run all commands in iTerm2

## Git - Linux (Ubuntu)

- `sudo apt update`
- `sudo apt install git`
- Restart the terminal
- `git --version`

# Git - OSX

- Install Command Line Tools
  - `xcode-select --install`
- Install [Brew](#)
  - Restart the terminal
- Install Git
  - `brew install git`
- Restart the terminal
  - `git --version`

# Configure Git

- `git config --global user.email "YOUR GITHUB EMAIL"`
- `git config --global user.name "YOUR GITHUB NAME"`
- `git config --global color.ui "auto"`
- `git config --global core.editor "code --wait"`

# Git Commands

# Git Commands

```
git init
```

Creates a new local **repository**

# Git Commands

```
git status
```

Shows what is currently happening with your repository

# Git Commands

```
git add README.md
```

```
git add .
```

Makes Git watch certain files



# Git Commands

```
git commit -m "A commit message"
```

Takes a snapshot of a file

# Git Commands

```
git log
```

Shows the previous commits

## Exercise

- Make a folder and a few files within it
- Turn it into a Git repository
- Make sure Git is keeping track of all files in the folder!
- Take a snapshot.

Note: it won't keep track of empty folders unless there is a file in there

## Resources

- [Atlassian: Learn Git](#)
- [Official GitHub Git Tutorial](#)
- [CodeSchool](#)
- [Code Academy](#)
- [Git & GitHub for Poets](#)
- [Git For Humans](#)

**Git != GitHub**

**GitHub**

# What is GitHub?

- It is a website that hosts Git repositories
  - Helps with collaboration
- It is a Graphical User Interface (GUI)
- Helps us perform common tasks
- The Dropbox or Google Drive for code

# Why do we use it?

- To share our code with other computers
- For collaboration (Pull Requests, Forks etc.)
- It acts as a portfolio
- To visualise what is going on
- As a project management tool (Projects)
- An error reporting system (Issues)
- Documentation (Wiki)
- Free hosting ([GitHub Pages](#))



# What will we be using it for in class?

- The lead instructor(s) will use it to:
  - Share code and slides
  - Share the Homework
- The IA(s) will use it to:
  - Share Warmup Exercises and solutions
  - Give feedback on Homework
- For the students:
  - You upload your homework to GitHub
  - You upload your projects to GitHub
  - It will act as your portfolio

# How do we use it?

Once you have a **local** Git repository...

- Create a repository on GitHub
- We need to tell Git where the code should be stored
  - `git remote add origin URL`
- We need to **push** (or upload) all of the code
  - `git push origin master`

## How do we use it?

Once you have a **local** Git repository...

- We need to **pull** (or download) all of the code
  - `git pull origin master`
- We can also **clone** a repository
  - `git clone URL`

# A Typical Upload Workflow

```
git init
```

```
git add -A
```

```
git commit -m "Made changes"
```

```
git remote add origin URL
```

```
git pull origin master
```

```
git push origin master
```

# Git Remotes

```
git remote -v
```

To see what GitHub repos you are connected to

# A Typical Clone Workflow

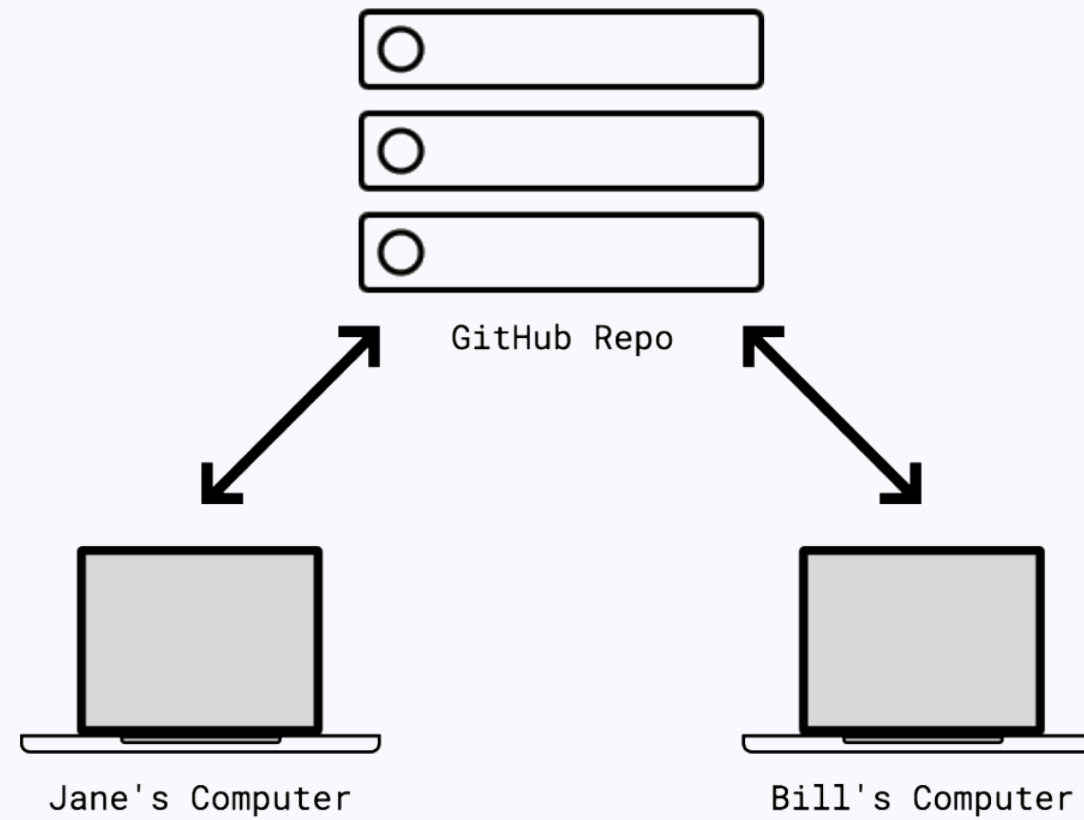
```
git clone URL
```

```
cd SOME_DIRECTORY
```

```
git pull origin master
```

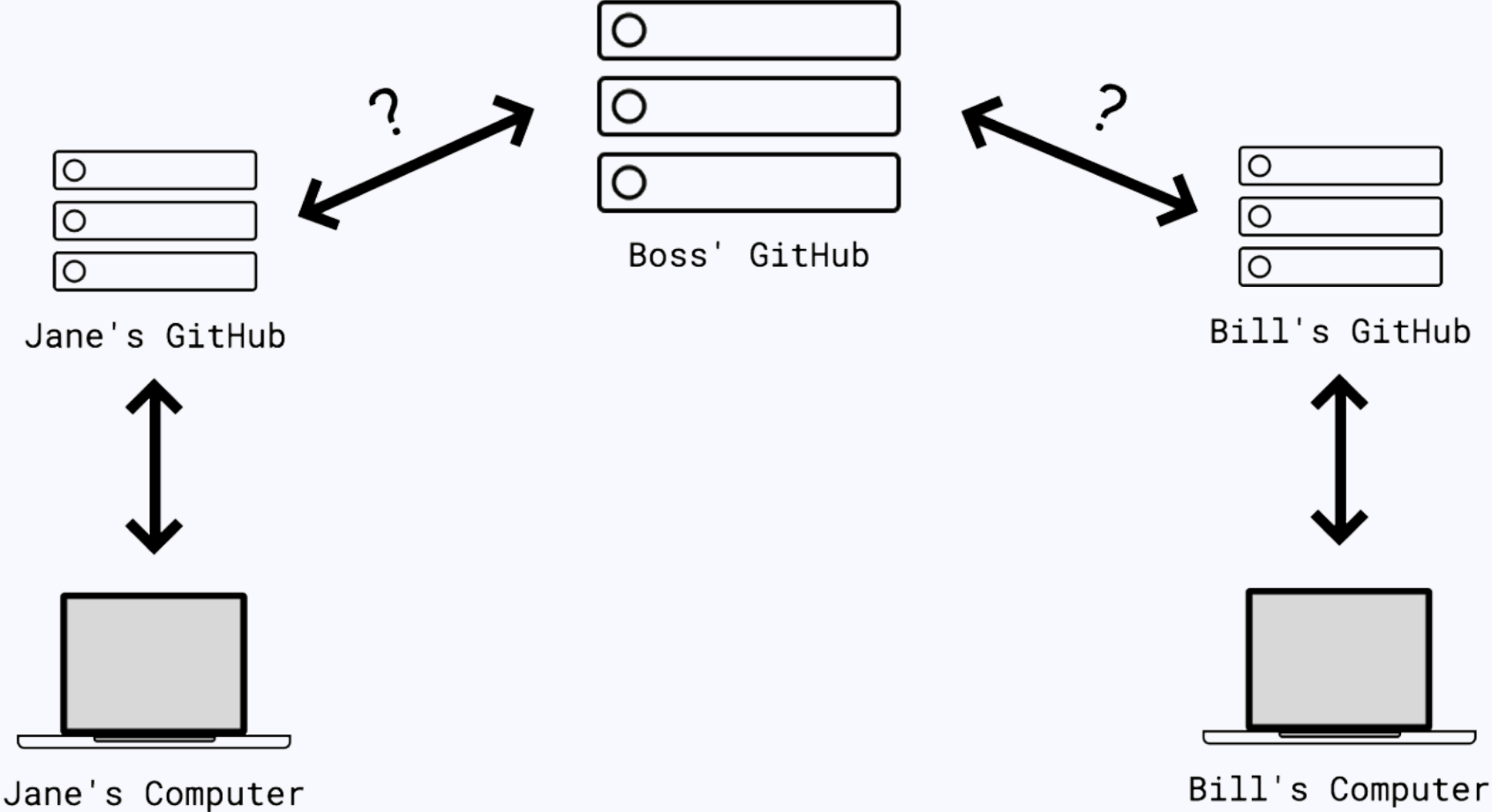
# Collaboration Approaches

# Push Access





# Fork & Pull Request



## Exercise

- Put the Git repository you created earlier on GitHub!
- Once you have it on GitHub:
  - Add a file locally
  - Upload that new file to GitHub