



# Git Basics

CS Honors Git Module

Slides by Matt Geimer (SP22)



# What is Git?

- Git is a **Version Control System**
  - Can recover previous versions of your work
- Can be used to collaborate with other programmers
- Most commonly used version control system





# How popular is Git?

## Companies that use Git





# 94%

Of professional developers use Git




# What is Git?

- Git is a **Version Control System**
  - Can recover previous versions of your work
- Can be used to collaborate with other programmers
- Most commonly used version control system
- How do you use Git?





# How does Git work?

- Git stores your code in snapshots
- These snapshots are called **commits** because the developer “commits” them to the log
- These commits are stored in a **repository**, which stores other metadata like when the commit was made or who made the commit
- Commits can be uploaded to external repositories on other servers (like **GitHub**) 
- At any point, a developer can **checkout** a commit



🙄 That's a lot of vocab



# Git: Visualized

## How does Git work?

### ● Initial Commit

#### Shopping List

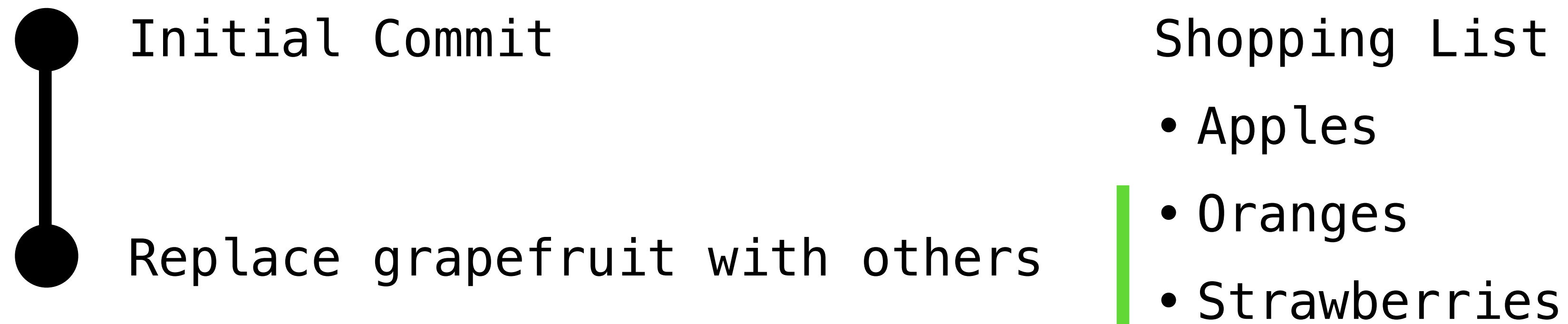
- Apples
- Grapefruit





# Git: Visualized

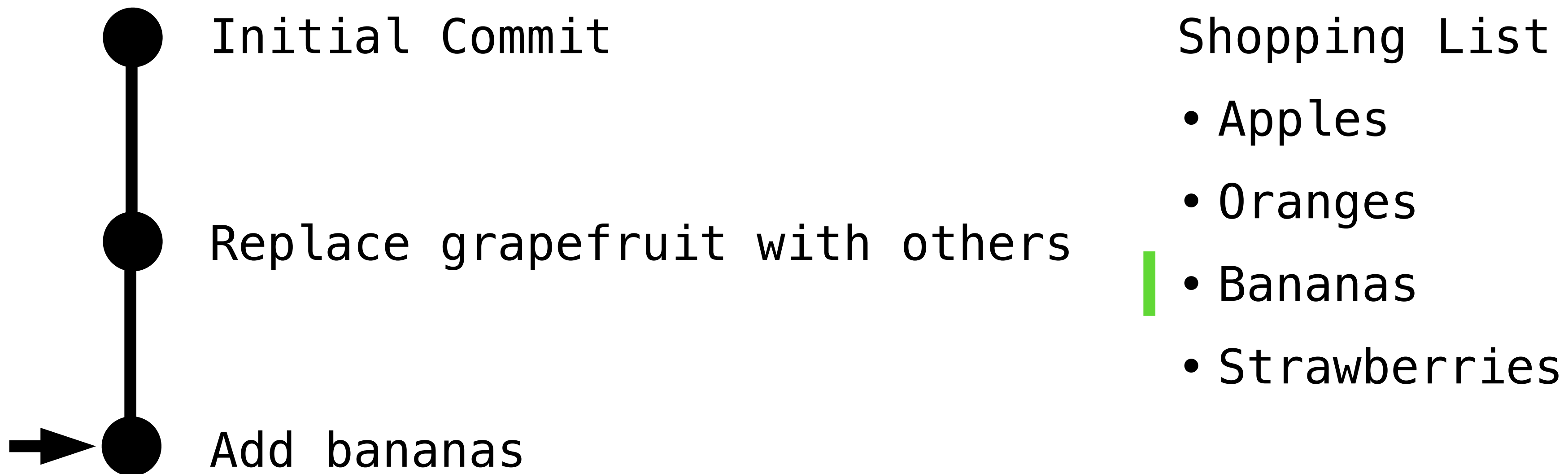
## How does Git work?





# Git: Visualized

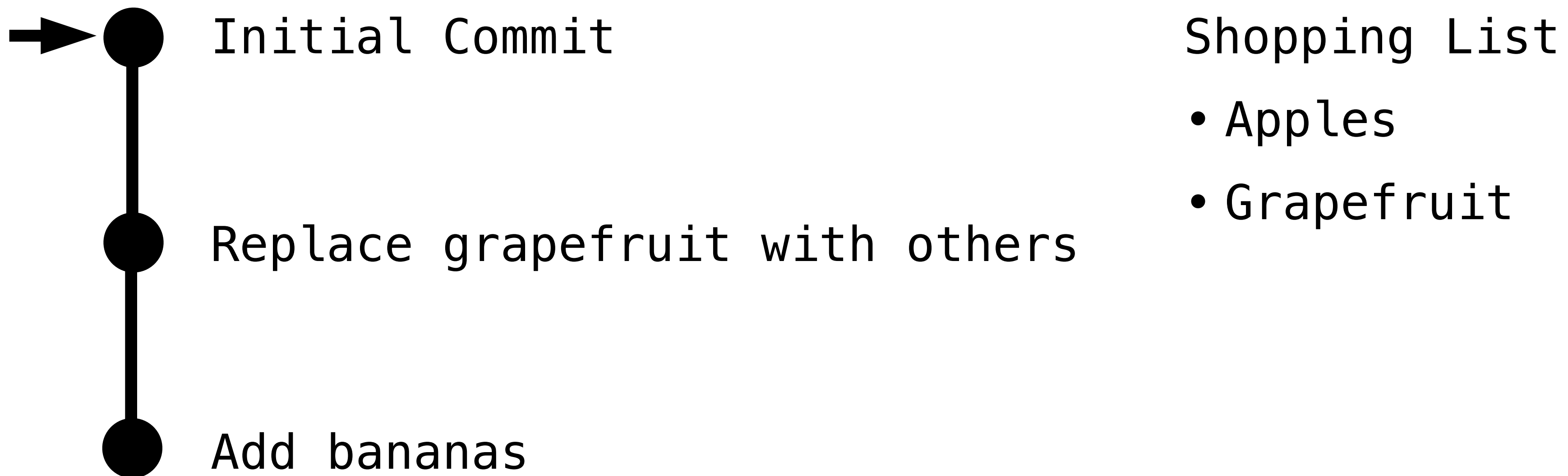
## How does Git work?





# Git: Visualized

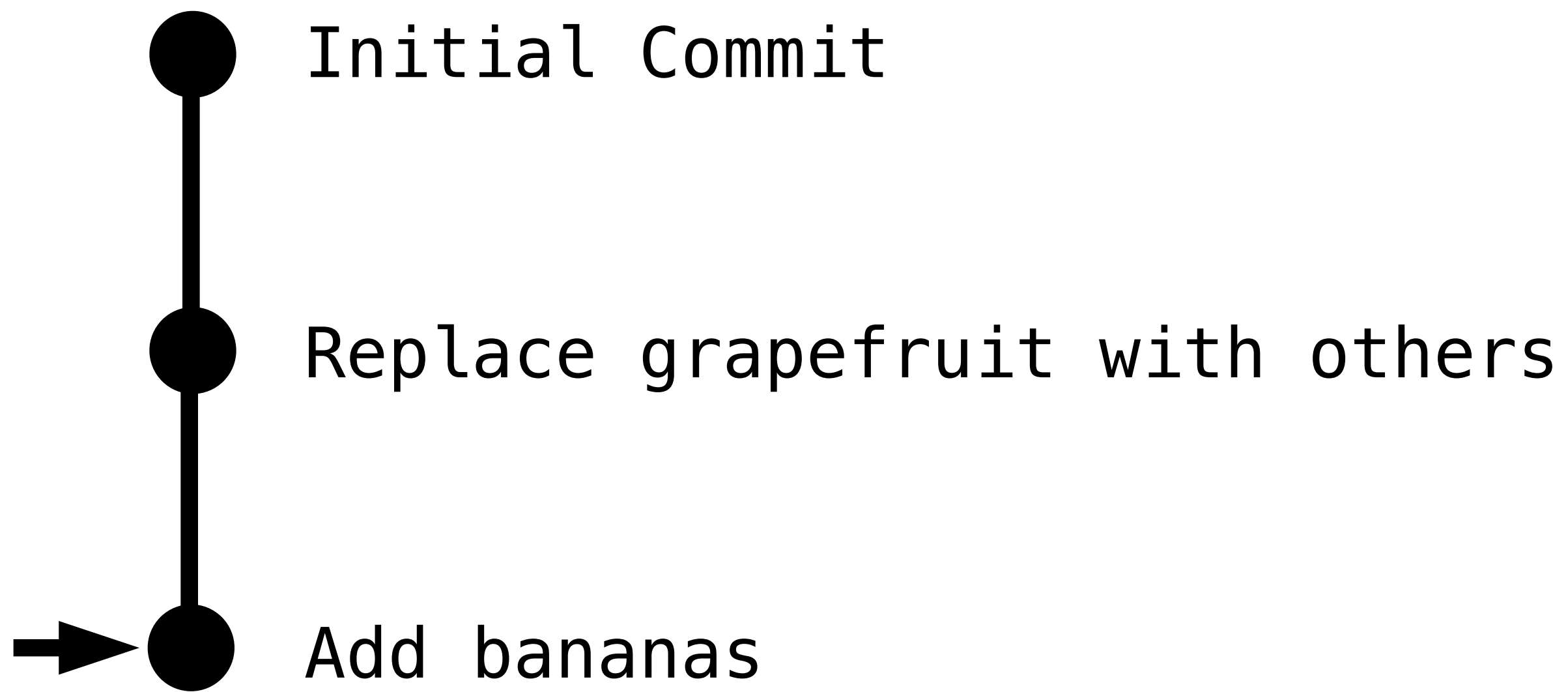
## How does Git work?





# Git: Visualized

## How does Git work?



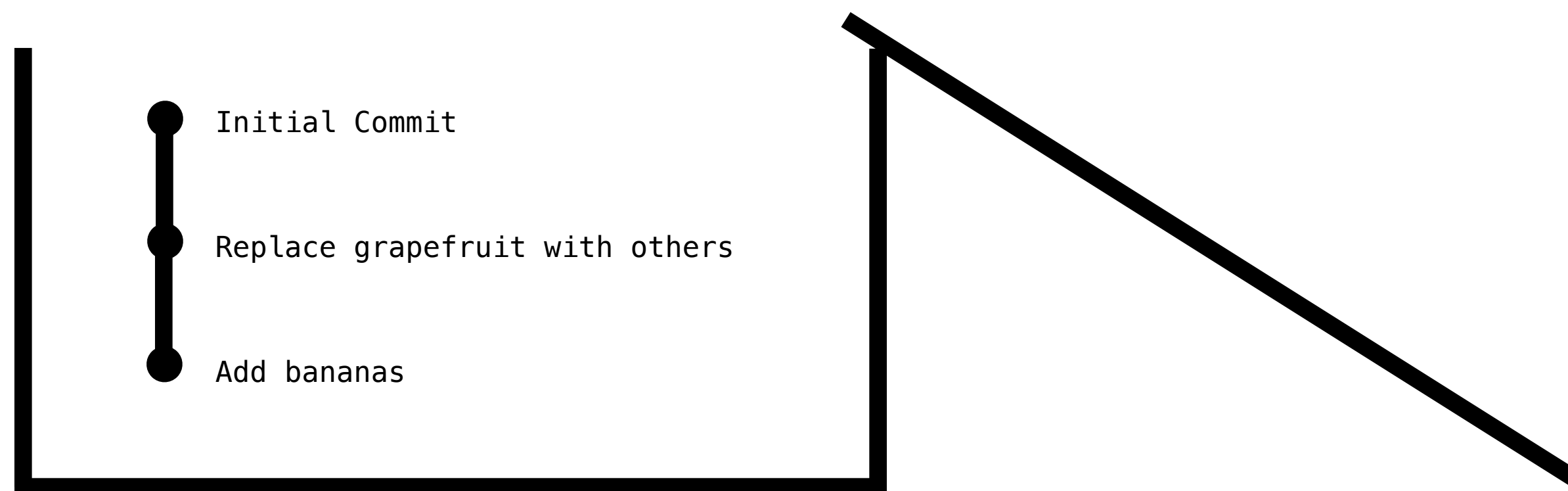
### Shopping List

- Apples
- Oranges
- Bananas
- Strawberries



# Git: Visualized

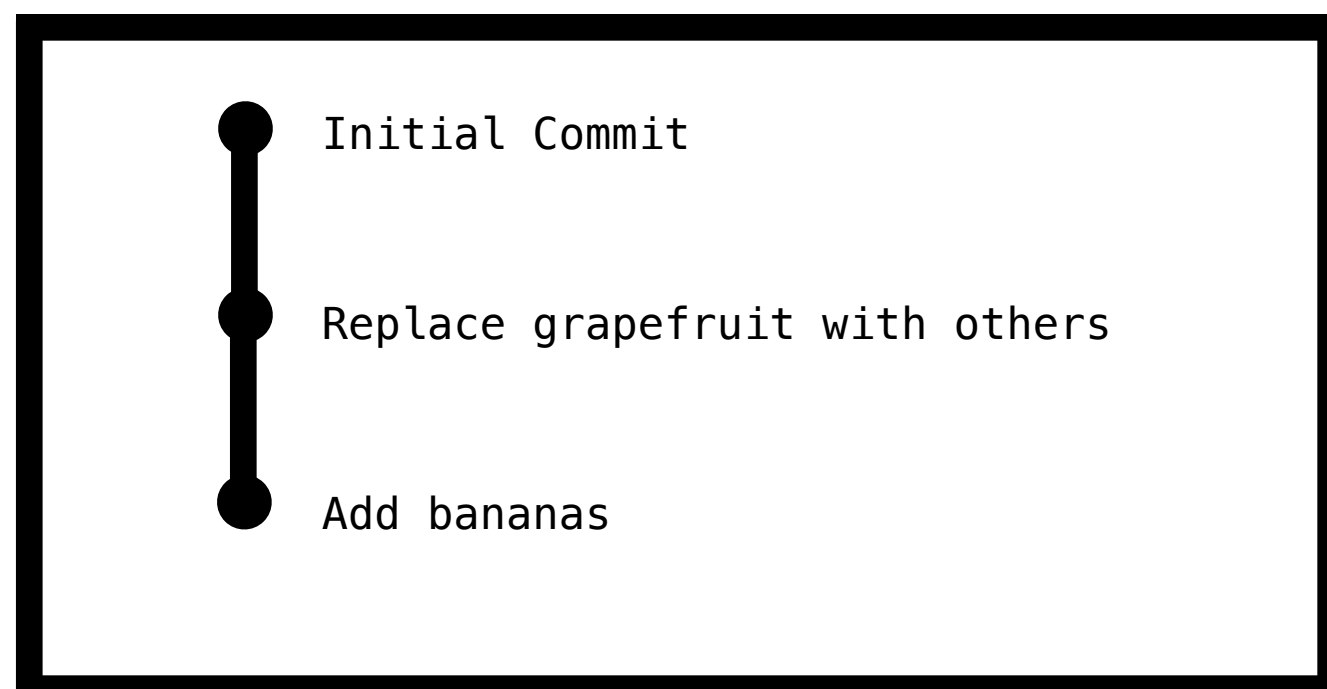
## How does Git work?





# Git: Visualized

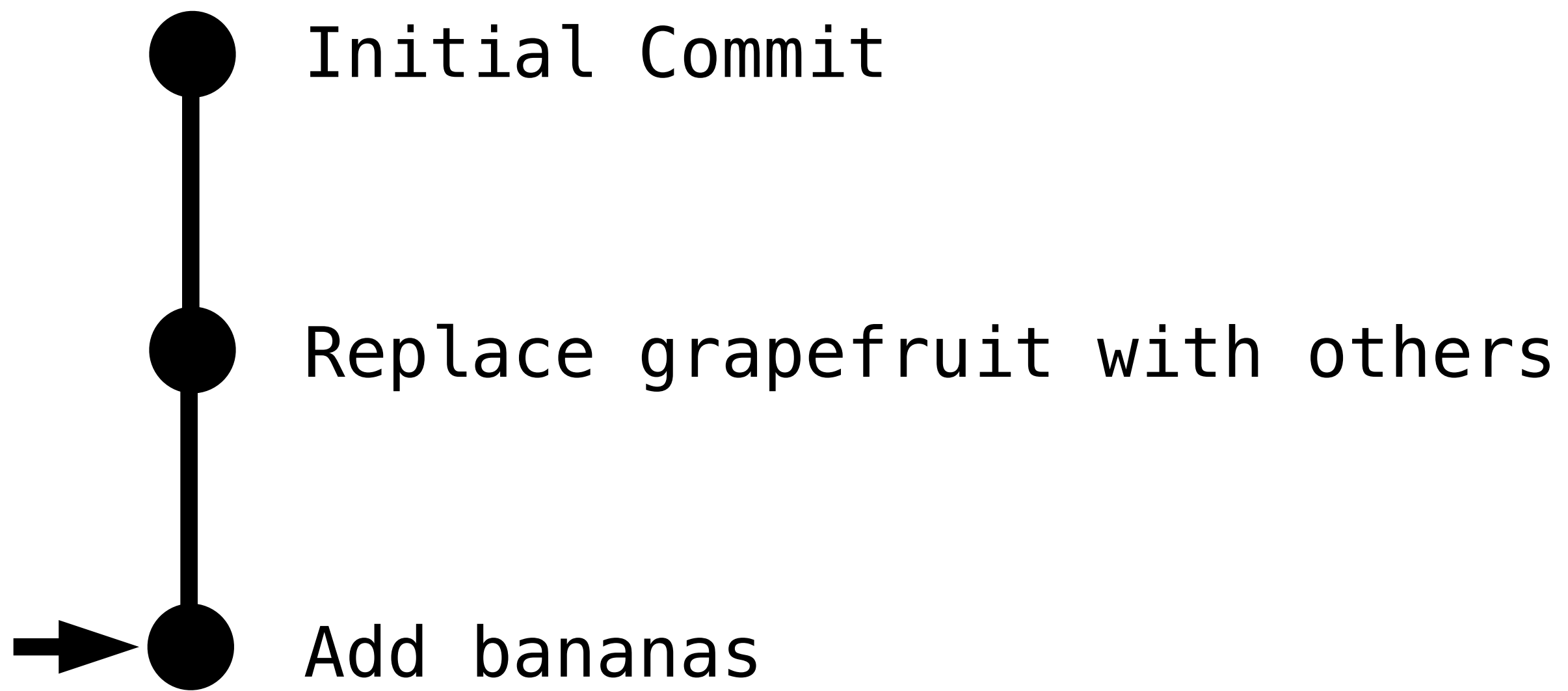
## How does Git work?





# Git: Visualized

How does Git work?





# Vocab Review

## Git Basics

- **Directory** - A folder on a computer system
- **Commit** - A snapshot of your code at the time the commit was made
- **Repository** - Where all the commits are stored
- **Local Repository** - The repository on your computer
- **Origin** - A repository hosted on a server (usually **GitHub**)





# So how do we use Git?



# git init / git clone

## Git Basics

- Init
  - Creates empty repository in the current directory
  - There will be no commits or files tracked initially
  - Usage: `git init`
- Clone
  - Downloads an external repository and sets it up as a local repository in the current directory
  - Usage: `git clone {url}`





# git add

## Git Basics

- Marks files to be committed in next commit
- Does **NOT** commit the files automatically
- Usage:
  - `git add {file/directory}`
  - `git add -A`
  - `git add -u`
- N.B. – `git add` will not add files in `.gitignore`





# git commit

## Git Basics

- Commits staged files to the repository
- Attaches your email/name and when you commit
- Usage:
  - `git commit`
  - `git commit -m "My commit message"`
  - `git commit -am "Commit all changed"`





# git push

## Git Basics

- Uploads any new commits to Origin (GitHub)
- Usage:
  - `git push`
  - `git push --set-upstream origin main`







# git pull

## Git Basics

- Downloads any commits in origin but not in your local repository
- Usage:
  - `git pull`





# Lesson Overview

## Git Basics

- Showed how to use
  - `git clone / git init`
  - `git add`
  - `git commit`
  - `git push`
  - `git pull`



# Git Basics

CS Honors Git Module

CS 124 Honors



CS 128



Slides by Matt Geimer (SP22)