# CLI

* Command Line interface
* It is a program that interacts with your computer to do certain operations
* It can make folders, open folders, do a plethora of functions
* Why is this important?
  + Well in the coding work, there is little to no such thing as a pretty UI
  + Everyone is focused on functionality than adding a nice graphic representation of it

## Arguments

* They are essentially ways to add a specific direction you want to do with your command
* There is a bunch of arguments each command has and knowing them all is next to impossible

## Commands

* Help – shows all potential commands you can use in a terminal
* Pwd – shows the current directory the terminal is in
* Ls – Lists all the first inside of that directory
* Cd – changes the current directory of the terminal
* Mkdir – makes a directory
* Echo – displays a line of text
* Touch – Allows you to create a file/change the data modified on that file
* Nano – Uses an interface to edit files (Think of a Microsoft word or notepad)
* Grep – Searches for patterns in a file (think of Microsoft word search functionality)
* Cat – concatenate/combine two files together
* Which – locate the path file of your commands in terminal
* Find – Finds certain directories of files

# Shell Scripting Fundamentals

## Terminal Emulator

* It is an application we use to communicate with the computer
* This can either be the default command prompt or git bash or powershell

## Command Line

* This is the different types of commands we use to start interacting with the computer
* So, think of the cd, ls, and mkdir commands we have been using so far
* Abbreviated as CLI or Command-line Interface
  + As we go deeper in this program, we will install other frameworks, SDKs, or libraries that will essentially expand our CLI to include.

## Shell

* A way for you to execute programs or gather inputs from the user to do something
* So, think of it as combining all those CLI commands to do something in your computer

# Basic Programming

## Variables

* They are just a way for us to store information that we can then reference to later on
* System defined variables
  + These variables change the information stored in them on the computer they are running on
  + System defined is the keyword in the phrase that essentially say System meaning the computer is the one that defines those variables for you
* User defined variables

# Control Flow

* They are a way to control what commands should go first or what commands should be repeated or what commands can only execute if some condition
* Technical way of saying what it is, they control the flow of the order of what commands/instructions are to be executed.

## Loops

* They will repeat something multiple times until you say no
* For loop – A useful way to have a more precise way to control of your loops
* While loop – When you want to keep repeating something as long as some condition is met before it loops again

## If

* They will only do something if some condition is met
* So, think of a power button, a device won’t do anything until someone turns on the power button

# Git Introduction

* It is a Version Control System (VCS)
  + Allows you to manage the changes/development made in a project
  + More specifically a Distributed Version Control System (DVCS)
  + Research CVCS
* It records any changes made to a project and maintains a history tree that contains the state of the project at a certain point (11.02)

## Pros

* Allows you to backtrack to previous version of your code if your current version is too unstable.
* Incredibly helpful when working on a team environment
* If local files get corrupted, you can just get the files stored in the cloud to restore it

## Cons

* You might have merge conflicts
  + Scenario in which one person worked with the same file as another person and git doesn’t know which modified version to use
* It is very confusing

## Repository

* This where your code is stored
* There is local repository is what is stored in your computer
* Remote repository is what is stored in the cloud

## CLI

* Git init – Creates an empty local git repository
* Git status – Checks which files are in staging and which aren’t
* Git add . – Will add every files to staging (Except for ignored files)
* Git commit – Records the changes to the repository
* Git branch – will create, delete, list branches for us
* Git -u origin main – pushes the changes you made to a remote repo you just created
* Git push – pushes the changes you made to a remote repo
* Git pull – pull the changes made in your remote repo