DAE Adult Full - Spring 2026 - Semester 1
Logic, Unix & Python

Darrell DeMakes 2/26/26

Logic 1: Foundations of Computational Thinking

Algorithms

Step-by-step procedures for solving problems — the blueprint before the build.

Flowcharts

Visual diagrams that map out an algorithm's logic, decisions, and flow paths.

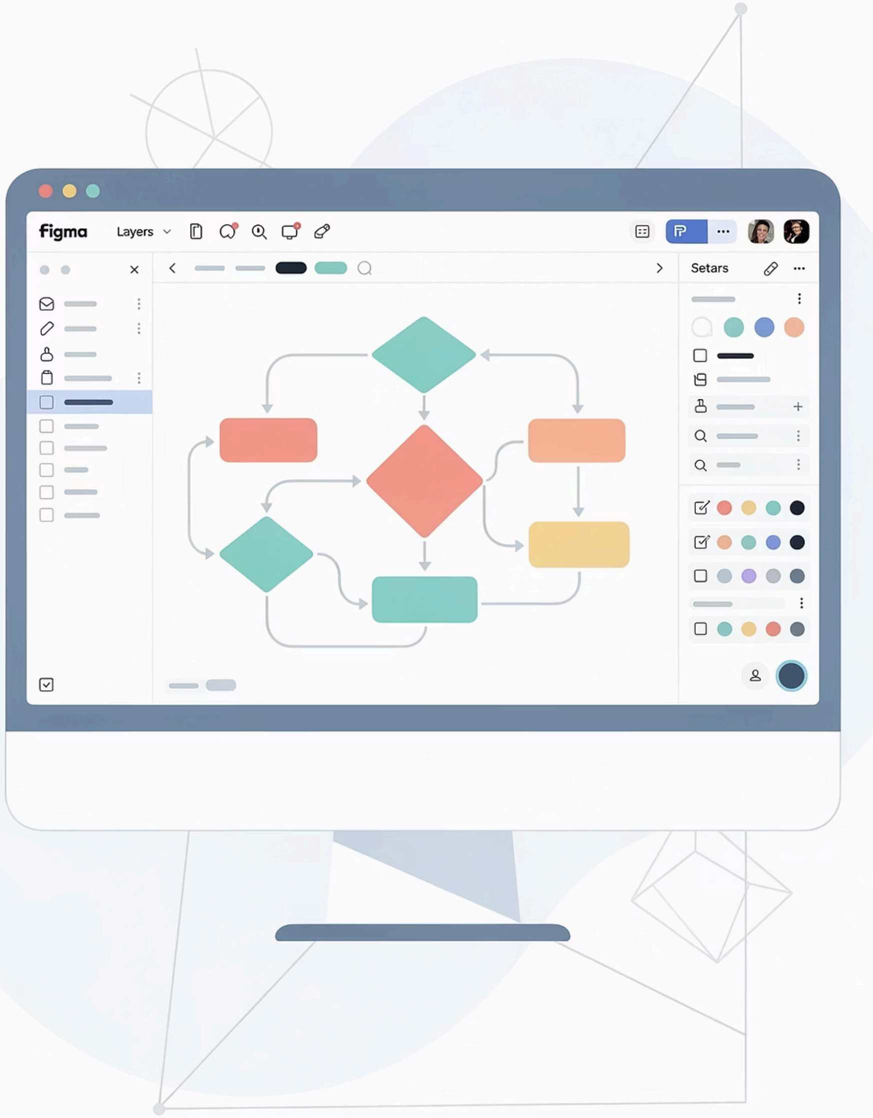
Boolean Logic

True/False values combined with operators — **and**, **or**, **not** — to form conditions.

Conditional Logic

Decision-making structures using **if-then** rules to branch program execution.

Algorithm & Flowchart
Figma



Unix 1: Command Line Essentials

Commands

`ls` list files · `pwd` print directory · `cat` view file · `touch` create file

Options

Modify command behavior with flags — e.g., `ls -l` returns a detailed, formatted file listing.

Arguments

Target a specific file or path — e.g., `cat filename.txt` reads and displays that file's contents.

Text Editors

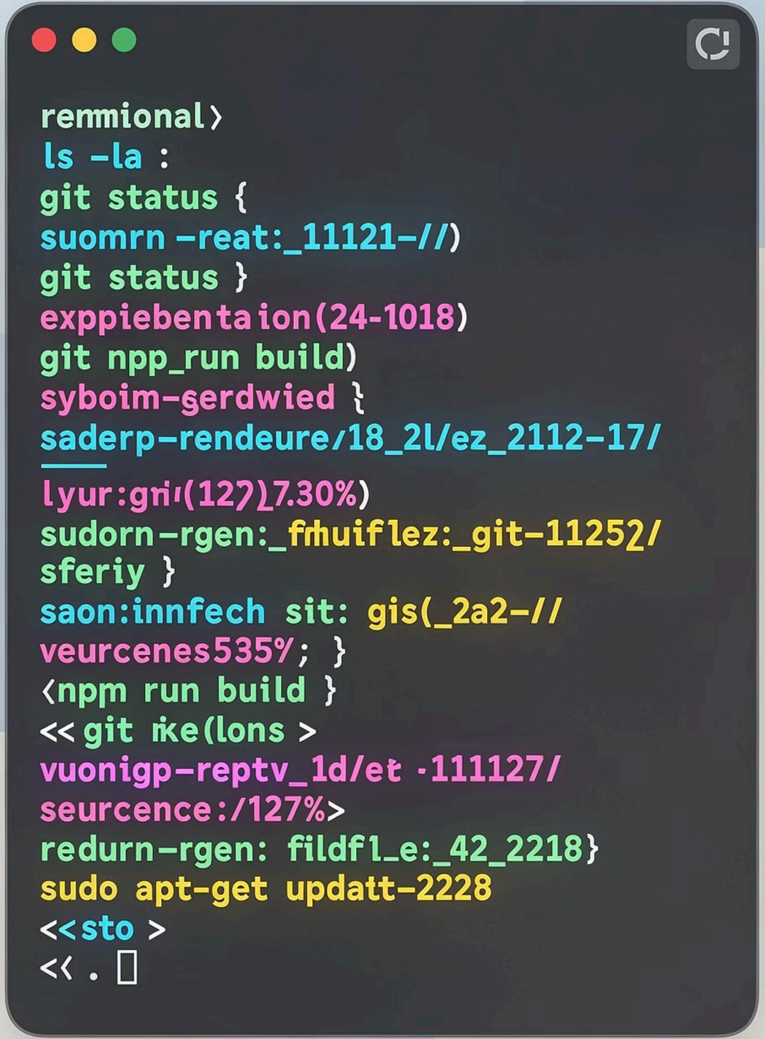
Create files with `touch`, then open and edit them directly in the terminal using `nano`.

Permissions

Control file access with **read**, **write**, and **execute** rights — view them with `ls -l`.

Unix

MacOS Terminal Zshell



```
remmional›
ls -la :
git status {
suomrn -reat:_11121-//)
git status }
exppiebenta ion(24-1018)
git npp_run build)
syboim-gerdwied }
saderp-rendeure/18_2l/ez_2112-17/
────
lyur:gni(127)∟7.30%)
sudorn-rgen:_fmuiflez:_git-11252/
sferiy }
saon:innfech sit: gis(_2a2-//
veurcenes535%; }
‹npm run build }
<< git ike(lons >
vuonigp-reptv_1d/et ·111127/
seurcence:/127%>
redurn-rgen: fildfl_e:_42_2218}
sudo apt-get updatt-2228
<<sto >
<‹ . ▯
```

Python 1: Programming Basics

Descriptive Variable Names

Write names that communicate intent — `student_age` beats `x` every time.

Three Data Types

Work with **numbers**, **strings**, and **booleans** — the core building blocks of any program.

Decision Structures

Use `if/else` to branch logic and respond dynamically to different conditions.

Loops

Automate repetition with `for` or `while` loops — let the computer do the work.

Functions

Encapsulate reusable logic in named blocks — cleaner code, easier debugging.

Lists

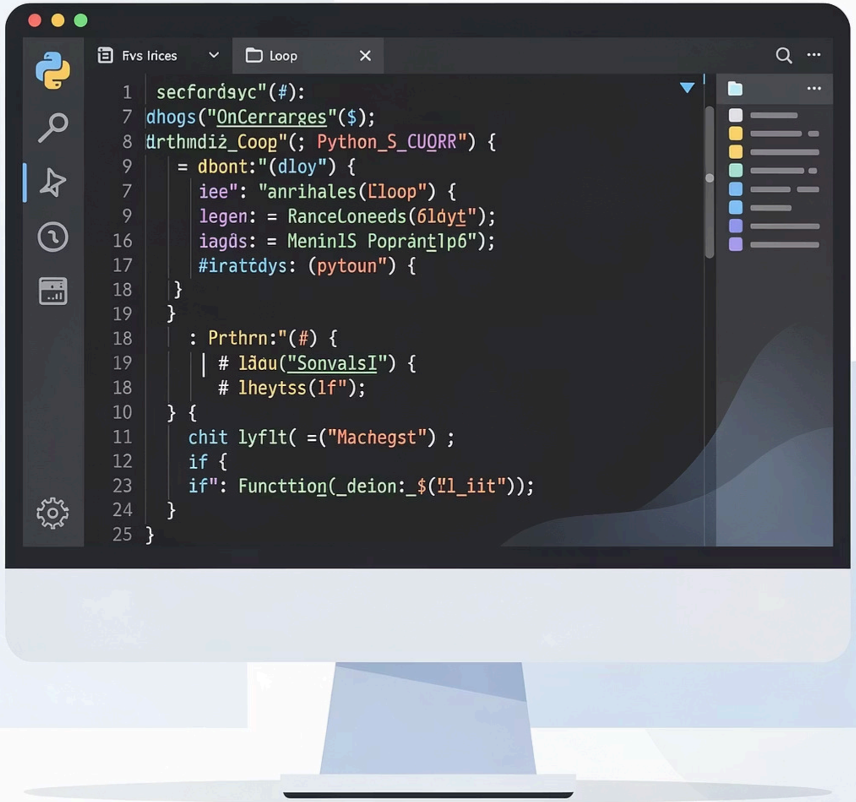
Store ordered sequences of data in a single variable — the foundation of data collections.

Comments

Document your code with `#` comments — readable code is professional code.

Python

VS Code & Github



```
 secfardayc"(#):
dhogs("OnCerrarges"($);
drthmdiż_Coop"(; Python_S_CUQRR") {
   = dbont:"(dloy") {
     iee": "anrihales(Lloop") {
     legen: = RanceLoneeds(6ldyt");
     iagds: = MeninlS Poprántlp6");
     #irattdys: (pytoun") {
  }
 }
    : Prthrn:"(#) {
      # lădu("SonvalsI") {
      # lheytss(lf");
 } {
    chit lyflt( =("Machegst") ;
    if {
    if": Functtion(_deion:_$(!l_iit"));
 }
}
```

Questions?