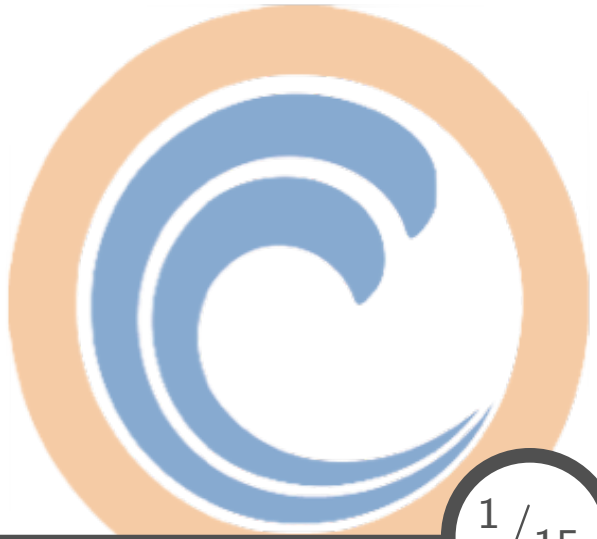
CS A131: Python Programming I

Nadia Ahmed

Orange Coast College, Computer Science

CS A131

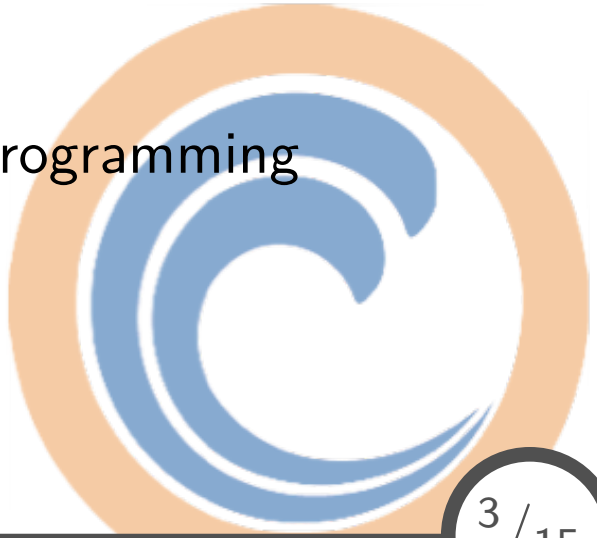


Overview

- Introduction to Computers
  - What is a computer?
  - What is programming?
- Getting Started
  - Opening a CS 50x Account
  - cs50.io (Cloud9 ide)
- Unix Environment
  - System commands
  - text editor

Introduction to Computers

- What is a computer?
  - Digital device capable of executing programs
    - performing computations
    - making logical decisions
- What is a program?
  - Set of instructions which process data
    - input data (from keyboard, mouse, or disk)
    - output data (to monitor, printer, disk)
- What is programming?
  - Creation of computer programs by use of a programming language.



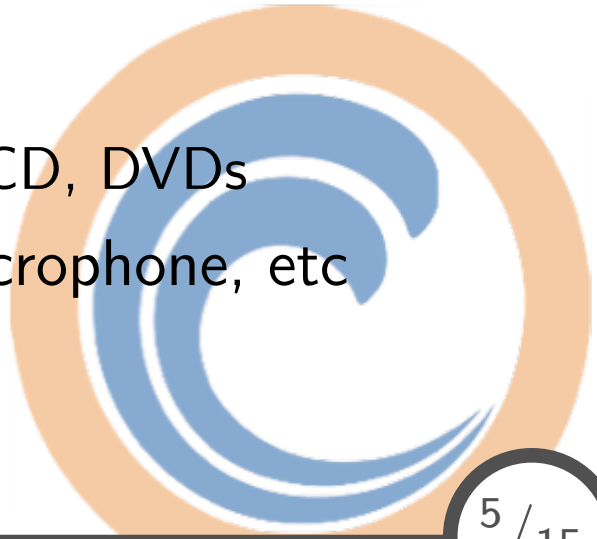
Introduction to Programming

- Categories of programming languages
  - Machine languages (1's and 0's)
  - Assembly languages (low-level CPU instructions)
  - High level languages (high-level CPU instructions)
- Translation of high-level languages
  - Interpreter (translation for each instruction)
  - Compiler (translation once for all code)
  - Hybrid (combination of the above)
- Types of programming languages
  - Functional (Lisp)
  - Structured (Pascal, C, Ada)
  - Object-oriented (C++, Java, Python)

Hardware

The physical devices the computer is made of

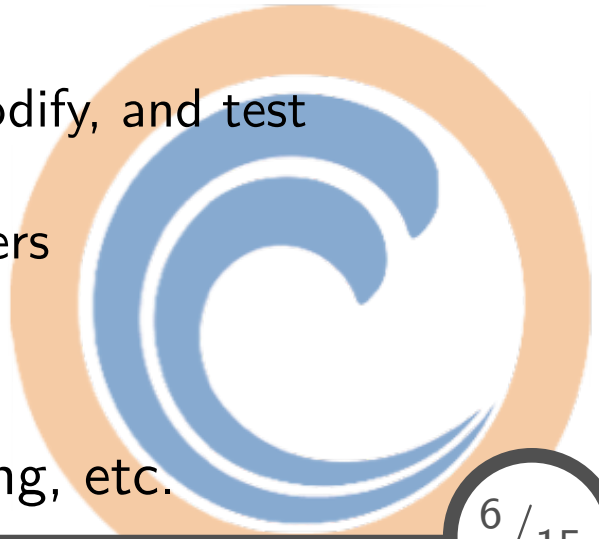
- The central processing unit (CPU): the part of the computer executing the program
- Main memory: where computer stores a program and relevant data while it is running
  - random-access memory (RAM)-volatile
- secondary storage devices
  - nonvolatile
  - disk drive
  - solid-state disk drive
  - external storage devices: floppy, flash drive, CD, DVDs
- input devices: keyboard, mouse, scanner, microphone, etc
- output devices: monitor, printer



Software

The programs that run on the computer

- System Software
  - Operating Systems
    - controls internal operations of hardware
    - manages devices connected to computer
    - saves and retrieves data from storage devices
    - allows other programs to run on computer
  - Utility Programs
    - performs a specialized task that enhances operation
    - example: virus scanners, file compression programs, and data backup
  - Software Development Tools
    - Programs that programmers use to create, modify, and test software
    - example: assemblers, compilers, and interpreters
- Application Software
  - programs for everyday tasks
  - example: email, web browsers, word processing, etc.



How Computers Store Data

All data in a computer is converted to sequences of 0s and 1s

- Byte: computer memory is divided into bytes
- 1 Byte = 8 bits
- Base conversions
  - binary
  - octal
  - hexadecimal
- Characters: ASCII
- Advanced number storage
  - twos complement
  - floating point notation

How a Program Works

CPU can only understand instructions in machine language

- Each instruction in a program is a command that tells the CPU to perform a specific operation
- Programs stored on secondary storage
- Fetch-Decode-Execute
  - Fetch: read instruction from memory into CPU register
  - Decode: decode instruction to determine operation
  - Execute: perform the operation using the arithmetic logic unit
- Keywords/Reserved words are words that make up high-level programming language (ex. `if`, `for`, `elif`)
- Operators
- Compiled languages
- Interpreted languages

Using Python

cs50X: Create an account!

Bookshelf Online: Starting Out | CS50: Introduction to Compute

Secure | https://www.edx.org/course/introduction-computer-science-harvardx-cs50x

edX Courses Programs Schools & Partners About Search: eecsNadia

Home > All Subjects > Computer Science > Introduction to Computer Science

Introduction to Computer Science

An introduction to the intellectual enterprises of computer science and the art of programming.

HARVARD UNIVERSITY

Self-Paced

Enroll Now

I would like to receive email from Harvard University and learn about other offerings related to Introduction to Computer Science.

About this course

154 Reviews 4.5/5

This is **CS50x**, Harvard University's introduction to the intellectual enterprises of computer science and the art of programming for majors and non-majors alike, with or without prior programming experience. An entry-level course taught by David J. Malan, **CS50x** teaches students how to think algorithmically and solve problems

See more

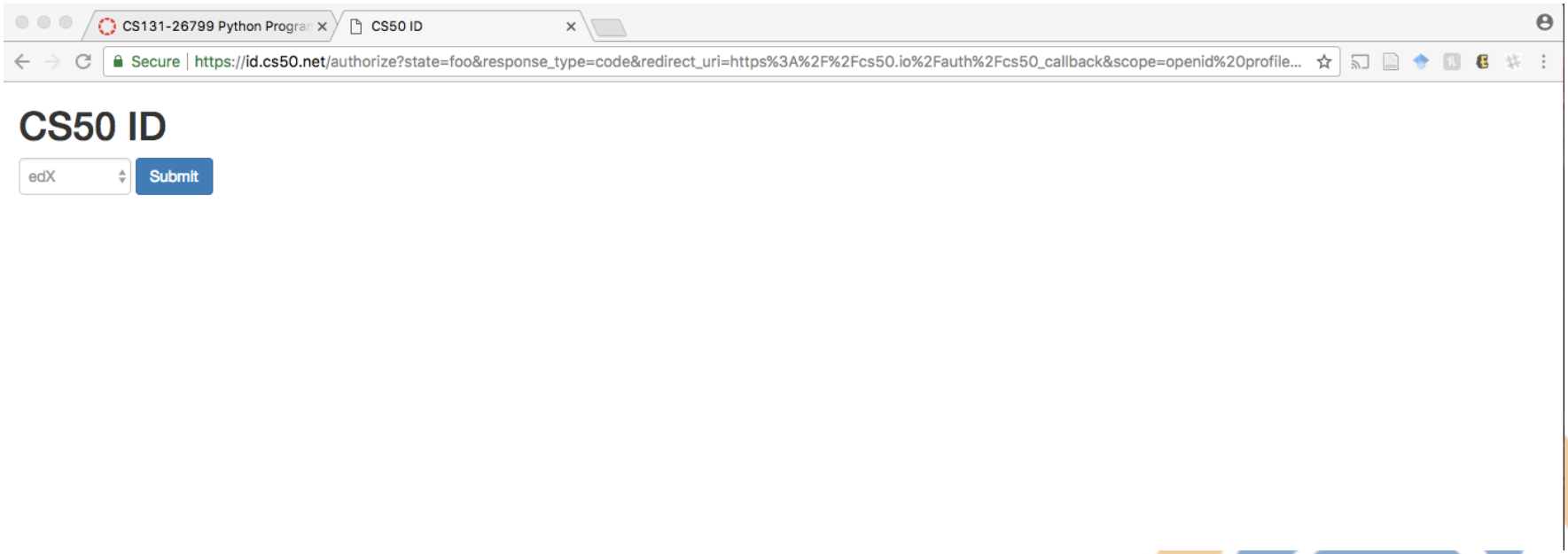
What you'll learn

Effort:	9 problem sets (10 to 20 hours each), 1 final project
Price:	FREE Add a Verified Certificate for $90
Institution:	HarvardX
Subject:	Computer Science

Using Python

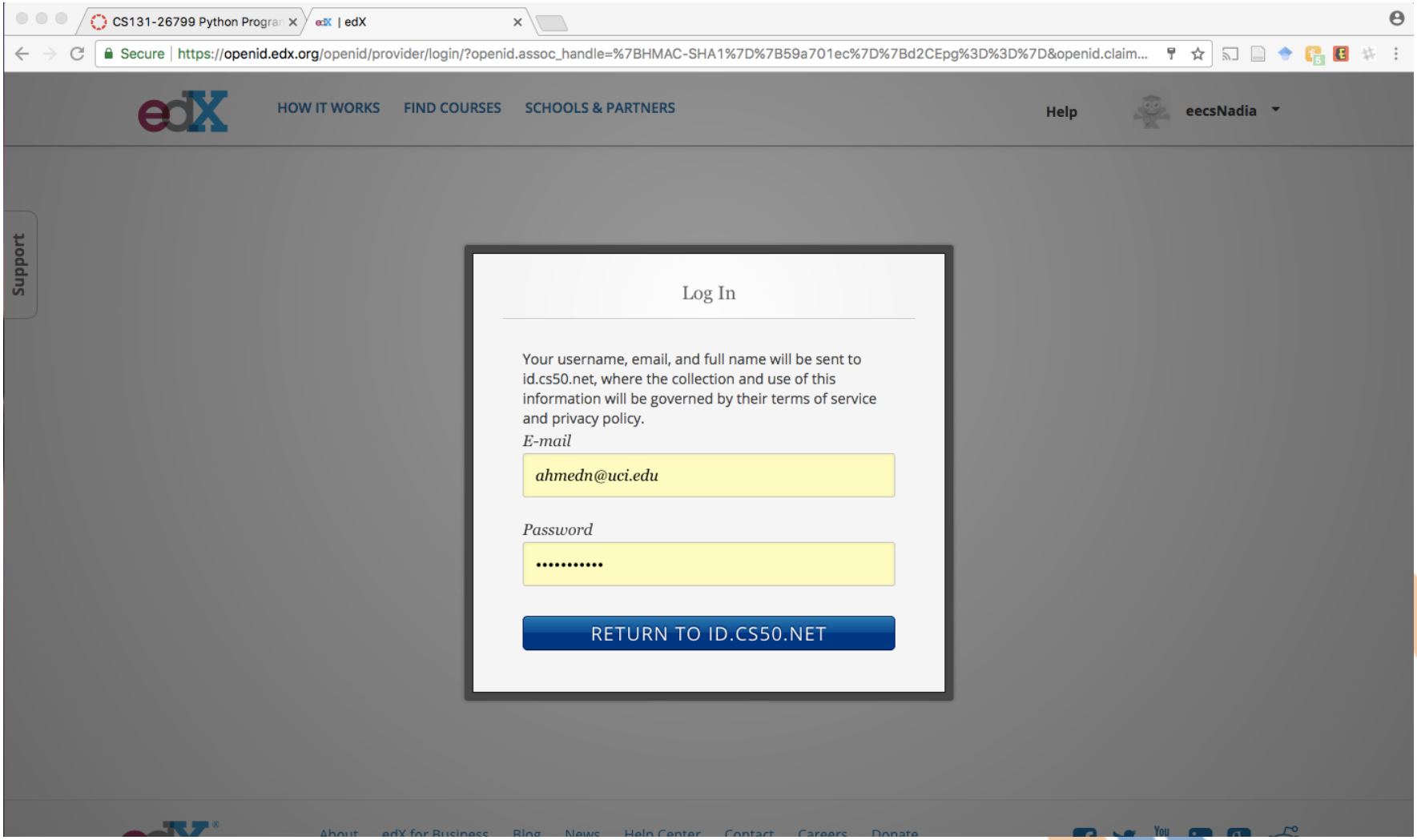
cs50.io Direct Link to Cloud9 IDE

IDE = integrated development environment



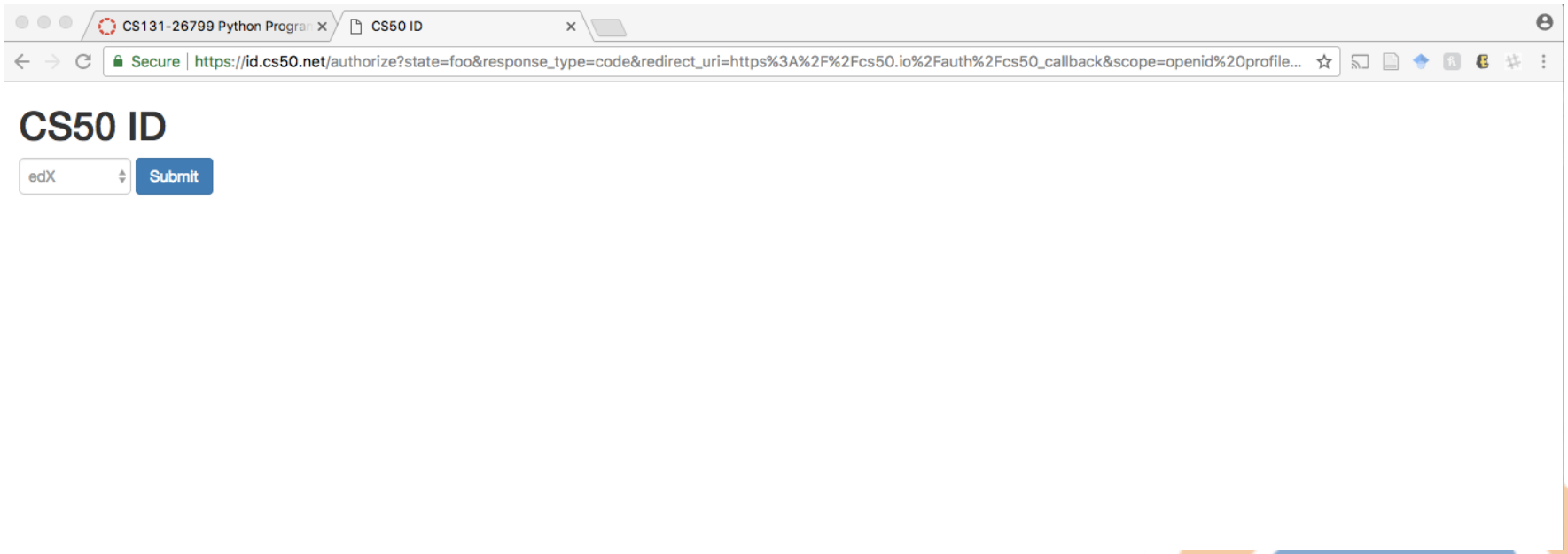
Using Python

cs50.io Direct Link to Cloud9 IDE



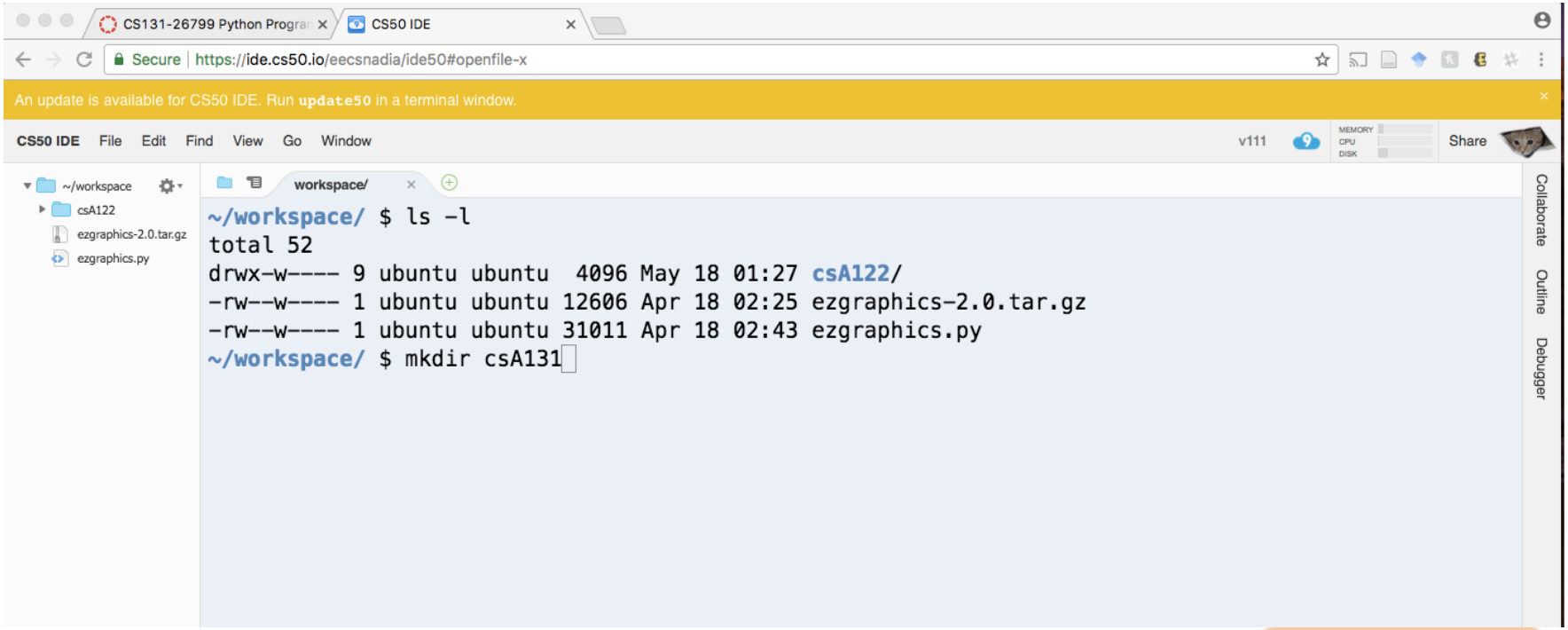
Using Python

cs50.io Direct Link to Cloud9 IDE



Using Python

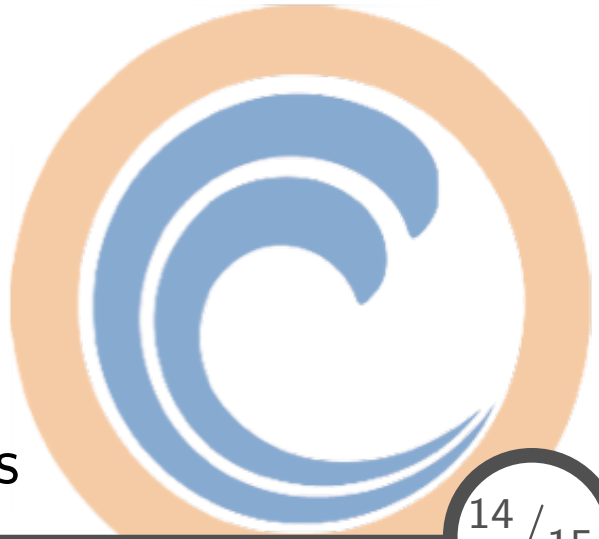
cs50.io Direct Link to Cloud9 IDE



- The left pane is similar to the graphical user interface you are used to in file manager.
- The right is the terminal where you use the Unix shell environment to manually manage your directories and files.

Linux System Environment

- Linux shell prints command prompt, awaiting input
- Type in system commands
  - `echo`: print a message
  - `date`: print the current date and time
  - `ls`: list the contents of the current directory
  - `cat`: list the contents of files
  - `man`: view manual pages for system commands
  - `more`: list the contents of files page by page
  - `pwd`: print the path to the current working directory
  - `mkdir`: create a new directory
  - `cd`: change the current directory
  - `cp`: copy a file
  - `mv`: rename and/or move a file
  - `rm`: remove (delete) a file
  - `rmdir`: remove (delete) a directory
- Refer to manual pages for help on commands



Linux System Environment

- Text editing
  - `vi`: standard Unix editor
  - `vim`: vi-improved (supports syntax highlighting)
  - `pico`: easy to use text editor
  - `emacs`: very powerful editor
  - many others...
- We will be using the `vi`/`vim` editor in the cs50 IDE.