Syllabus

Codigo Ecuador

Intro to Computer Programming with Python

Course Overview and Goals

The Codigo Ecuador Intro to Computers with Python course introduces students to programming and computer science. Students will learn to write programs in Python, scrape the web, and save their work in a database. They will further tune their intuition and shapen their logic and critical thinking skills for stronger performance in coding and any subject that requires problem solving or analysis. At the conclusion of the course, students will be empowered to use Python for software engineering, data science, and for automating their own daily tasks.

Programming Tools: Codigo Ecuador will provide each student with a Raspberry Pi. Each Pi will come with the official case and keyboard, along with a monitor. All required programming tools are included on this computer. Students will use Python 3 using the IDLE programming environment.

Website: <http://www.codigoecuador.org>

Prerequisites

This course assumes no prior knowledge of computer science or programming. The material will be presented with enough complexity to understand the subject, and no more.

Schedule

This course will meet once a week on Saturdays at 1:00pm, and will continue until 4:00pm.

Course Breakdown

Week 1:

-Raspberry Pi unboxing and introduction.

-Simple introduction to coding with the print statement

-Essential information/warnings: history of Python, differences between Python 2 and 3.

-Why we use Python 3 rather than Python 2.

-Initial discussion of exceptions using the print statement as an example.

-Where to enter Python code: the terminal, IDLE, and other development environments

-A list of words you cannot use when coding python (e.g. class, for, list, etc)

Week 2: Python’s built-in objects

-The pythonic view of objects: Everything is an object in Python, and all objects of the same type have the same properties.

-An object is a type of thing.

-Introduction to variables with Python’s built-in objects: String, Integer, and float.

- identifying and checking types.

-Operators: +, -, \*, /, and % for mathematical operations. Using the + operator to add integers and concatenate strings.

-String slicing and concatenation

-String indexing

-Converting between different types.

-Exceptions one might find with string, integer, and float operations.

-Basic arithmetic: how Python can be your calculator.

Week 3: Control statements

-if/else statement

-elif statement

-boolean datatype

-Introduction to external libraries: the date and datetime datatypes.

-How to load modules

Week 4: Data Structures I:

-List datatype

-Tuple datatype

-Distinction between lists and tuples: how to tell them apart and how they are different.

-Accessing, appending, slicing, and changing list items.

-Introduction to immutability with Tuples.

-Introduction to for loops

Week 5: Data Structures II:

-Dictionary datatype

-Set datatype

-Using for loops to iterate over dictionaries and sets.

-Using while loops when the number of iterations is uncertain.

-Dictionary example: list of students in the class and their ages.

-Set example: finding duplicates in a bag of groceries.

-Using JSON module to save dictionaries to the disk.

Week 6: Functions

-Functions that do not take an argument

-Functions that take an argument

-Local versus global variables; overwriting global variables

-Return statements

-Introduction to the typing library, with optional type annotation. Why you should use type annotation even though you don’t have to.

-Public versus private functions

Week 7: Working with files

-Creating, reading and writing files.

-Storing text on separate lines of a file.

-Loading saved text from a file.

-using the “with” statement to manage file handling

-Using pickle to save lists.

-Reading and writing csv data to a file.

-the FileNotFoundError and OSError exceptions

-**Avoiding Terminalphobia: how to enter and execute code on the terminal.**

Week 8: Data Structures III:

-String methods

-Splitting and rejoining strings using the split and join functions.

-Slicing strings with indexing.

-Nesting data structures within one another.

-Assignment: saving and loading strings as JSON objects.

Week 9: Custom Objects (a preview)

-introduction to namedtuple and NamedTuple types (the simplest type of custom class)

-creating an object with elements of different types (e.g. a NamedTuple with a string and an integer).

-Introduction to the Python standard library.

Week 10: Exceptions

-Outline of exceptions in Python.

-Common types of exceptions: ValueError, TypeError, IndexError

-Catching exceptions with the try and except statements

Week 11: Numbers and Algorithms

-Working with numbers

-math module

-Sorting algorithms: Bubble sort, merge sort, quick sort

Week 12: Working with the Operating System

-os module: getting and changing the current working directory, checking to see if a file is present

Week 13: Object oriented Programming

-The namedtuple class review

-The NamedTuple class (typed version) using the typing library

-Introduction to regular python classes

Week 14: Object oriented programming II:

-

Week 15:

Week 16:

Week 17:

Week 18:

Week 19:

Week 20:

Week 24: Introduction to SQL.

Week 25: Introduction to SQLite module in python. Create a simple database application tracking class attendance with the sqlite module.

Week 26:

Week 28: SQL, SQLalchemy and the CMD module. Students will create a simple application using customized classes. Goal: to make use of the ordinary Python standary library functions as “glue” to connect the external libraries, which do most of the heavy lifting.

Week 42: Final project working period

Week 43: Final Project working period

Week 44: Final exam/project TBD