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. Any knowledge of programming or other disciplines is always helpful but never required

Schedule

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

Course Breakdown

**Part I: Introduction to Python**

Week 1:

-Raspberry Pi unboxing and introduction.

-Raspbian operating system

-Simple introduction to coding with the print statement.

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

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

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

-Starting python using the terminal

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

-How to write comments in your code

Exercise:

Week 2: Python’s built-in objects

Topics:

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

-String, int, float datatypes

-Basic arithmetic: how python can be your calculator

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

-Identifying and fixing basic errors (e.g. adding a string to an integer)

-identifying and checking types.

-Basic string methods: slicing, indexing, and concatenation

-Converting between different types. E.g. the difference between ‘4’ and 4 in Python.

-introduction to libraries with random.

-Assignment and comparison statements

Assignment: Guessing game with random numbers

Week 3: Control statements

-if/else statement

-elif statement

-boolean datatype

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

-How to import modules, the import statement

-Creating your own modules and importing them

Assignment: Restaurant order taker

Week 4: Data Structures I:

-List datatype

-Tuple datatype (and immutability)

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

-While and for loops

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

Assignment: Ice cream stand

Week 5: Data Structures II:

-Dictionary datatype

-Set datatype; basic set methods including add, intersection, and difference

-Using for and while loops to iterate over dictionaries and sets.

-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 and load them.

Week 6: Functions

-Concept of namespaces and scope in Python

-Local versus global variables; overwriting global variables and how this is not recommended

-Functions that take/do not take an argument

-Return statements

-Public versus private functions, anonymous functions

-Creating placeholders with the “pass” statement

-intro to the if \_\_name\_\_ == ‘\_\_main\_\_’: statement, used when a file is the main module.

-Assignment: Use a function to calculate the total price of items on a receipt.

-Assignment: Use loops and functions with the turtle library

- Week 8: Data Structures III:

-Advanced String methods: Splitting and rejoining strings using the split and join functions.

-Nesting data structures within one another.

- Using the map function to apply another function over a list.

-advanced String methods: including splitting and joining strings, lower/uppercase

Week 9: Exceptions

-Outline of exceptions in Python.

-Common types of exceptions: ValueError, TypeError, IndexError

-Catching exceptions with the try and except statements

-The finally statement

-Using the “assert” statement to catch exceptions

-Raising exceptions yourself

Week 9: 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.

-csv module for loading data; read and write data to a csv file

-the FileNotFoundError and OSError exceptions

-Copying, deleting, and executing files.

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

Week 10: Numbers, Algorithms, and more Functions

-Working with numbers

-math module, fractions module

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

-Functions taking variable number of arguments

-Public versus private functions

-Generator functions and Iterators

Week 11: Introduction to Numpy

Topics:

-the np.array datatype

-using basic python structures to load, manipulate, and save data as NamedTuples

-performing numerical analysis on data

-data security (don’t load raw data from an unfamiliar source)

-using requests to get data from the web

Assignment:

Week 12: Custom Objects (a preview)

-Introduction to the Python standard library

-re module: regular expressions

-collections module: counters, ordereddicts, and queues. Overriding custom types.

-introduction to third party libraries, pip, and downloading libraries.

-getting help (the python help function) on the various object types

-using \_\_dict\_\_ to get the attributes of an object in dictionary form

Week 13: Object oriented Programming I

-Introduction to custom objects with previous examples

-The namedtuple class review

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

-Introduction to regular python classes, \_\_init\_\_ function

-Types of function with default \_\_init\_\_ e.g. NamedTuple

-Introduction to instance methods, and the word self.

Week 14: Object oriented programming II:

-Public and private methods

-Class methods and the word cls, static methods, instance methods

-encapsulation, inheritance, polymorphism and “duck typing”

-overloading the \_\_repr\_\_, \_\_add\_\_, \_\_hash\_\_, and \_\_eq\_\_ methods.

-class properties

-Storing custom objects with pickle and shelve

Assignment: bus ticket application.

Week 15: Advanced custom objects

-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).

**Part II: Putting Python to Work**

Weeks 15-16: More Numpy

-The np.array type

-Basic math operations with numpy

-Making charts with matplotlib

Week 16-17: Pandas

-The dataframe type

-Data visualization with matplotlib and pandas

Exercise: make a bar chart with the existin

Weeks 18-20: Creating command-line games

-The cmd and pygame modules

-Creating a command line interface with the cmd module.

-Defining commands and leaving help for the user.

-Handling an empty command.

-Commands that take arguments

**Part II: Advanced Python Topics**

Week 21-22: New features of the language

-Typing library: optional type notations for more readable code

Week 23: Object-Oriented Program design

-The importance of good design

-Three tier model: database, business, and presentation tiers

-Identifying data to be stored in objects

-Example: separating a name and address into its components

Assignment: Choose something (it can be almost anything at all) and make a Python class about it. Store that class in a pickle object and then retrieve it.

-Version control

Week 24: Introduction to SQL.

What is SQL?

Creating tables

Inserting data

Reading data

Deleting data

SQL datatypes

Week 25: Introduction to SQLite module in python

-importing sqlite3

-the connection object

-the cursor object

-creating a simple db api in Python

Assignment: Create a database project:

Example Project 1: Creating a book lending library application.

Example Project 2: Create a simple database application tracking class attendance with the sqlite module.

Week 26-27: More SQLite

Project: Continue with either book lending or class attendance application

Week 28: SQL without SQLite: introduction to SQLalchemy

Project: Replace database project backend with SQLalchemy backend

Weeks 30-33: Introduction to web programming

Topics:

HTML and CSS

Web scraping with BeautifulSoup

Flask

APIs

Assignments:

-Help with your homework: scrape data and save it in a sqlite database or csv file.

-APIs: scrape data from a publicly available site, and save to a JSON file.

Weeks 34-36: Maxing out the Raspberry Pi

-Introduction to electronics with raspberry pi

Assignment: Creative project with Raspberry Pi and/or Arduino, eg soil sensor, moisture detect, self-driving robot

Weeks 37-39: Introduction to Javascript with Raspberry Pi

-Node.js for communicating with peripherals

Assignment: Controlling lights and motors with Javascript

Week 40: Final Project topic selection – The final project is an assignment consisting of one or more of the topics we have discussed in class. The class will be divided into groups of 3-4 students, with each group proposing a project.

Week 41: Final project working period

Week 42: Final project working period

Week 43: Final Project testing session

Week 44: Final exam/project TBD