TechForNonTechies

Prerequisites

- No background in programming required
- No software needs to be installed. All coding is done using repl.it.
- Programming will be done in Python

Goals

- A whistle stop tour of the life a program
- Give the absolute basics of programming
- Each class broken up into two sections:
    - Background
    - Coding tutorial

- At the end you should be able to call an API and do something with the response
- Teach programming concepts while teaching standard practices within Mastercard

Week 1

Note: Look at Harvard CS50 for sample course outline

The Basics

What is a computer?

What is an operating system?

A brief history and the Pirates of Silicon Valley

Introduction to the early OS wars involving Xerox, Microsoft, Apple, IBM, etc.

Components of an OS

- Kernel

- Interrupts
- Memory management
- Thread management
- File organization
    - Command line exercises to navigate the directory tree
- User interface

What is a program?

Programming languages

- Common characteristics of programming languages
    - We can focus on Python language features since we're going to be doing exercises
    - Compare python with other languages (e.g. C++, Java, Ruby)
    - Explain what domains each language might be used for (C++ embedded systems, Java microservices, Javascript front end web pages, Python + data science / ML)
    - Exerices related to this?
        - Match a code snippet with functionality / feature
        - Identify the language feature (e.g. For Loop, function definition)
- Type systems
- Compilers

Week 1

Variables

Some paragraph defining this

How to declare a variable?

Some example

Some other section

- Variables
    - Types

        - Primitives
            - What is a primitive?
                - Has to do with how the data is retrieved from memory
                - Basic building blocks of all applications
                - Create new data types (objects) out of these primitives
            - Integer
            - Boolean
            - Float
            - String
        - String
            - Some languages consider String a primitive, some don't
        - Collections
            - List
            - Array
            - Dictionary
            - Tuple
            - Set

- Operators
    - Mathematical
        - +
        - -
        - *
        - /
        - **
        - %

Week 2

- Operators
    - Comparison
        - ==
        - !=
        - >
        - >=
        - <
        - <=

    - Logical
        - not
        - is
        - is not

- Control structures
    - Conditionals
        - if
        - elif
        - else

    - Loops
        - While
        - For-in

Week 3

- Objects
    - Functions

Week 4

- HTTP requests / responses
    - Calling the Mastercard API
    - Final project

Exercises

See appendix

Quizzes

- Simple quizes to reinforce the information presented
- Quiz before and after each session?

"Hello world"

Input and Output

Implement in python, but show implementations in other languages e.g. C, Javascript, R, Erlang, to cover different programming paradigms, OO, Functional, Logical, Procedural

Iterate over a list

Week 2

Development Environments

- Dependency management
    - Python - pip / PyPI
    - NodeJS - npm
    - Java, Scala, Kotlin - Maven, Gradle
    - Ruby - RubyGems
- Why use dependency management?
- What is git and version control?
-

Where is code stored?

Frameworks

- What is a library?
    - Spring Boot
        - Goal to reduce boilerplate or ramp-up of a project
- Packaging libraries/frameworks
- Distributing libraries/frameworks
    - Artifactory

Exercises

- Walkthrough of a sample Spring Boot project
    - Look at pom.xml OR build.gradle
    - Look at src/main/java directory
    - Where are classes saved
    - Building an application

Types and variables

Week 3

Running software

- Pivotal Cloud Foundry
  - Sample push application
  - Pipelines
- Jenkins
- CI/CD

The cloud

Exercises

Dictionaries

- Iterate over values in a dictionary

Week 4

How does the internet work?

- Show the basics of an HTTP request?
- What are microservices?
- What is an API?

Exercises

Call an API

- Make a call to the DarkSky API to retrieve the weather for Dublin
- Do something with the result (e.g. get the max wind speed for the day)
- **Advanced**: call a Mastercard API
  - Requires using the OAuth signer

Summary of learning

Goals

- Graduates should be able to identify terminology
  - API
  - JSON
  - Keys
  - XML
  - Git
  - Maven, Gradle
  - Microservices

- Relate what they learned in the course to their daily work lives
- Capstone project
  - DarkSky API integration

Tools

- Repl.it
- Jupyter?
- Documentation!

Exercises

Week 1

Variables

Exercise 1: Variables

Input/Output

Exercise 2: Print "Hello, <your name>"

Code

```
print('Please tell me your name')
name = input()
print(f'Hello {name}')
```