Lecture #1: Where it starts!

By Jamal Bouajjaj

2023–10–02

For University of New Haven's Fall 2023 CSCIxx51 Course

First Slide

Welcome to Script Programming/Python

Syllabus

The Syllabus

Placeholder

Installation

Intro to Python

What is it?

Python is a dynamially-typed, high-level, intepreted, script-able language

High-Level

Python does a lot of heavy lifting for you!

Memory managment for example is not heard of in Python-land, it allocates the memory it needs for you and collects your garbage.

Generally, it's easier to do tasks in Python in C/C++ for example.

All this goodness has a sacrifice: speed.

Intepreted?

Running a Python line does not require what is before it to be compiled down.

Sort of just runs the application. Exectues your application line by line as it comes.

Intepreted Caveat

Because Python is intepreted and dynamially typed, it will not check some errors ahead of time like undeclared variables

```
x = input("Please don't type 0 ")
if x == "0":
    fprint("How could you")
else:
    print("Yeah, we're happy!")
```

Python must understand what it’s given, but it makes no assumption of previous declarations.

Implementation

Just like C, Python is a language specification. Like the different compilers, there are different Python intepreters.

The most popular and widely used, CPython, is the one that you just downloaded. It's what people just refer to as "Python". It's also the reference implementation.

There are several other intepreters, some of which are:

- Jython: A Java implementation with nice integration with Java bytecode
- PyPy: Implementation that uses a Just-In-Time compiler similar to Java, is faster
- RustPython: A Rust implementation

Syntax

We'll learn as we go along for the most part, but there are a couple of things I would like to note:

- It enforces whitespace indentation, so indenting is required. No more cursed one-lined obfuscated JS
- The whitespace can be space or tabs, but must be consistent
- The recommended is 4 spaces
- While the intepreter doesn't force you, there is a consensus on naming convensions (PEP 8). I personally will follow PEP 8

Variable Types

There are some variable types, some of which we'll cover later in the course. Here are some basic ones:

- integers
- boolean: True or False
- floating point numbers
- complex numbers: yes Python has native support for those
- strings
- lists
- etc (not a type)

End

The end