Python: A Primer

A HMS Research Computing Training Session

Alex Truong

alex_truong@hms.harvard.edu

https://rc.hms.harvard.edu/

Introduction

Why Python?

Python is...

simple - resembles plain English
easy - no need to declare (in most cases), memory management
clean - whitespace-formatted for visibility
interpreted - good for developing, bad for performance (more on this later)

Interpreted	Compiled
Rapid prototyping	Faster Performance

Accessing Python on O2

Logging into O2

Open a terminal and ssh into o2.hms.harvard.edu

```
$ ssh rc_training01@o2.hms.harvard.edu
rc_training01@o2.hms.harvard.edu's password:

rc_training01@login01:~$
```

Accessing Python on O2

```
$ module avail python
No modules found!
Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all po
any of the "keys".

$ module spider python
   Versions:
      python/2.7.12
      python/3.6.0
```

Let's Learn Python!

Syntax

Statements are terminated by newlines (e.g. enter key)

Examples:

```
>>> a = 1
>>> print("hello world")
hello world
```

Nuances: Quotations

In general, double and single quotes are interchangeable, both for printing and argument passing:

```
>>> print("hello world")
hello world
>>> print('hello world')
hello world

>>> str = 'abcdefg'
>>> str1 = "abcdefg"
>>> str == str1
True
```

A brief Introduction to Scripting

Up until now, we've mostly been playing inside the interpreter. Here, we'll briefly go over what is required to write a proper Python program.

To start:

Strictly speaking, all you need for a Python program is a text file with the shebang line on top. Recall:

```
#!/usr/bin/env python3
```

This line indicates to the computer that this is a python program, and it should look in this location to execute. Similar shebangs may look like:

```
#!/usr/bin/python
#!/bin/bash
#!/usr/bin/perl
etc.
```

The shebang is telling the computer to look in the specified directory for the proper method of execution.

Why use env?