

Lesson 10.1

Python Scripting 1

Learning Objectives

Understand what is an interpreted programming language and that Python is an interpreted language

Understand basic Python programming ideas including saving variables, writing lists, for loops and conditional statements.

Interpreted vs Compiled Programming Languages

Interpreted

Definition: Source code is compiled when executed by an interpreter

Advantages: Development of new code and changes to existing code are generally faster than with compiled code. Prototyping code is typically simpler. Code is portable to any system.

Disadvantages: The execution of code is generally slower than compiled languages because it is 'compiled' on the spot each time you run the code.

Compiled

Definition: Source code is translated into machine code to be executed later

Advantages: Once compiled, the execution of code is generally faster than interpreted languages. This is important for heavy resource applications

Disadvantages: Speed of development is typically slower since each iteration must be compiled. Compiled software can only be run on the system it was compiled on.

Python is an Interpreted Language

- High level, general purpose programming language
- Object oriented
- Design philosophy emphasizes code readability
- Named after the sketch comedy troupe “Monty Python”



The Python Interpreter

Typing 'python3' on the command line will start the Python interpreter

This gives a spot to try some commands.

To exit type: `exit()`

```
~]$ python3
Python 3.7.7 (default, Apr 14 2020, 20:44:16)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 4+4
8
>>> 4*4
16
>>> 4/4
1.0
>>> 4-4
0
>>> a=4+4
>>> a
8
>>> exit()
~]$
```

“Hello_world.py” Python Program

It is usually better to write Python scripts so that your code is saved. The following instructions are to write a first Python Program:

- 1) Once logged into a cluster, open a text editor (eg. vi or emacs)
- 2) Type the following line: `print(“Hello World!”)`
- 3) Save and exit the text editor
- 4) Type: `python3 Hello_world.py`

“Hello World!” should be printed to the screen.

Note how the adding the name of a Python script after the python executable runs the program.

Lists

Building vectors and matrices in Python involves a structure called a List. Lists are enclosed by square brackets and can be made of any data type:

Vector: `a = [0,'t',2,'Y',4]`

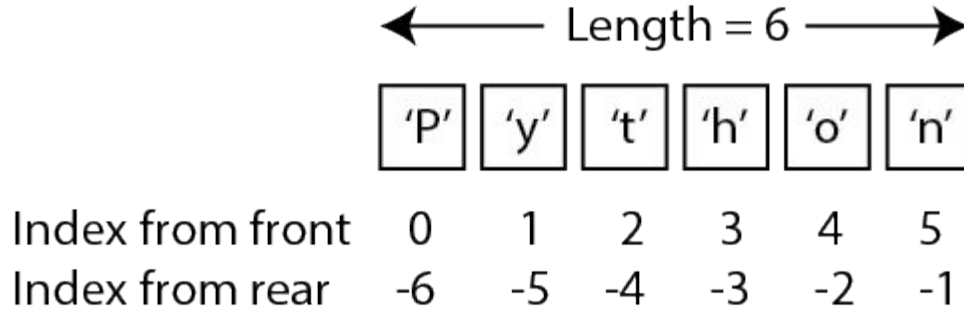
Matrix: `b = [[1,2,3],[4,5,6]]`

Strings are just a list of characters.

`c = 'Python'`

Calling List Elements

Python using an indexing scheme to call elements from a list. For example, the individual letters from the string `c='Python'` can be called according to:



So, `c[0]` and `c[-6]` will return the letter 'P'

For Loops

For loops are used to iterate through a list and perform an operation on each element. For example, to print each letter of `c='Python'` using a for loop:

for letter in c:

```
    print(letter)
```

Note two things:

- 1) The syntax of the for loop is 'for element in list' so each element of the list will take a turn as element
- 2) The indentation. Python uses significant whitespace. The end of the for loop occurs when the indented block ends.

Conditional Statements

If/else statements also use significant whitespace. Assume that `a=2`:

if `a > 1`:

```
    print('a is greater than 1')
```

elif `a < 1`:

```
    print('a is less than 1')
```

else:

```
    print('a is equal to 1')
```

Summary

- Python is an interpreted language that lends itself to quick prototyping of code ideally to perform non-resource intensive operations
- Variables can be saved to lists where indexing starts at 0. Indexing also starts at the end of the list with -1
- Python uses significant whitespace to indicate when for loops and conditional statements are in effect.