Computational Science and Scientific Computing Workshop

Data Programming - Python as a scientific computing tool

Elliot S. Menkah, Ph.D
Daniella N. Apeadu

National Institute for Mathematical Sciences
Kwame Nkrumah University of Science and Technology

October 13, 2025

Python

Why Python?

- It is interpreted and NOT compiled
  - E.g. of Compile languages are C/C++, FORTRAN, etc.
- It's a dynamically-typed language.
- It can be used interactively.
- Syntax is simple, elegant and easily readable.
- Free and open source.
- It's powerful due to its ecosystem of libraries.

Python

Why Python?

- It is interpreted and NOT compiled
    - E.g. of Compile languages are C/C++, FORTRAN, etc.
- It's a dynamically-typed language.
- It can be used interactively.
- Syntax is simple, elegant and easily readable.
- Free and open source.
- It's powerful due to its ecosystem of libraries.

Python

Why Python?

- It is interpreted and NOT compiled
  - E.g. of Compile languages are C/C++, FORTRAN, etc.
- It's a dynamically-typed language.
- It can be used interactively.
- Syntax is simple, elegant and easily readable.
- Free and open source.
- It's powerful due to its ecosystem of libraries.

Python is versatile.

- Download information from a web page.
- Manipulate tests to extract and create information.
- Animate a world in 3D.
- Process huge data sets.
- Make publication-quality graphics.

Which version of Python should I use?

- Currently two(2) versions: 2.7 and 3.7
  - Some packages still work **only** with 2.7
- Versions: 2.7 is deprecated
- Recommend you use version 3.7

Which version of Python should I use?

- Currently two(2) versions: 2.7 and 3.7
  - Some packages still work **only** with 2.7
- Versions: 2.7 is deprecated
- Recommend you use version 3.7

Which version of Python should I use?

- Currently two(2) versions: 2.7 and 3.7
  - Some packages still work **only** with 2.7
- Versions: 2.7 is deprecated
- Recommend you use version 3.7

Which version of Python should I use?

- Currently two(2) versions: 2.7 and 3.7
  - Some packages still work **only** with 2.7
- Versions: 2.7 is deprecated
- Recommend you use version 3.7

ipython Notebook

Why ipython Notebook?

- Gives us a computational notebook with lots of inclusions

- Source code in python and other languages
- Rich text
- Equations written in Latex
- Ready output of results
- Graphics
- Multimedia

ipython Notebook

Why ipython Notebook?

- Gives us a computational notebook with lots of inclusions

- Source code in python and other languages
- Rich text
- Equations written in Latex
- Ready output of results
- Graphics
- Multimedia

ipython Notebook

Why ipython Notebook?

- Gives us a computational notebook with lots of inclusions

- Source code in python and other languages
- Rich text
- Equations written in Latex
- Ready output of results
- Graphics
- Multimedia

ipython Notebook

Why ipython Notebook?

- Gives us a computational notebook with lots of inclusions

- Source code in python and other languages
- Rich text
- Equations written in Latex
- Ready output of results
- Graphics
- Multimedia

ipython Notebook

Why ipython Notebook?

- Gives us a computational notebook with lots of inclusions

- Source code in python and other languages
- Rich text
- Equations written in Latex
- Ready output of results
- Graphics
- Multimedia

ipython Notebook

Why ipython Notebook?

- Gives us a computational notebook with lots of inclusions

- Source code in python and other languages
- Rich text
- Equations written in Latex
- Ready output of results
- Graphics
- Multimedia

ipython Notebook

Why ipython Notebook?

- Gives us a computational notebook with lots of inclusions

- Source code in python and other languages
- Rich text
- Equations written in Latex
- Ready output of results
- Graphics
- Multimedia

Installing Python.

- Alternate: package manager '-apt-get' on Linux or 'brew' on Mac to install python
- Anaconda

How do I run python?

```
#!/bin/(bash or zsh)
$ python
Python 3.6.7 — packaged by conda-forge — (default, Nov 6 2019, 16:03:31)
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This is mainly good for running scripts.

```
#!/bin/bash/zsh
$ ipython
Python 3.6.7 — packaged by conda-forge — (default, Nov 6 2019, 16:03:31)
Type 'copyright', 'credits' or 'license' for more information.
IPython 7.10.2 – An enhanced Interactive Python. Type '?' for help.
In [1]
```

Anaconda - Conda virtual environment

- **exclusive environment**
- **reinstall anaconda**
- **package dependencies resolution**

Download anaconda via the link: https://www.anaconda.com/distribution/ and download the installer for your respective OS [Linux , mac , windows]

Create an environment:

```
conda create <envname>
Eg.
conda create scim561

```

Connect to environment

```
conda activate scim561

```

Installing packages into an environment

```
conda install <package>
Eg.
conda install matplotlib

```

Deactivate/disconnect from present working environment:

Python Basics

print function, variables, operators

Interpreter - strings and print() function

Print functions and strings:

```
>>> print("Hello World")
Hello World

```

Use double outer quotes (” ”) over single outer quotes (’ ’)

```
>>> print(’We\’re here’)
We’re here

```

to avoid complications.

```
>>> print("We’re here")
We’re here

```

Interpreter - Variable assignment and Data types

Variables take on the data type of the values being assigned to them.

```
>>> var0 = "hello"
>>> var1 = 7
>>> var2 = 5.2
>>> var3 = True

```

String Variable:

```
>>> print(var0)
hello
>>> type(var0)
<type ’str’>

```

Integer Variable:

```
>>> print(var1)
7
>>> type(var1)
<type ’int’>


```

Interpreter - Variable assignment and Data types

Variables take on the data type of the values being assigned to them.

Floating point Variable:

```
>>> print(var2)
5.2
>>> type(var2)
<type 'float'>
```

Boolean Variable:

```
>>> print(var3)
True
>>> type(var3)
<type 'bool'>
```

Python Operators

Special symbols that carry out arithmetic or logical computation.

Arithmetic Operators

```
+ addition
- substraction
* multiplication
/ division
\% Modulos
// Floor division
** Exponential

```

Logical Operators

```
= assignment operator
== Equal to
< less than
> greater
<= less than or equiv.
>= greater or equiv.
and
or
not

```

Exercises 1

Given an initial velocity, $\mathbf{u}$, as 10.2 ms^{-1}, an acceleration, $\mathbf{a}$, of 10.01 ms^2 and a time, $\mathbf{t}$, of 4 seconds, using the python programming language, write a code to compute the final velocity of a moving particle with the following formulation
v = u + at.