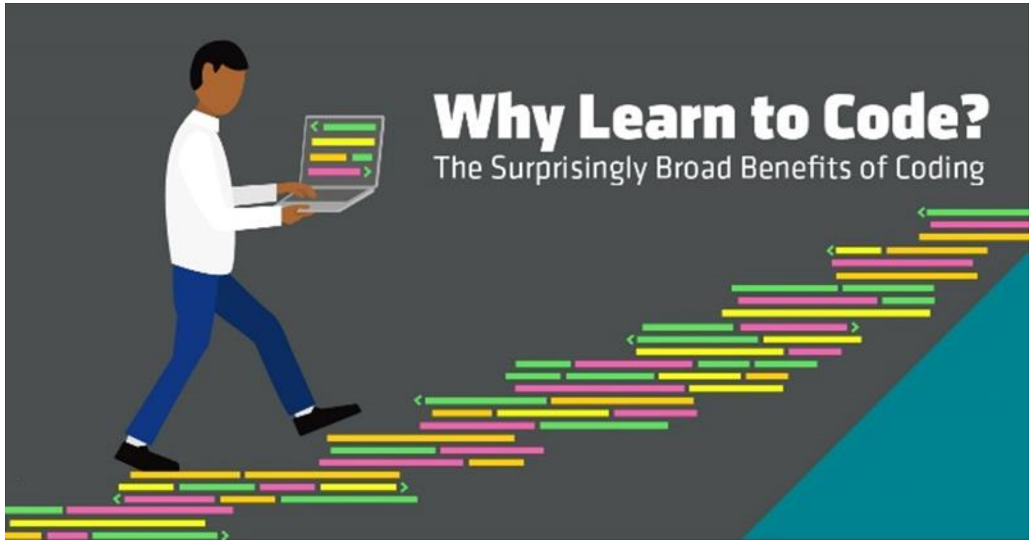
Introduction to Data Science for Geoscientists

GeoML 1 – Python Basics

Amr. Moslim



Today’s Agenda

- Download Python
- Working with Github
- Python Basics:
  - Variables
  - Data Structure
  - Operators
  - Control flow
  - Functions
  - Errors
- Python Working Libraries

Coding Workflow Basic Aspects

- **Assignment:**
  - Types of data structure (integer, float, String, Boolean)
- **Control flow:**
  - If statement
  - While loops
  - For loops
- **Mathematical Operators:**
  - (+, -, *, /)
  - (>, <, =, >=, <=, !=)
  - Logical operators:
    - (+=, -=, //, %, %%)
- **Functions:**

  A set of commands that works in sequence to perform a certain task that can include assignment, flow control tools and or mathematical expressions.
  - def: in Python
  - Function (x) in R
- **Error handling:**
  - Avoid having user errors
  - Handling errors
- **Reviewing:**
  - Debugging : to check that all the results as it should be even if you didn’t get any errors explicitly

Python Basics – Variables

Variables (identifiers):

can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore

⟶ Do Use normal letters (A-Z)

⟶ Don’t use special characters (@,!,#,..etc)

⟶ Don’t use keywords (for, import, ….)

⟶ Don’t start with numbers

⟶ It is better not to use (.) dot in variable name

Examples:

```
int_var = 10 # Integer variable
Float_var = 2.57 # Float Variable
Str_var = "Python Language" # String variable
Bool_var = TRUE
```

Python Basics – Variables

Keywords that can not be variable names

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif’, 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Python Basics – Operators

Operators:

Mathematical operators : (+ , - , * , /)

%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

Relational Operators : (==, < ,> ,<= ,=> , !=)

Logical Operators: (and, or, not)

- And : if both conditions are true
- Or : if any of them are true
- Not : If the boolean value is True it returns False and vice-versa.

Membership operators : (in, not in)

Python Basics – Data Structure

- **Dictionaries**: are used to store data values in key: value pairs. A dictionary is a collection which is unordered, changeable and does not allow duplicates. Dictionaries are written with curly brackets and have keys and values. Ex : **dict = {“name": “Amr", “Address": “Luxor", “DOB": 1964}**

- **Lists**: are used to store multiple items in a single variable. List are written in square brackets.

  Ex : **Lst = [“Amr”, “Mohamed”, “Ali”]**

Python Basics – Data Structure

- **Tuple** :is a collection which is ordered and unchangeable. Tuples are written with round brackets.

  Ex: **tpl = (“Amr”, “Mohamed”, “Ali”)**

- **Set** :is a collection which is both unordered and unindexed. Sets are written with curly brackets.

  Ex: **st = {“Amr”, “Mohamed”, “Ali”}**

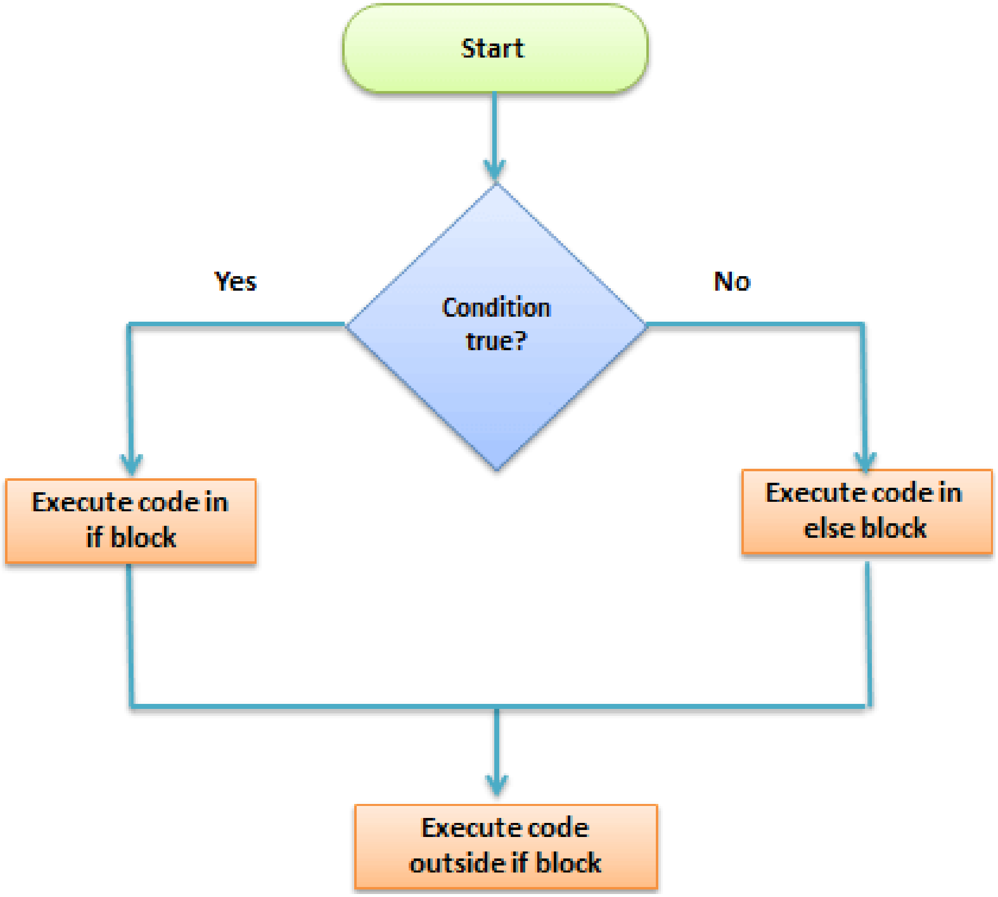
Python Basics – Data Structure

Data	Changeable	Ordered	Duplicates	Indexed	Iterations
Dictionary	Yes	No	No	By Key	Sometimes
List	Yes	Yes	Yes	Yes	All the time
Tuple	No	Yes	Yes	Yes	Not
Sets	No	No	No	No	Never

Amr.Moslim

Python Basics – Flow Control

- If Statement: a way of control the flow of the programming sequence to different directions based on the condition results.
- If statement can be used alone or with “else” or with “elif” as many times and it should be.



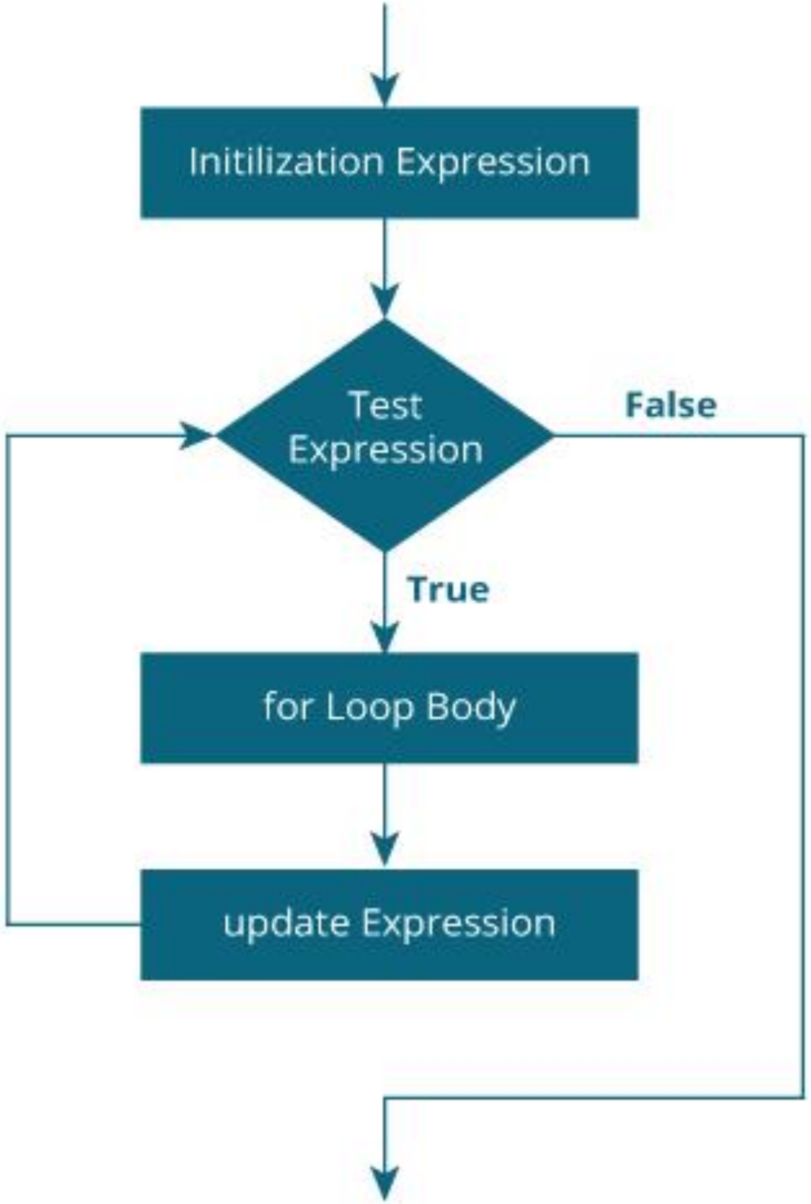
```
Example:
a = 33
b = 200
if b > a:
  print("b is greater than a")

>> b is greater than a
```

Python Basics – Flow Control

- For Loop: another way of controlling the flow of the programming sequence based on the desired output.
- For Loop needs only one condition to repeat the process.
- It uses mainly in the iteration process where we need to repeat the iteration process and update the output as many times as it should be to get accurate output.

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

```
>>"apple", "banana", "cherry"
```



Amr.Moslim

Python Basics – Functions

- A function is a block of sequential commands of code which only runs when it is called. You can pass data, known as parameters, into a function.
- A function can (or can not) return data as a result.

- Ex:

```
def velocity(distance , time):
  Velocity  = Distance / time
  return (Velocity)
```

- Velocity is name of the function, (distance and time) are the arguments of the functions , the value to be sent to the functions

Python most popular packages

- **Analysis packages**
  - Numpy : Numerical Manipulation and linear alegabra
  - Pandas : building & Manipulating DataFrames

- **Visualization packages**
  - Matplotlib : plots and contours
  - Seaborn : beautiful plots
  - Plotly : interactive plotting

- **Machine Learning packages**
  - Tensorflow : Neural NetWork and Deep learning
  - Keras: ML algorithms
  - Scikit Learn: ML algorithms and model evaluations

- **Scientific packages**
  - Scipy : scientific equations in python
  - Obspy : seismic manipulation and reading segy

- **Geoscience Package**
  - Welly : reading / write well logs las files
  - Lasio : reading / write well logs las files
  - Segyio : seismic Segy files reading / writing and manupliation.
  - Petopy : Petrophysical evaluation