Interactive Technologies Institute
LARSyS

Google Developer Groups
Madeira

UNIVERSIDADE da MADEIRA

MADEIRA INTERNATIONAL WORKSHOP IN MACHINE LEARNING

IKnowD

ORDEM DOS ENGENHEIROS TÉCNICOS
PORTUGAL

ORDEM DOS ENGENHEIROS TÉCNICOS

Platinum sponsor:

malogica solutions

Silver sponsor:

asseco
PST

Institutional support

arditi

startup madeira

Introduction to Python, pandas, NumPy and Matplotlib

Initial Quiz:

slido.com > #2433300 or scan the QR code:

What is Python?

- Python is a **general-purpose programming language**.
  - Python is not limited to a specific domain and can be used for a wide range of applications.
  - Commonly used for web development, data analysis, scientific computing, automation, and more.
- It was created by Guido van Rossum and first **released in 1991**.
- Named after the British comedy group Monty Python.

What is Python? (cont.)

- Python uses **dynamic typing**, allowing variables to change types during runtime.
- It includes various **data structures and abstractions**:
  - Lists;
  - Dictionaries;
  - Comprehensions.
- Python is **available on various platforms**, including Windows, macOS, and Linux.
- Code written in Python is **often portable across different operating systems**.

Why Python?

- Python is known for its **simplicity and efficiency**.
  - Prioritizes **readability** and uses a clean and straightforward syntax.
  - **Code is often described as being close to pseudocode**, making it easy for anyone to understand and develop.
- A wide variety of **open-source packages** and a **significant online community** are available.
- **Effectively and efficiently** analyze and visualize large amounts of data.
- Allows working with **multiprocessing** (multithreading and multiprocessing).

Basic Python Syntax and Structure

Algorithm

- **A set of operations and instructions** that, when applied to a problem, provide a solution to it.
- Examples of algorithms range from simple tasks like dividing two numbers to more complex ones like three-dimensional modeling.

Variable

- Variables are units used to **store data values**. Each variable is assigned a name.

Assignment Expression

- When we want to **assign a specific value** to a variable, that value is stored in the computer's memory. For example:

```
x = 3
is_equal = True
```

Basic Python Syntax and Structure (cont.)

Functions

- A function is a **container consisting of a set of operations and instructions**.
- Just like in mathematical functions, a function **can take a set of parameters**.
- This container **can be reused multiple times in our program** without the need to repeat the set of operations and instructions. For example:

```
def sum(x, y):
    return x + y
```

Composite Expressions

- These involve applying a set of operations to a set of operands. For example:

```
sum(multiply(x, y), 3)
```

Basic Python Syntax and Structure (cont.)

Mathematical Expressions

- Operations like addition (+), subtraction (-), multiplication (*), division (/), and others. For example:

```
x = 3
y = 2
result = x * y
```

Boolean Expressions

Logical operations such as logical AND (and), logical OR (or), and negation (not).

Basic Python Syntax and Structure (cont.)

Relational Expressions

- x == y to check if two objects are equal.
- x != y to check if two objects are not equal.
- x > y to check if x is greater than y.
- x < y to check if x is less than y.
- x >= y to check if x is greater than or equal to y.
- x <= y to check if x is less than or equal to y.

Comments:

- Comments are notes that we can leave to **help us** (and other developers) explain a complex part of an algorithm and/or maintain the algorithm. In Python, the use of "#" indicates a comment.

Reserved Keywords:

- **Words that cannot be used as variable names**. For example, we cannot name a variable "and."

Types of Variables

Type	Internal representation
Text	`str`
Numeric	`int, float, complex`
Sequence	`list, tuple, range`
Boolean	`bool`

Selection Structures

- Structures that allow **us to create a control flow in our program**. In other words, they enable us to execute one or more specific instructions if a condition is true. For example:

```
if condition (i.e., a boolean expression):
    # instructions
elif another_condition:
    # instructions
# ... (more elif conditions if needed)
else:
    # instructions
```

Loop Structures

- Structures that allow you to **repeatedly execute a set of instructions** as long as a condition is true. For example:

```
while condition:
    # instructions
```

And:

```
for i in range(5):
    # Code to be executed repeatedly
```

Library or Module

- A file that contains a **set of functions and tools that can be imported into your program without explicitly defining them**. These libraries/modules extend the capabilities of your code by providing pre-written functions and classes for specific tasks. For example:
  - **NumPy:** Allows you to work with multidimensional arrays and provides tools for numerical computing.
  - **pandas:** Enables the creation of DataFrames for data manipulation and analysis.
  - **Matplotlib:** Provides tools for creating graphical representations of data in Python.

Library or Module

- These libraries/modules **save you time and effort by providing ready-made solutions** for common programming tasks and complex computations.
- **Most of the libraries are interoperable!**
- **In order to install a library**, one can use the following command:

```
pip install <name of the library>
```