# Programming Mental Health in Python

## Course Content

- Python Basics
- Python Data Structures
- Looping and Function in Python
- Importing Datasets
- Data Wrangling
- Exploratory Data Analysis
- Data Visualization Using Python
- Univariate Analysis
- Model evaluation and Refinement

## 1.0 Python Basics

### i) What is Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility.

It is general-purpose programming language created by Guido van Rossum and first released in 1991.

Python is widely used in various industries and domains, including web development, data analysis, machine learning, artificial intelligence, scientific computing, and automation.

### Why Python

- Simplicity and readability
- Versatility
- Large and Active Community
- Cross-Platform Compatibility
- Open Source and Free

### ii) Installing Python

Link to Installation guide

### Installing python IDE

- **PyCharm :**Developed by JetBrains, PyCharm is one of the most feature-rich IDEs for Python development. It offers code analysis, debugging, and support for web development frameworks like Django and Flask.
  Link
- **IDLE** (Integrated Development and Learning Environment) comes bundled with the Python distribution
- **Visual Studio Code (VS Code)**: VS Code is a highly customizable and lightweight IDE developed by Microsoft. It has a large extension ecosystem, making it suitable for various programming languages, including Python
  Link
- **Jupyter Notebook / JupyterLab**: Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text
  Link

**iii) Variables, Types and assignments**

Variables are used to store data in memory.

In Python, variables are dynamically typed, meaning you don't have to explicitly declare the data type.

```
#Assigning values to variables
# variable = value

age = 25
price = 19.99
name = "Alice"
is_student = True

print(age)
a, b, c = 1, 2, 3 #multiple assignment

PI = 3.14159 #constants
print(PI)
x = 10
y = str(x)
```

**iv) Data types**

- Numeric types: int, float, complex
- Text type: str
- Boolean type: bool

- Sequence types: list, tuple, range
- Mapping type: dict
- Set types: set, frozenset

**v) Basic arithmetic operations**

Python supports basic arithmetic operations such as addition, subtraction, multiplication, division, exponentiation, and modulo.

These operations can be performed using operators like +, -, *, /, **, and %, respectively.

in the expression x + y, x and y are operands, and + is the operator.

```
# Adding variables

a = 10

b = 20

result = a + b

print("Result of addition:", result)  # Output: 30


#Subtracting variables
a = 20
b = 10
result = a - b
print("Result of subtraction:", result)  # Output: 10


# Multiplying variables
a = 10
b = 5
result = a * b
print("Result of multiplication:", result)  # Output: 50


# Division of integers

result = 10 / 3

print("Result of division:", result)  # Output: 3.3333333333333335

# Integer division

result = 10 // 3

print("Result of integer division:", result)  # Output: 3


# Division by zero
```

```
# Exponentiation
result = 2 ** 3
print("Result of exponentiation:", result)  # Output: 8
# Calculating powers
result = pow(2, 3)
print("Result of exponentiation using pow:", result)  # Output: 8
# Calculating roots
result = 9 ** 0.5
print("Square root of 9:", result)  # Output: 3.0
# Negative exponent
result = 2 ** -3
print("Result of negative exponentiation:", result)  # Output: 0.125


# Modulo operation
result = 10 % 3
print("Remainder of division:", result)  # Output: 1
```

**vi) Conditional statements** (if elif else)

Python conditional statements allow you to control the flow of your program based on conditions.

Using keywords like if, elif (short for "else if"), and else, you can execute different blocks of code depending on whether certain conditions are true or false

These conditional statements are essential for building decision-making logic into your Python programs

```
# if else to check whether a number is even/odd
number = 11
if number % 2 == 0:
    print("Even number")
else:
    print("Odd number")


# Using elif
z = 0
if z > 0:
    print("z is positive")
elif z == 0:
    print("z is zero")
else:
    print("z is negative")

# Ternary operator
x = 10
result = "x is greater than 5" if x > 5 else "x is not greater than 5"
print(result)
```