**AL-2001**

**Programming for AI**

**Lab # 1**

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* web development (server-side)
* software development
* mathematics
* System scripting

What can Python do?

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-oriented way or a functional way.

### Example:

print("Hello, World!")

**Output:**

Hello, World!

**Example:**

student\_name = "Ali"

roll\_number = "12345"

# Print student details

print("Student Name:", student\_name) **Output:** Student Name: Ali

print("Roll Number:", roll\_number) Roll Number: 12345

**Python Syntax (Basics)**

## Python Indentation:

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

**Example:**

if 5 > 2:  
  print("Five is greater than two!")

**Output:**

Five is greater than two!

**OR**

if 5 > 2:  
  print("Five is greater than two!")

**Output:**

**Cell In[7], line 2**

**print("Five is greater than two!")**

**^**

**IndentationError:** expected an indented block after 'if' statement on line 1

**Example:**

if 5 > 2:  
 print("Five is greater than two!")   
if 5 > 2:  
        print("Five is greater than two!")

**Output:**

Five is greater than two!

Five is greater than two!

**OR**

if 5 > 2:  
 print("Five is greater than two!")  
        print("Five is greater than two!")

**Output:**

**Cell In[10], line 3**

**print("Five is greater than two!")**

**^**

**IndentationError:** unexpected indent

* **Python Variables:**

In Python, variables are created when you assign a value to it:

**Example**

**Variables in Python:**

x = 5  
y = "Hello, World!"

print(x)

print(y)

**Output:**

5

Hello, World!

Python has no command for declaring a variable.

* **Comments:**

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

**Example:**

**Comments in Python:**

#This is a comment.  
print("Hello, World!")

**Example:**

#This is a comment  
#written in  
#more than just one line  
print("Hello, World!")

**Example:**

"""  
This is a comment  
written in  
more than just one line  
"""  
print("Hello, World!")

# **Python Variables**

## Variables

Variables are containers for storing data values.

In Python, variables are used to store data values. Here's a basic overview of how variables work in Python:

**1. Declaring Variables**

* In Python, you don't need to declare a variable explicitly with a type. Python is dynamically typed, meaning the variable type is inferred based on the value assigned to it.
* Example:

x = 5

y = "Hello, World!"

z = 3.14

**Creating Variables:**

Python has no command for declaring a variable.

A variable is created the moment you first assign a value to it.

**Example:**

x = 5  
y = "John"  
print(x)  
print(y)

**Example:**

x = 4       # x is of type int  
x = "Sally" # x is now of type str  
print(x)

**2. Variable Naming Rules**

* A variable name must start with a letter or the underscore character.
* A variable name cannot start with a number.
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_).
* Variable names are case-sensitive (e.g., myVar and myvar are two different variables).

**Example:**

my\_var = 10

\_var = "Underscore at the start"

VAR123 = 5.5

**3. Reassigning Variables**

* Variables can be reassigned to different types of values.
* Example:

a = 10 # a is an integer

a = "Hi!" # Now a is a string

**4. Multiple Assignments**

* Python allows multiple variables to be assigned in a single line.
* Example:

a, b, c = 1, 2, 3

**5. Type Casting**

* You can convert from one type to another using casting.
* Example:

x = int(3.5) # Converts float to int

y = str(100) # Converts int to string

z = float("4.2") # Converts string to float

* Example:

x = str(3)    # x will be '3'  
y = int(3)    # y will be 3  
z = float(3)  # z will be 3.0

**6. Checking Variable Type**

* You can check the type of a variable using the type() function.
* Example:

a = 10

print(type(a)) # prints <class 'int'>

**7. Get the Type**

You can get the data type of a variable with the type() function.

**Example:**

x = 5  
y = "John"  
print(type(x))  
print(type(y))

**Output:**

<class 'int'>

<class 'str'>

**Single or double quote:**

x = "John"

print(x)

#double quotes are the same as single quotes:

x = 'John'

print(x)

These are some basic concepts related to variables in Python. Variables are a fundamental part of Python programming, and understanding how to use them effectively is key to writing efficient code.