PYTHON-BASIC TO INTERMEDIATE

Learn with N@ima

Day 2

Variables and Datatypes

Python Variables

- **Variables** are containers for storing data values.
- Python has **no command** for declaring a variable.
- A variable is created the moment you first assign a value to it.
- Variables do not need to be declared with any particular **type**, and can even change type after they have been set.

Example

```
x = 5              # x is of type int
x = "John“         # x is now of type str

print(x)
```

Python Variables

Casting:

- If you want to specify the data type of a variable, this can be done with casting.

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

Get the Type:

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

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

Single or Double Quotes?

- String variables can be declared either by using single or double quotes

```
x = "John"
# is the same as
x = ’John’
```

Variable Names

A variable can have a short name (like x and y) or a more descriptive name.

Rules for Python variables:

- 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** (age, Age and AGE are three different variables)

Legal variable names:

```
myvar = "John"
my_var = "John"
_my_var = "John"
myVar = "John"
MYVAR = "John"
myvar2 = "John"
```

Multi Words Variable Names

- Variable names with more than one word can be difficult to read.
- There are several techniques you can use to make them more readable:

Camel Case:

Each word, except the first, starts with a capital letter:

❖ myVariableName = "John“

Pascal Case:

Each word starts with a capital letter:

❖ MyVariableName = "John“

Snake Case:

Each word is separated by an underscore character:

❖ my_variable_name = "John"

Python Variables - Assign Multiple Values

Many Values to Multiple Variables:

- Python allows you to assign values to multiple variables in one line:

```
x, y, z = "Orange", "Banana", "Cherry"
print(x)
print(y)
print(z)
```

Note: Make sure the number of variables matches the number of values, or else you will get an error.

Python Variables - Assign Multiple Values

One Values to Multiple Variables:

- You can assign the same value to multiple variables in one line:

```
x = y = z = "Orange"
print(x)
print(y)
print(z)
```

Unpack a Collection:

- If you have a collection of values in a list, tuple etc. Python allows you to extract the values into variables. This is called **unpacking**.

```
fruits = ["apple", "banana", "cherry"]

x, y ,z =  fruits

print(x)
print(y)
print(z)
```

Python – Output Variables

The Python print() function is often used to output variables.

In the print() function, you output multiple variables, separated by a comma.

You can also use the + operator to output multiple variables

For numbers, the + character works as a mathematical operator

In the print() function, when you try to combine a string and a number with the + operator, Python will give you an error

The best way to output multiple variables in the print() function is to separate them with commas, which even support different data types

Python - Global Variables

- Variables that are created **outside of a function** are known as global variables.
- Global variables can be used by everyone, both inside of functions and outside.
- If you create a variable with the same name inside a function, this variable will be local, and can only be used inside the function.
- The global variable with the same name will remain as it was, global and with the original value.
- Normally, when you create a variable inside a function, that variable is **local**, and can only be used inside that function.
- To create a global variable inside a function, you can use the ***global*** keyword.
- Also, use the ***global*** keyword if you want to **change a global variable** inside a function.

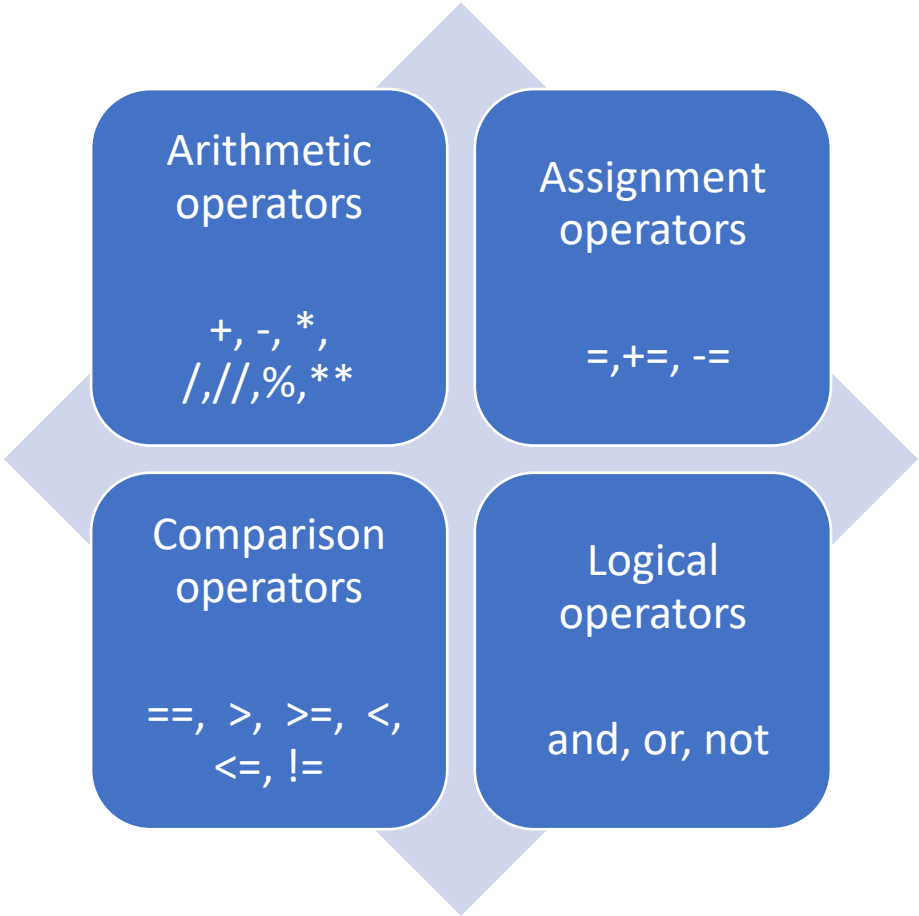
Python Data Types

Built-in Data Types

- In programming, data type is an important concept.
- Variables can store data of different types, and different types can do different things.
- You can get the data type of any object by using the type() function

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview
None Type:	NoneType

Operators in Python



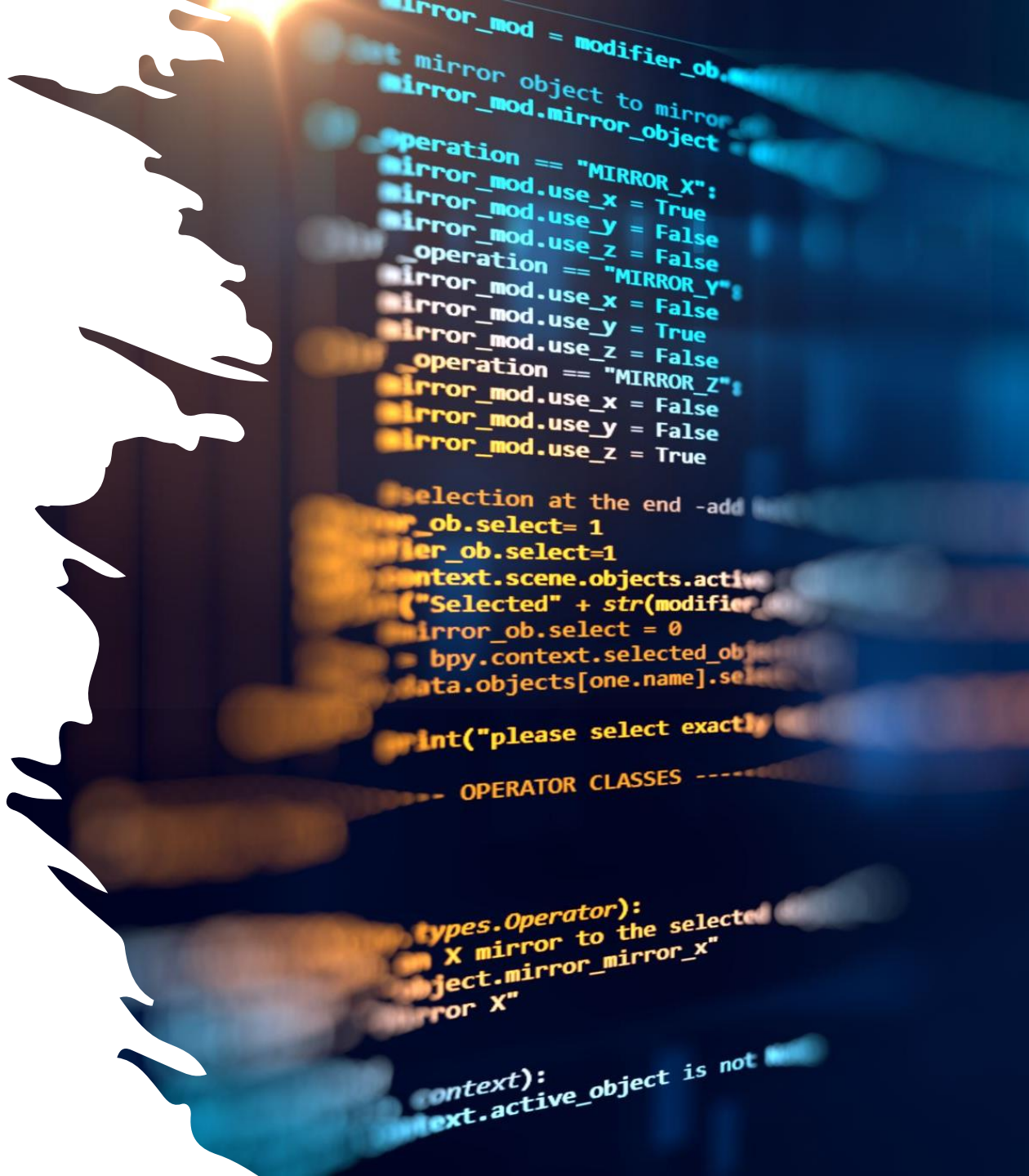
User Input

User Input:

- Python allows for user input.
- That means we are able to ask the user for input.
- Uses the **input()** method

Example: A = input(“Enter Your Name:”)

****Note:** the output of input is always a string (even if the number is entered)



Practice Work

1. Write a program to add three numbers
2. Write a Python program to find remainder when a number is divided by 5
3. Check the type of the variable assigned using input() function
4. Write a python program to find average of two numbers entered by the user.
5. Write a python program to calculate square of a number entered by the user.

THANK YOU