Newton School of Technology

4

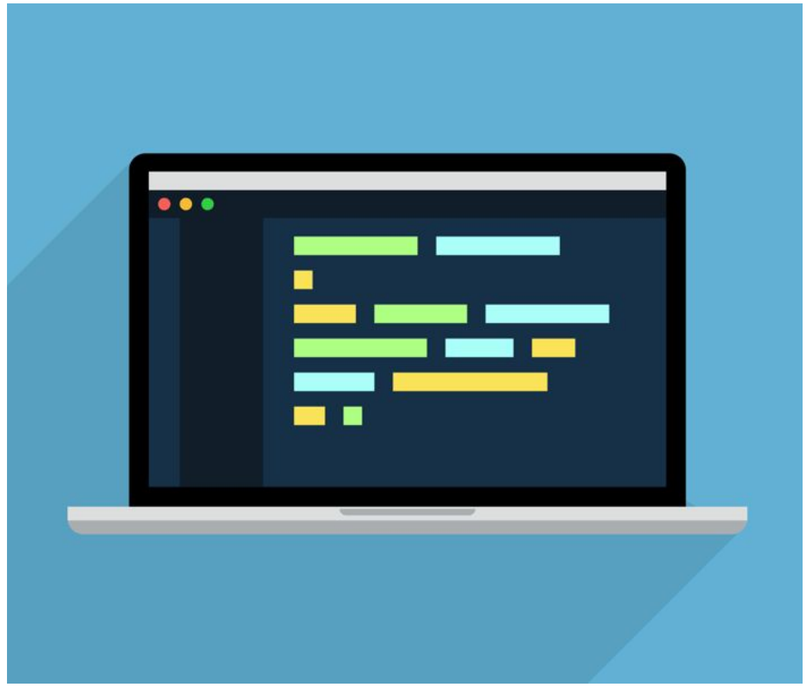
# Variables and Datatypes

by Gladden Rumao

CSA101 : Problem Solving with Programming

# Quick Recap :

- Introduction to python
- Print function
- newline \n character
- end and sep parameters
- Comments in Programming



# The Hiking Backpack :

Imagine you are going on a hike with a backpack. **You label each pocket with names like "snacks" , "water bottle" , "map" and "first aid kit".** When you need something, you check the label to find the item.

# The Hiking Backpack :

In programming, your **backpack is like the computer's memory**. Each **pocket is a variable with a name** that tells you what information it holds, and the items inside are like the values stored in the variables.

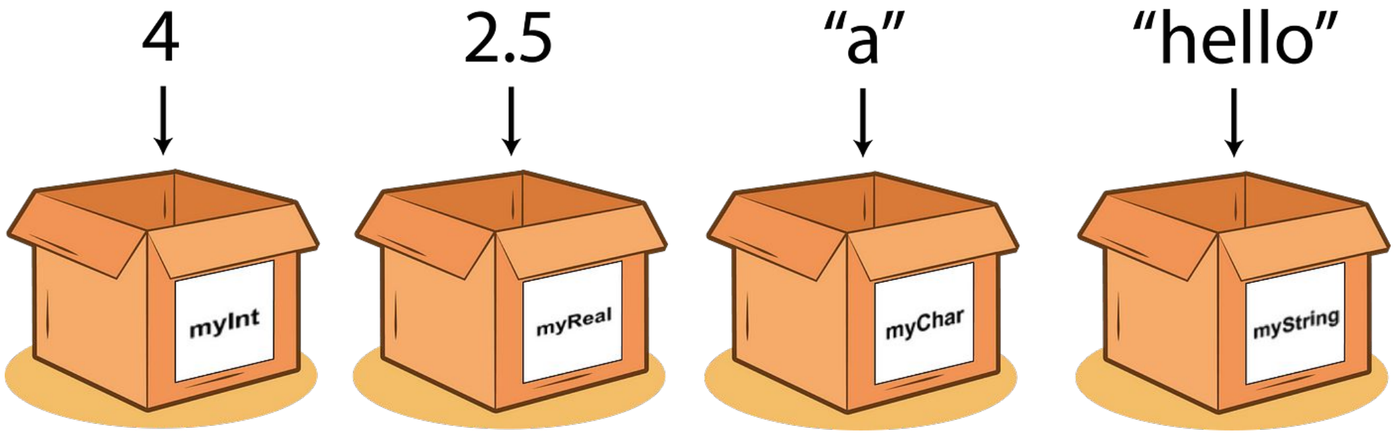
Newton School of Technology

# Variables in Programming

# Definition :

A variable in programming is like a labeled **container where you can store data**.

**Name associated with a memory location where data can be stored, modified, and retrieved** during the execution of a program.



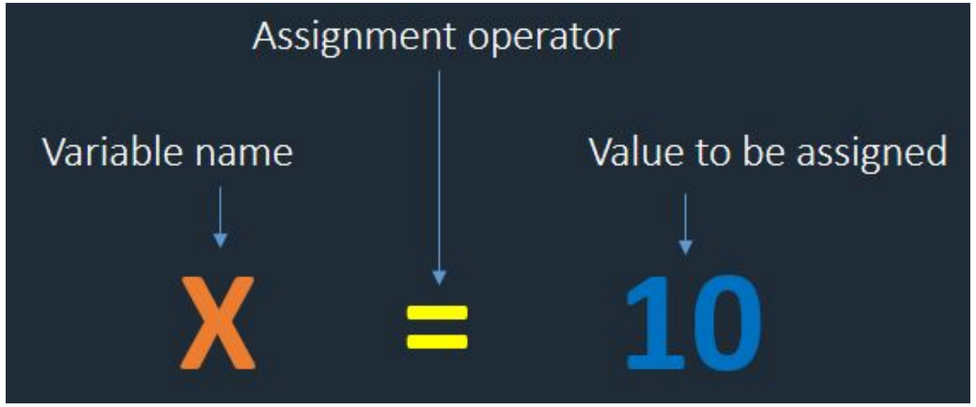
# Variables in Python :

In Python, a **variable is a reference to an object stored in memory**, created by assigning a value to an identifier, which then allows the program to interact with the object using the identifier.

```
x = 10
```

# Variables in Python :

- **Object Creation:** The value 10 is an object created in memory.
- **Variable Assignment:** The variable x is a reference to the object 10.
- **Interaction:** You can now use x to interact with the object 10.



# Variables in Python :

- **Step 1 : Choose a variable name**
- **Step 2 : Assign a value**

```
x = 5
```

# Assignment Operator (=)

Newton School of Technology

The assignment operator **‘=’** in programming is **used to assign a value to a variable**, storing the value in the variable for later use.

```
name = "Gladden"
city = "Mumbai"
temperature = 32
```

# Creating Variables

In Python, creating a variable is as simple as assigning it a value. The variable is created the moment you first assign a value to it.

```
x = 5
name = "John"
print(name)
```

**Output:**

```
John
```

Newton School of Technology

# Guess the Output :

```
best_opener = "David Warner"
best_opener = "Rohit Sharma"

print(best_opener)
```

# Re Assigning Variables

Variable can be **assigned a new value**.

```
x = 5
x = 7


print(x)
```

# Re Assigning Variables

Variable can be **assigned a new value**.

This can even change the variable's type, due to Python's dynamic typing.

```
x = 5
x = "Sara"
print(x)
```

**Output:**

```
Sara
```

# Re Assigning Variables

```python
gg = "Good Game"
gg = "Gladden Gang"
print(gg)
```

# Re Assigning Variables

python

```python
gg = "Good Game"
gg = "Gladden Gang"
print(gg)
```

Output:

Copy code

```
Gladden Gang
```

# Who’s the real GOAT ?

```
goat = "messi"
goat = "ronaldo"
print(goat)
```

Newton School of Technology

# Variable Naming Conventions :

# Start with a letter or underscore :

```
valid_name = 5          # Valid
_valid_name = 10        # Valid
name1 = "Alice"         # Valid
1name = "Bob"           # Invalid, starts with a number
```

# Snake Case for naming :

```
user_name = "Alex"
number_of_items = 10
```

# Use meaningful abbreviations:

```
max_height = 200
avg_score = 85
```

# Maintain Consistency :

```
# Consistent naming
item_price = 19.99
item_quantity = 5
total_amount = item_price * item_quantity

# Inconsistent naming
itemPrice = 19.99
Item_Quantity = 5
TOTALAMOUNT = itemPrice * Item_Quantity
```

# Assigning values to multiple variables

Python allows **assigning values to multiple variables in one line**, making your code concise and readable.

```
a = b = c = 5
print(a)
print(b)
print(c)
```

**Output:**

```
5
5
5
```

# Swapping Variables in Python :

# Incorrect Swapping :

```
a = 5
b = 10


a = b
b = a


print(a)
print(b)
```

# Guess the Output :

```
a = 5
b = 10


a = b
b = a


print(a)
print(b)
```

**a = 10**

**b = 10**

Newton School of Technology

# Swapping Variables in Python :

```
# Initial values
a = 5
b = 10

# Swapping values
a, b = b, a

print(a)  # Output: 10
print(b)  # Output: 5
```

Newton School of Technology

# Using f-string to embed variables :

```
a = 10
b = 20


result = f"The sum of {a} and {b} is {a + b}."
print(result)  # Output: The sum of 10 and 20 is 30.
```

# Change to f-string :

```
mess_food = "Poha"

print("Todays breakfast is Poha")
```

# Change to f-string :

```
mess_food = "Poha"

print(f"Todays breakfast is {mess_food}")
```

Newton School of Technology

# Using f-string to embed variables :

```
name = "Alice"
age = 30
message = f"Hello, {name}! You are {age} years old."
print(message)
```

Newton School of Technology

# Guess the Output :

```
name = "Gladden"
city = "Mumbai"
city = "Sonipat"
name = "Batman"
print(f"My name is {name}, I live in {city}")
```

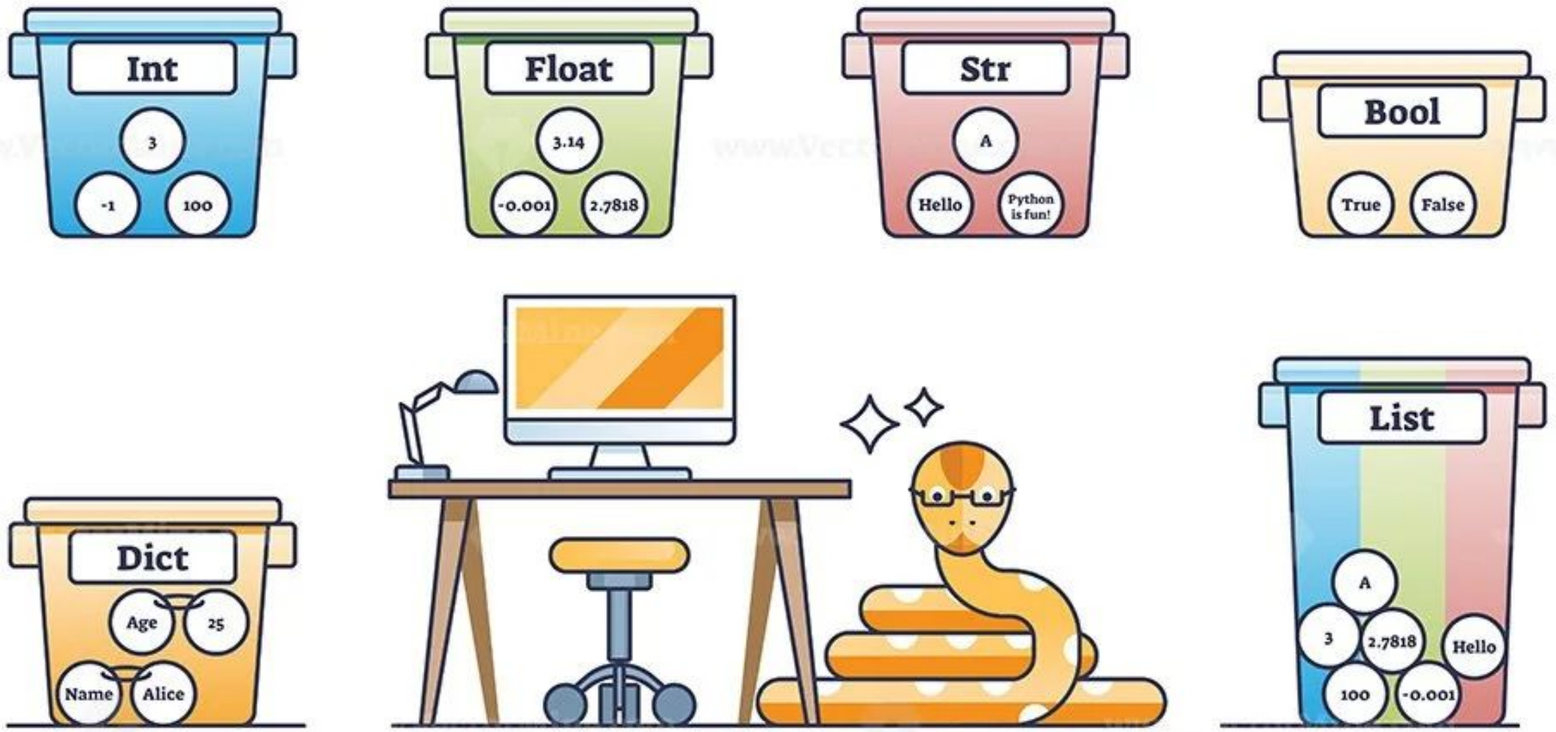
# Guess the Output :

```
event = "birthday party"
location = "park"
location = "beach"
event = "picnic"
print(f"The {event} will be held at the {location}.")
```

Newton School of Technology

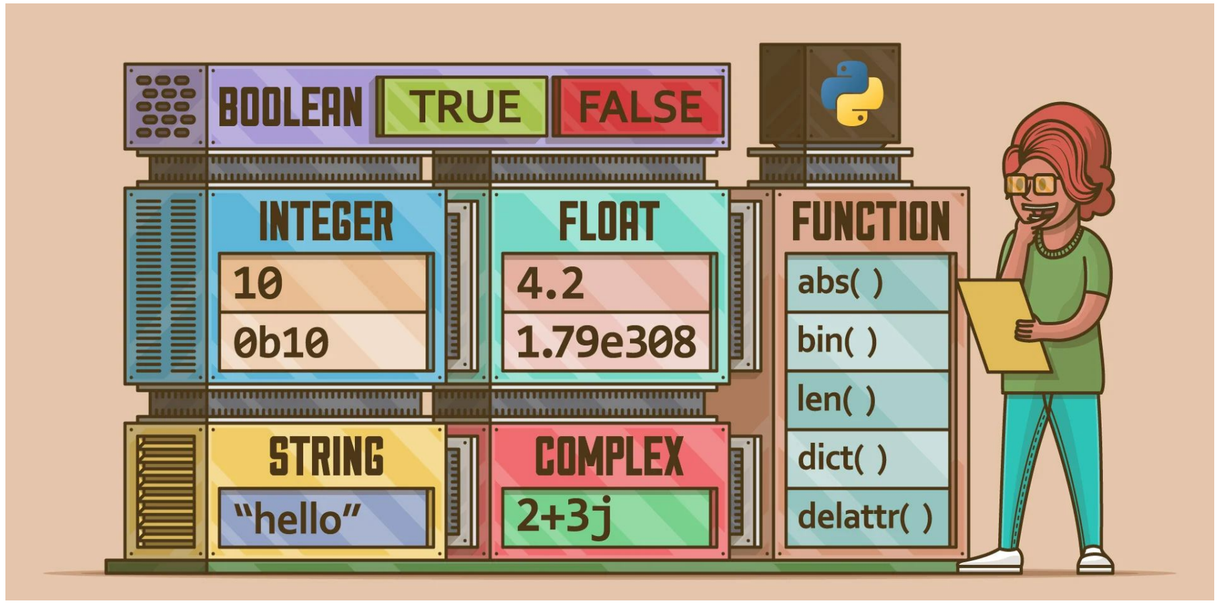
# Coding Assignment!

# Exploring Types of Data :



# Exploring Types of Data :

Similarly, in programming, data is categorized into different types to keep it organized and manageable. Data types (like **integers, strings, floats, and booleans**) help manage various kinds of data in your programs.

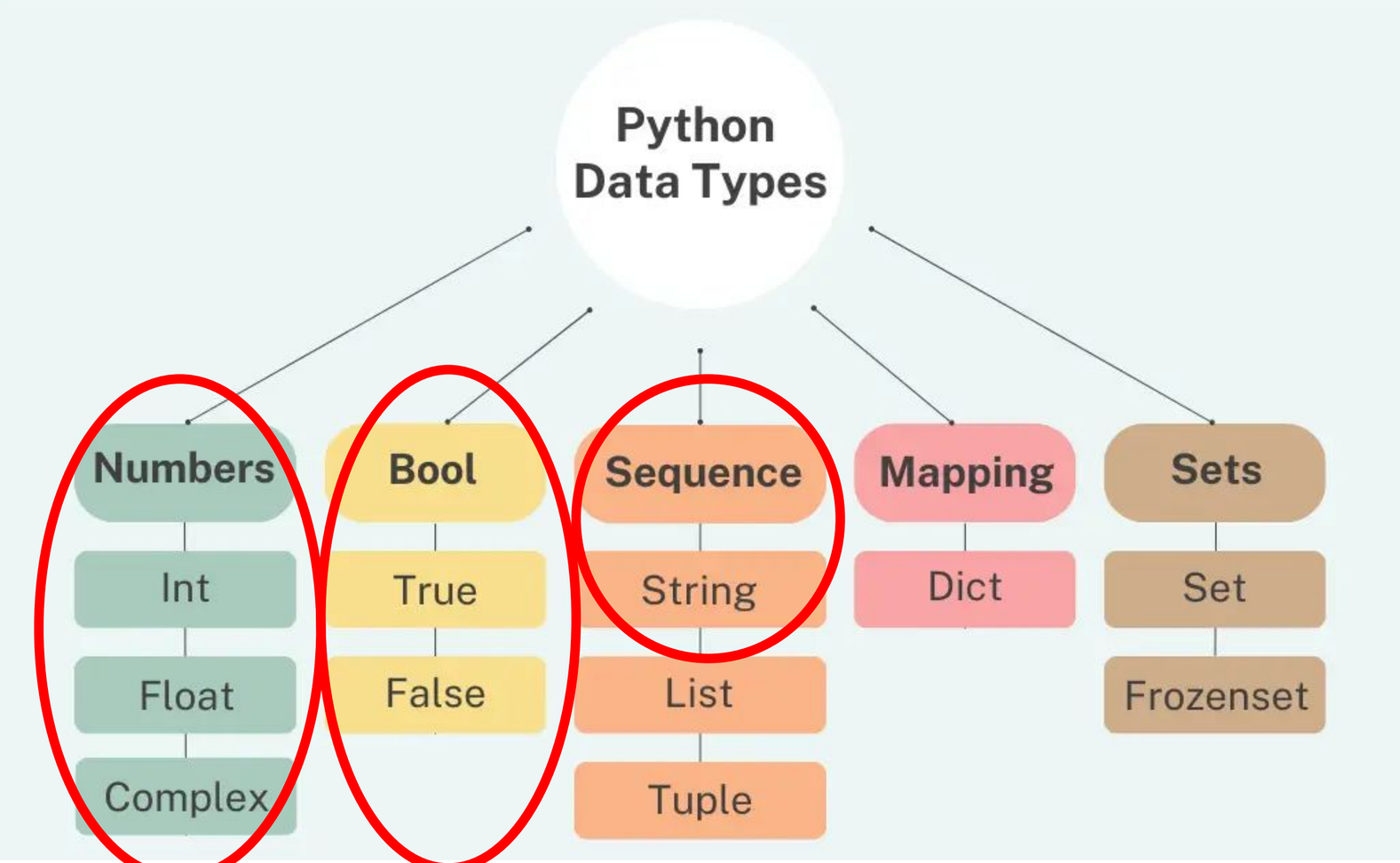


Newton School of Technology

# Data Types

# Data Types in Python

Newton School of Technology



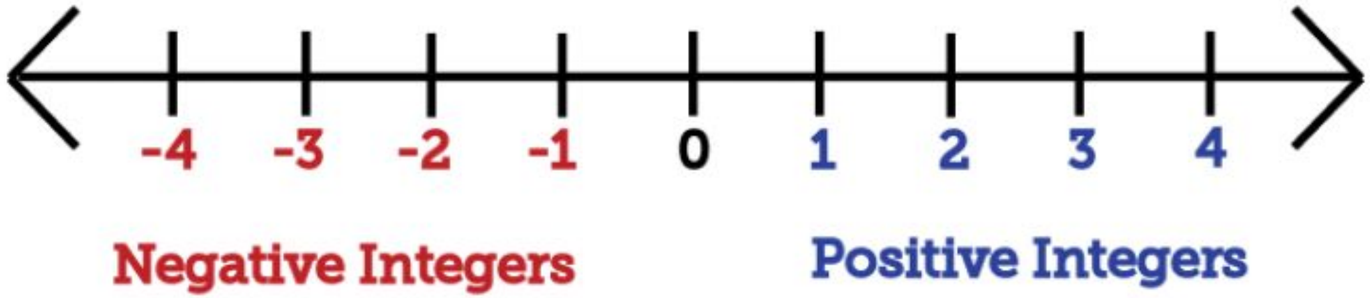
# Numeric Data types in Python :

1. Integers
2. Floating Point Numbers
3. Complex Numbers

# Integers in Python

Newton School of Technology

- **Integers are numbers without any fractional part.**
- **They can be 0, positive or negative.**
- **There is no limit to how long an integer can be in Python.**



# Integers in Python

Python is **dynamically typed**, so you don't need to declare the type. The variable is created the moment you first assign a value to it.

```
# Printing an integer in Python
num = 5
print("number =", num)
```

**Output:**

```
number = 5
```

# Floating Point Numbers

Newton School of Technology

- **Floating point numbers (float) are real numbers with floating point representation, they are specified by a decimal point.**
- **Floating point numbers are of float type.**
- **It is written as a number with a decimal point.**

# Floating Point Numbers

Newton School of Technology

```
# Printing a floating point number in Python
num = 5.55
print("number =", num)
```

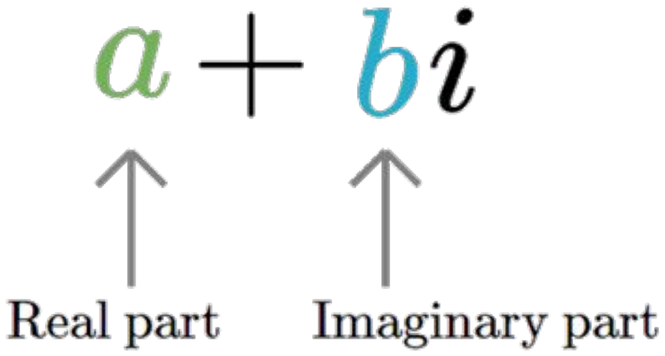
Output:

```
number = 5.55
```

# Complex Numbers

Newton School of Technology

- **Complex numbers in python are specified as (real part) + (imaginary part)j.**
- **They are written in the form a+bj in which a is the real value and b is the imaginary value, where j represents the imaginary part.**



# Complex Numbers

Newton School of Technology

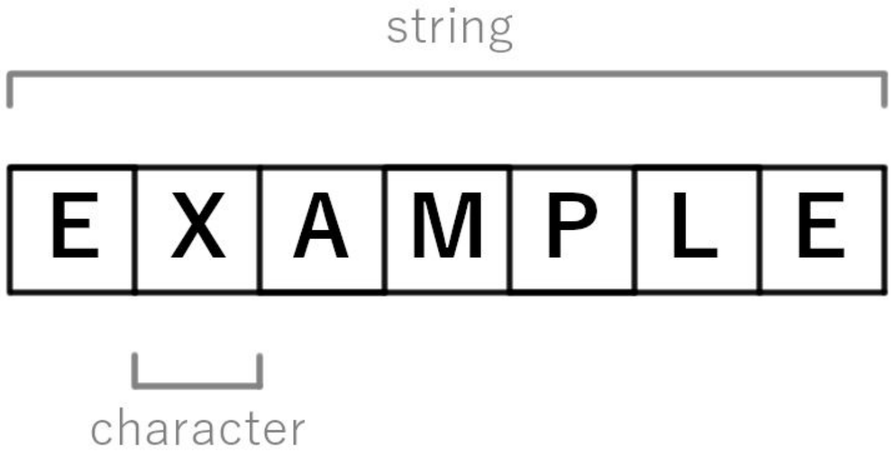
```
# Printing a complex number in Python
num = 5 + 5j
print("number =", num)
```

Output

```
number = (5+5j)
```

# Text Type in Python – String

- **A string can be defined as a sequence of characters.**
- **We use single and double quotation marks to represent strings**
- **Triple quotation marks are used for multi-line strings.**



# Text Type in Python – String

Newton School of Technology

python Copy code

```python
# Creating a string variable
my_string = "Hello, World!"

# Printing the string
print(my_string)
```

Output:

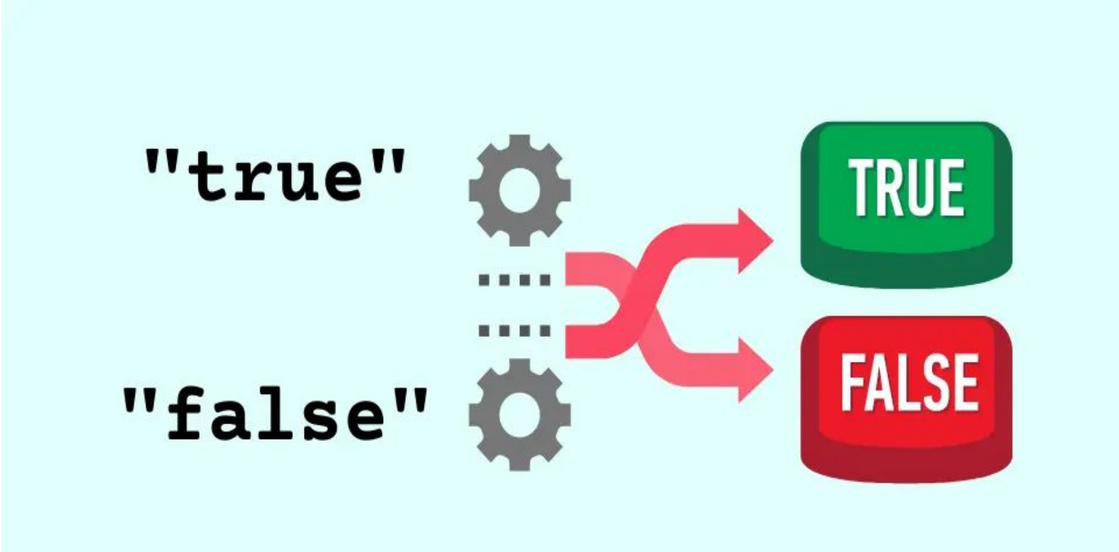
Copy code

```
Hello, World!
```

# Boolean Type

Newton School of Technology

- **Boolean is a data type that has one of two possible values, True or False.**
- **They are mostly used in creating the control flow of a program using conditional statements.**



# Boolean Type

Newton School of Technology

```
a = True
b = False

# Printing boolean values
print("a =", a)
print("b =", b)
```

**Output:**

```
a = True
b = False
```

# The type() function

Newton School of Technology

# The type() function

Newton School of Technology

- **The type() function is an inbuilt function in Python.**
- **It returns the class type of the argument passed to it.**

# The type() function

Newton School of Technology

```
# Program to demonstrate numerical values in Python.

x = 10
y = 10.0
z = 10 + 10j

print("Data type of", x, ":", type(x))
print("Data type of", y, ":", type(y))
print("Data type of", z, ":", type(z))
```

# The type() function

Newton School of Technology

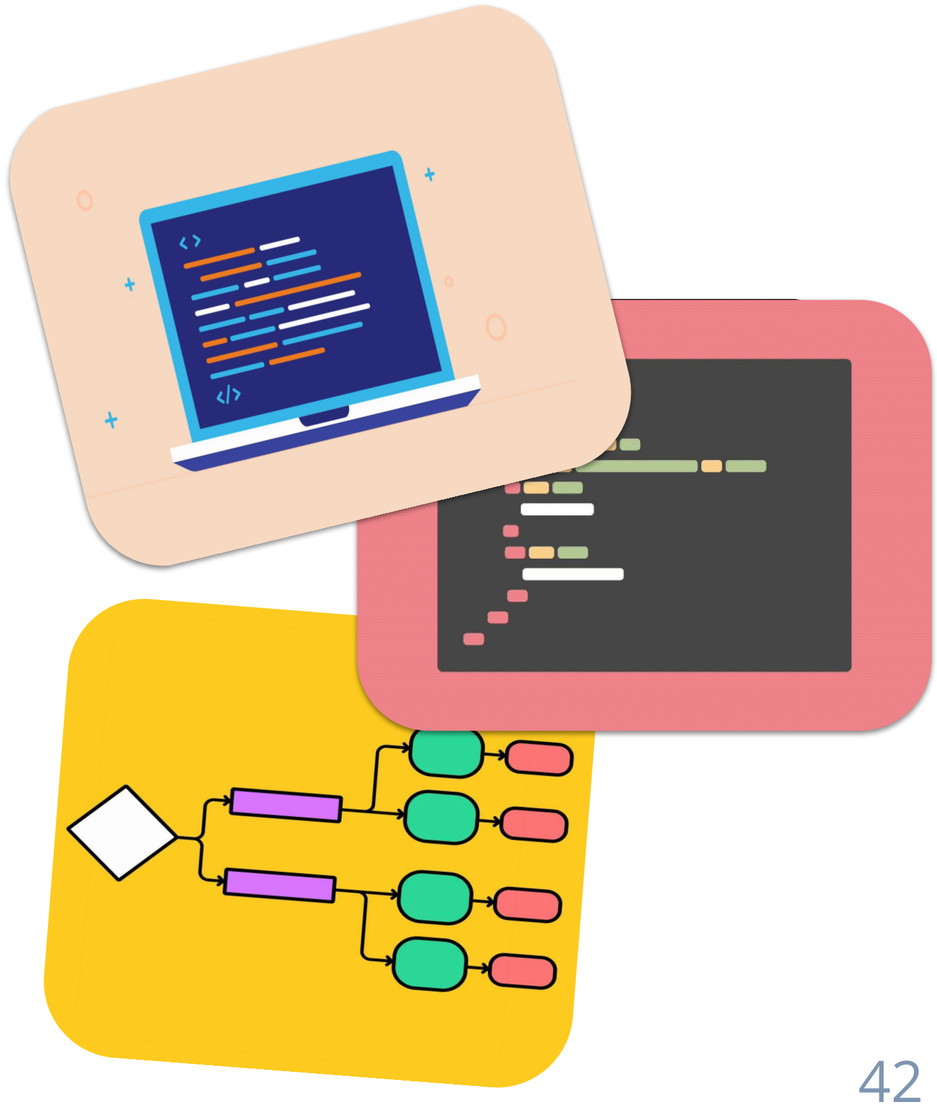
```
# Program to demonstrate numerical values in Python.

x = 10
y = 10.0
z = 10 + 10j

print("Data type of", x, ":", type(x))
print("Data type of", y, ":", type(y))
print("Data type of", z, ":", type(z))
```

**Output:**

```
Data type of 10 : <class 'int'>
Data type of 10.0 : <class 'float'>
Data type of (10+10j) : <class 'complex'>
```

# Summary

- **Variable** - In Python, a variable is a reference to an object stored in memory.
- **Datatype** -
  - Each variable has a data type in Python.
  - Numeric data types in python represent data with numeric values
  - Boolean has two types of values, True or False



# Thank You!