**DATATYPES**

**There are two types of datatypes. They are:-**

1.Primitive datatypes

2.Non-primitive datatypes

**1.Primitive datatypes:-** Primitive Data Types (or Basic Data Types) are the simplest data types that store single values.

In Python, the primitive data types are:

1. int → Integer numbers
2. float → Decimal numbers
3. complex → Complex numbers
4. bool → Boolean values (True / False)
5. str → Text (String)

**1). Integer (int):-**Stores whole numbers (positive, negative, or zero).

**Example:-**

x = 10

y = -5

print(x, type(x)) Output: 10 <class 'int'>

print(y, type(y)) Output: -5 <class 'int'>

**2). Float (float):-**Stores decimal numbers (floating-point numbers).

**Example:-**

pi = 3.14

temperature = -27.5

print(pi, type(pi)) Output: 3.14 <class 'float'>

print(temperature, type(temperature)) Output: -27.5 <class 'float'>

**3). Complex (complex):-**Stores numbers with real + imaginary parts.

**Example:-**

z = 2 + 3j

print(z, type(z)) Output: (2+3j) <class 'complex'>

print(z.real, z.imag) Output: 2.0 3.0

**4). Boolean (bool):-**Stores only True or False.

* Often the result of comparisons.

**Example:-**

is\_sunny = True

is\_raining = False

print(is\_sunny, type(is\_sunny)) Output: True <class 'bool'>

print(10 > 5, type(10 > 5)) Output: True <class 'bool'>

**5). String (str):-**Stores text data inside quotes (' ' or " ").

**Example:-**

name = "Python"

letter = 'A'

print(name, type(name)) Output: Python <class 'str'>

print(letter, type(letter)) Output: A <class 'str'>

**2.Non-Primitive datatypes:-** Non-primitive (or derived) data types are built using primitive types.

They can store multiple values and are often mutable (can be changed).

They are more complex than primitive types.

In Python, the common non-primitive data types are:

1. List
2. Tuple
3. Set and Frozen Set
4. Dictionary

**1. List**

* Ordered, mutable, allows duplicate elements.
* Written in square brackets [ ].

Example:-

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

print(fruits) Output: ['apple', 'banana', 'cherry']

print(fruits[0]) Output: apple

fruits.append("orange") # Adding element

print(fruits) Output: ['apple', 'banana', 'cherry', 'orange']

**2. Tuple**

* Ordered, immutable (cannot change), allows duplicate elements.
* Written in parentheses ( ).

Example:-

numbers = (1, 2, 3, 2)

print(numbers) # Output: (1, 2, 3, 2)

print(numbers[1]) # Output: 2

# numbers[0] = 10 Error (immutable)

**3. Set:-**

* Unordered, mutable, no duplicate elements.
* Written in curly braces { }.

Example:-

colors = {"red", "green", "blue", "red"}

print(colors) Output: {'red', 'green', 'blue'} (No duplicates)

colors.add("yellow")

print(colors) Output: {'red', 'green', 'blue', 'yellow'}

**4. Dictionary:-**

* Unordered, mutable, stores key-value pairs.
* Written in curly braces { key: value }.

Example:-

student = {"name": "Akshitha", "age": 20, "grade": "A"}

print(student["name"]) Output: Akshitha

student["age"] = 21 # Update value

print(student) Output: {'name': 'Akshitha', 'age': 21, 'grade': 'A'}