Exercise 44:

### ****Declaring Variables in Python:****

In Python, variables are declared by simply assigning a value to a variable name using the = operator. You don't need to specify the data type explicitly because Python is dynamically typed, meaning the data type is inferred based on the value assigned.

#### Examples:

# Declaring variables

x = 5 # x is an integer

y = 3.14 # y is a float

name = "Alice" # name is a string

is\_valid = True # is\_valid is a boolean

### ****Checking the Data Type of Variables:****

You can check the data type of a variable using the built-in type() function. This function returns the data type of the variable.

#### Example:

x = 5

y = 3.14

name = "Alice"

is\_valid = True

# Checking the type of variables

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

print(type(y)) # Output: <class 'float'>

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

print(type(is\_valid))# Output: <class 'bool'>

### Common Data Types in Python:

* **int**: Integer values, e.g., 5, 100, -10
* **float**: Floating-point (decimal) numbers, e.g., 3.14, -0.001
* **str**: String values (text), e.g., "Hello", 'Python'
* **bool**: Boolean values, True or False
* **list**: List, e.g., [1, 2, 3]
* **tuple**: Tuple, e.g., (1, 2, 3)
* **dict**: Dictionary, e.g., {"key": "value"}
* **set**: Set, e.g., {1, 2, 3}

### Example Checking Different Types:

my\_list = [1, 2, 3] # List

my\_tuple = (1, 2, 3) # Tuple

my\_dict = {"name": "Alice", "age": 25} # Dictionary

print(type(my\_list)) # Output: <class 'list'>

print(type(my\_tuple)) # Output: <class 'tuple'>

print(type(my\_dict)) # Output: <class 'dict'>