Great 👍 Let’s continue with your **roadmap → Module 3: Data Types & Type Conversion**.  
This is one of the most important basics in Python, because everything you work with is a **data type**.

**🔹 Module 3: Data Types and Type Conversion**

**1. Numbers**

Python supports three main types of numbers:

**a) Integers (int)**

* Whole numbers, positive or negative, without decimals.

x = 10

y = -50

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

**b) Floating point numbers (float)**

* Numbers with decimals.

pi = 3.14

g = -9.81

print(type(pi)) # <class 'float'>

**c) Complex numbers (complex)**

* Numbers with a real and imaginary part.

z = 2 + 3j

print(type(z)) # <class 'complex'>

print(z.real) # 2.0

print(z.imag) # 3.0

**2. Text**

**Strings (str)**

* Sequence of characters inside " " or ' '.

name = "Python"

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

* Strings can contain letters, numbers, and symbols.

**3. Boolean**

**Boolean (bool)**

* Only two values: True or False.

is\_active = True

is\_logged\_in = False

print(type(is\_active)) # <class 'bool'>

* Often used in conditions:

x = 5

print(x > 2) # True

print(x < 2) # False

**4. Type Conversion (Casting)**

Sometimes you need to convert data from one type to another.

**a) int() → Convert to integer**

num = int("10")

print(num) # 10

print(type(num)) # <class 'int'>

**b) float() → Convert to float**

num = float("3.14")

print(num) # 3.14

print(type(num)) # <class 'float'>

**c) str() → Convert to string**

age = 25

text = str(age)

print(text) # "25"

print(type(text)) # <class 'str'>

**d) bool() → Convert to boolean**

print(bool(0)) # False

print(bool(1)) # True

print(bool("")) # False (empty string is False)

print(bool("Hi")) # True

**🔑 Quick Notes**

1. Python is **dynamically typed** → you don’t declare type explicitly.
2. x = 10 # int
3. x = "Hello" # now string
4. Conversion is necessary when dealing with **input()**, because input always returns a **string**.
5. age = int(input("Enter your age: "))
6. print("Next year you will be", age + 1)

✅ **By the end of Module 3, you should be able to:**

* Identify number, string, and boolean types.
* Check type with type().
* Convert between types using int(), float(), str(), bool().

Do you want me to continue with **Module 4: Operators** next, or should I give you **practice questions for Module 3** so you can strengthen this topic before moving forward?

Perfect 👍 Let’s go deep into **Data Types in Python** (this is Module 3 from your roadmap, but in full detail).

**🔹 Python Data Types (Detailed Explanation)**

In Python, **everything is an object**.  
Data type defines the kind of value a variable can hold and what operations can be performed on it.

**1. Numeric Types**

Python has three main numeric data types:

**a) Integer (int)**

* Whole numbers (positive, negative, or zero).
* No decimal point.
* Can be arbitrarily large in Python (no fixed size like C/Java).

x = 10

y = -200

z = 0

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

**b) Float (float)**

* Numbers with decimals.
* Stored as double-precision floating-point internally.
* Can represent scientific notation (e.g., 1.2e3 = 1200).

pi = 3.14159

g = -9.81

sci = 1.2e3 # 1200.0

print(type(pi)) # <class 'float'>

**c) Complex (complex)**

* Numbers with **real** and **imaginary** parts.
* Written as a + bj (where j = √-1).

z = 2 + 3j

print(type(z)) # <class 'complex'>

print(z.real) # 2.0

print(z.imag) # 3.0

**2. Text Type**

**String (str)**

* A sequence of Unicode characters inside ' ' or " ".
* Python also allows multi-line strings using ''' ''' or """ """.

name = "Python"

message = 'Hello World'

multi = """This is

a multi-line

string."""

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

👉 Strings are **immutable** (once created, cannot be changed directly).

**3. Boolean Type**

**Boolean (bool)**

* Represents **True** or **False**.
* Often used in conditions and logical operations.

is\_active = True

is\_logged\_in = False

print(type(is\_active)) # <class 'bool'>

print(10 > 5) # True

print(5 == 7) # False

👉 Behind the scenes: True = 1, False = 0.

**4. Sequence Types**

**a) List (list)**

* Ordered, mutable (can be changed).
* Can hold different data types.

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

nums = [1, 2, 3, 4, 5]

mixed = [10, "hello", 3.14, True]

print(type(fruits)) # <class 'list'>

**b) Tuple (tuple)**

* Ordered, immutable (cannot be changed).
* Useful for fixed collections.

colors = ("red", "green", "blue")

print(type(colors)) # <class 'tuple'>

**c) Range (range)**

* Represents a sequence of numbers.
* Mostly used in loops.

r = range(5) # 0,1,2,3,4

for i in r:

print(i)

**5. Mapping Type**

**Dictionary (dict)**

* Stores key–value pairs.
* Keys must be unique & immutable (strings, numbers, tuples).
* Values can be anything.

student = {

"name": "Alice",

"age": 21,

"is\_active": True

}

print(type(student)) # <class 'dict'>

print(student["name"]) # Alice

**6. Set Types**

**a) Set (set)**

* Unordered collection of unique elements.
* No duplicates allowed.

nums = {1, 2, 3, 3, 4}

print(nums) # {1, 2, 3, 4}

print(type(nums)) # <class 'set'>

**b) Frozen Set (frozenset)**

* Like a set, but **immutable**.

fs = frozenset([1, 2, 3])

print(type(fs)) # <class 'frozenset'>

**7. Binary Types**

Used for storing binary data.

* **bytes** – immutable sequence of bytes.
* **bytearray** – mutable sequence of bytes.
* **memoryview** – view object of another binary object.

b = bytes([65, 66, 67]) # ASCII for A, B, C

print(b) # b'ABC'

ba = bytearray([65, 66, 67])

ba[0] = 90

print(ba) # bytearray(b'ZBC')

**8. None Type**

**None**

* Represents the absence of a value.
* Equivalent to null in other languages.

x = None

print(type(x)) # <class 'NoneType'>

**9. Type Conversion (Casting)**

Python lets you convert between types using built-in functions:

x = "100"

print(int(x)) # 100 (string → int)

print(float(x)) # 100.0 (string → float)

num = 25

print(str(num)) # "25" (int → string)

print(bool(0)) # False

print(bool(1)) # True

**✅ Quick Summary**

* **Numbers** → int, float, complex
* **Text** → str
* **Boolean** → bool
* **Sequences** → list, tuple, range
* **Mapping** → dict
* **Set types** → set, frozenset
* **Binary** → bytes, bytearray, memoryview
* **Special** → None

👉 Now you have a **detailed picture of all Python data types**.  
Would you like me to create a **comparison table with examples for each data type** (easy reference), or should I move ahead to **Operators (Module 4)** in your roadmap?