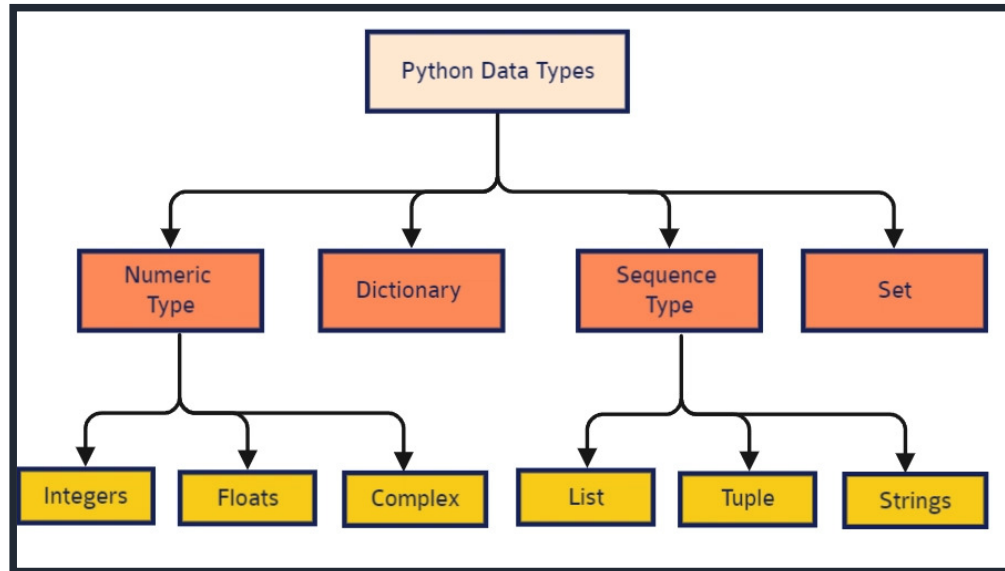


**Course
&
Test Series**

Data Types in Python

Data Types in Python

A data type in programming refers to the kind or category of data a variable can hold. It defines what kind of value a variable can store and what operations can be performed on that value.



For example:

```

x = 10      # Integer
y = "Hello" # String
z = 10.5    # Float
    
```

Here:

- x → numeric (integer)
- y → text (string)
- z → decimal number (float)

Each type is a class in Python.

type() in Python

What it does: The `type()` function tells you the **data type** of a value, variable, or object in Python.

Syntax: `type(object)`

- `object` → the variable or value whose type you want to check.

- CBSE
- ICSE
- NTSE
- Banking & Insurance
- Central Govt. Service
- State Govt. Services
- LAW Entrance
- MBA Entrance
- Railways & Metro Services
- ...many more

abhyasonline.in



**Course
&
Test Series**

Data Types in Python

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking &
Insurance**

 **Central Govt.
Service**

 **State Govt.
Services**

 **LAW
Entrance**

 **MBA
Entrance**

 **Railways & Metro
Services**

...many more
abhyasonline.in

Built-in Data Types in Python

Python has 14 standard data types, categorized into 5 major groups:

Category	Data Types
1. Basic / Standard	int, float, complex, bool, str
2. Sequence	list, tuple, range
3. Set	set, frozenset
4. Mapping	dict
5. Binary	bytes, bytearray, memoryview
6. Special	NoneType

Numeric Data Types

a) Integer (int)

- Whole numbers (positive, negative, or zero).

```
x = 10
y = -5
print(type(x)) # <class 'int'>
```

b) Float (floating point numbers)

- Numbers with decimals.

```
pi = 3.14
temp = -7.5
print(type(pi)) # <class 'float'>
```

Notes:

- Has decimal point.
- Used for scientific or precise calculations.
- Supports arithmetic and rounding operations.

c) Complex (complex numbers)

- Numbers with real and imaginary parts. Used to represent complex numbers of form $a + bj$.

```
z1 = 2 + 3j
z2 = complex(1, 4)

print(z1)
print(type(z1))
print(z1.real) # real part
print(z1.imag) # imaginary part
```

Course
&
Test Series

Data Types in Python

 CBSE

Output:
(2+3j)
<class 'complex'>
2.0
3.0

Note:
Used in advanced math and engineering

 ICSE

String Data Type (str)

Represents sequence of characters, enclosed in quotes.
name = "Python"
greet = 'Hello'
sentence = """This is a multi-line string."""

 NTSE

Example:
print(type(name))
print(name[0]) # indexing
print(name[-1]) # negative index
print(name[0:3]) # slicing
print(name.upper()) # string method

 Banking & Insurance

Output:
<class 'str'>
p
n
Pyt
PYTHON

 Central Govt. Service

Notes:

- Strings are immutable (cannot be changed after creation).
- You can concatenate with +, and repeat with *.

 State Govt. Services

Sequence Data Types

 LAW Entrance

a) List (list)

- Ordered, changeable collection, written in square brackets [].

 MBA Entrance

Example:
fruits = ["apple", "banana", "mango"]
print(fruits)
print(fruits[1])

 Railways & Metro Services

...many more

abhyasonline.in

Course
&
Test Series

Data Types in Python

```
fruits[1] = "grape"  
print(fruits)
```

Output:

```
['apple', 'banana', 'mango']  
banana  
['apple', 'grape', 'mango']
```

Notes:

- Mutable → you can modify elements.
- Can contain **different types** of data.

b) Tuple (tuple)

- Ordered, unchangeable collection, written in round brackets ().

Example:

```
colors = ("red", "green", "blue")  
print(colors)  
print(colors[0])  
# colors[0] = "yellow" # ❌ Error: tuples are immutable
```

Output:

```
('red', 'green', 'blue')  
red
```

Notes:

- Tuples are faster and memory-efficient.
- Used for fixed data (e.g. coordinates).

c) range

Represents a **sequence of numbers**.

```
r = range(1, 6)  
print(list(r))
```

Output:

```
[1, 2, 3, 4, 5]
```

Notes:

- Commonly used in loops like for i in range(5):



CBSE



ICSE



NTSE



Banking &
Insurance



Central Govt.
Service



State Govt.
Services



LAW
Entrance



MBA
Entrance



Railways & Metro
Services

...many more

abhyasonline.in



Course
&
Test Series

Data Types in Python

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more
abhyasonline.in

Mapping Data Type

Dictionary (dict)

- Stores data in key-value pairs { }.

Example:

```
student = {  
    "name": "Alice",  
    "age": 20,  
    "marks": 85  
}
```

```
print(student)  
print(student["name"])  
student["age"] = 21 # modify value  
print(student)
```

Output:

```
{'name': 'Alice', 'age': 20, 'marks': 85}  
Alice  
{'name': 'Alice', 'age': 21, 'marks': 85}
```

Notes:

- Keys are unique and immutable.
- Values can be any data type.
- Used widely for structured data.

Set Types

a) Set (set)

- Unordered collection of unique items { }.

Example:

```
nums = {1, 2, 3, 2, 1}  
print(nums) # duplicates removed  
nums.add(4)  
print(nums)
```

Output:

```
{1, 2, 3}  
{1, 2, 3, 4}
```

Notes:

Course
&
Test Series

Data Types in Python

- No duplicate values.
- Unordered (no indexing).
- Useful for mathematical operations like union, intersection.

 **CBSE**

b) Frozen Set (frozenset)

- Like a set, but immutable (cannot be changed).

Example:

```
fs = frozenset([1, 2, 3])
print(fs)
# fs.add(4) # ❌ Error: cannot modify Boolean Data Type
```

Output:

```
frozenset({1, 2, 3})
```

 **ICSE**

 **NTSE**

Bool (bool)

- Represents True or False.

Example:

```
x = True
y = False
print(type(x))
print(10 > 5)
print(10 < 5)
```

Output:

```
<class 'bool'>
True
False
```

 **Banking & Insurance**

 **Central Govt. Service**

 **State Govt. Services**

None Data Type

None

- Represents "nothing" or "no value".
- ```
data = None
print(type(data)) # <class 'NoneType'>
```

 **LAW Entrance**

 **MBA Entrance**

**Summary:**

- Numeric → int, float, complex
- Sequence → str, list, tuple
- Mapping → dict
- Set Types → set, frozenset
- Boolean → bool
- None Type → None

 **Railways & Metro Services**

...many more

abhyasonline.in

Course  
&  
Test Series

Data Types in Python

**Example Code (Mix of Types):**

```
a = 10 # int
b = 3.5 # float
c = "Hello" # str
d = [1, 2, 3] # list
e = (4, 5, 6) # tuple
f = {"x": 1, "y": 2} # dict
g = {1, 2, 3} # set
h = True # bool
i = None # NoneType
```

**Solved Example 1:** Write a Python program to create variables of different data types and print their types.

**Solution:**

```
a = 100 # Integer
b = 25.5 # Float
c = "Hello, Python!" # String
d = True # Boolean
e = [1, 2, 3, 4] # List
f = (10, 20, 30) # Tuple
g = {"name": "Aarav", "age": 20} # Dictionary
h = {1, 2, 3, 3, 2} # Set
```

```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
print(type(g))
print(type(h))
```

**Output:**

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'list'>
<class 'tuple'>
<class 'dict'>
<class 'set'>
```

- CBSE
- ICSE
- NTSE
- Banking & Insurance
- Central Govt. Service
- State Govt. Services
- LAW Entrance
- MBA Entrance
- Railways & Metro Services
- ...many more

abhyasonline.in



Course  
&  
Test Series

Data Types in Python

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Explanation:

- type() tells the data type of a variable.
- Each variable stores a different type of data.
- The set automatically removes duplicates ({1, 2, 3} only).

Solved Example 2: Write a Python program to convert one data type into another and print the results.

Solution:

```
Given variables
x = "50" # String type
y = 10 # Integer type

Convert string to integer and perform addition
result = int(x) + y
```

```
Convert result to float and string
result_float = float(result)
result_str = str(result)
```

```
print("Integer result:", result)
print("Float result:", result_float)
print("String result:", result_str)
```

Output:

```
Integer result: 60
Float result: 60.0
String result: 60
```

Explanation:

- "50" (string) is converted to integer using int(x).
- The addition  $50 + 10 \rightarrow 60$ .
- Then  $\text{float}(60) \rightarrow 60.0$ , and  $\text{str}(60) \rightarrow "60"$ .
- Type casting lets you change data types easily.

Assignment

Ques 1: Write a Python program that takes:

- a number as a **string** (for example "25")
- converts it to an **integer**, adds 10 to it,
- then converts the result to a **float** and prints all three values with their data types.

Hint: Use int(), float(), and type() functions.

Course  
&  
Test Series

Data Types in Python

**Ques 2:** Create:

- A list of 5 favorite fruits.
- Convert that list into a tuple.

Then,

- Print both the list and the tuple.
- Try to change one element in both and observe what happens.

**Hint:** Use tuple() function and note the difference between **mutable** and **immutable** types.

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

