**Python Data Types - Beginner Questions**

1. **What is a data type in Python?**

Answer:

In Python, a **data type** refers to the kind of value a variable holds. It determines what operations can be performed on that data.

1. **List all the data types that we have in Python?**

**Numeric Types**

* **int** – Whole numbers like 5 or -3.
* **float** – Decimal numbers like 3.14.
* **complex** – Numbers with a real and imaginary part, like 2 + 3j.

**Text Type**

* **str** – Text (strings), like "Hello".

**Sequence Types**

* **list** – Ordered, changeable collection (e.g., [1, 2, 3]).
* **tuple** – Like a list, but fixed (e.g., (1, 2, 3)).
* **range** – A sequence of numbers, often used in loops (range(5)).

**Mapping Type**

* **dict** – Collection of key → value pairs, like {"name": "Alice", "age": 25}.

**Set Types**

* **set** – Unordered collection of unique items (e.g., {1, 2, 3}).
* **frozenset** – Same as set but cannot be changed.

**Boolean Type**

* **bool** – Truth values: True or False.

1. **What is the difference between mutable and immutable data types?**

Mutable data types are like clay—you can alter them after creating them. Immutable types are like stone—you can’t change them once made.

1. **What is the difference between int, float, and complex?**
   1. **int (Integer):** Represents whole numbers without any decimal part, like 5, 0, or -42.
   2. **float (Floating‑point number):** Represents decimal numbers, like 3.14 or -0.001.
   3. **3. complex (Complex number**): Represents numbers with a real part and an imaginary part, written like 2 + 3j.
2. **Which data type is used to represent text in Python?**

In Python, the data type used to represent text is **str**, which stands for **string**.

str – Textual Data (Strings)

 It holds **sequences of characters**, such as letters, numbers, punctuation, and emoji.

 You can create one by wrapping text in **single quotes**, **double quotes**, or even **triple quotes** for multi-line strings.

1. **What is the output of type(521) and type("521")?**

OUTPUT OF type(521)

521 is a whole number --🡪 It’s type is int.

OUTPUT OF type(521)

"521" is **text** (because of the quotes) → its type is **str**.

1. **What is the difference between list, tuple, and set?**

| **Feature** | **List** | **Tuple** | **Set** |
| --- | --- | --- | --- |
| Ordered? | ✅ Yes | ✅ Yes | ❌ No |
| Mutable? | ✅ Yes | ❌ No | ✅ Yes |
| Allows duplicates? | ✅ Yes | ✅ Yes | ❌ No |
| Supports indexing? | ✅ Yes | ✅ Yes | ❌ No |

1. **How is a dictionary different from a list?**

 **List**: like a numbered list you go through by position.

 **Dictionary**: like a label on a box—open it by name to find what’s inside.

1. **What is the default data type of a number with a decimal point?**

In Python, when you write a number with a decimal point—like 3.14 or 2.0—it’s automatically treated as a **float**, which is Python’s default decimal/real number type.

10.**Declare variables of type int, float, string, and complex.**

1. Integer (whole numbers):

my\_int = 42

print(type(my\_int))

2. my\_float = 3.14

print(type(my\_float))

3. 3. String (text)

my\_str = "Hello, world!"

print(type(my\_str))

4.Complex (numbers with real + imaginary parts)

my\_complex = 2 + 3j

print(type(my\_complex))

 **my\_int = 42** → a whole number (int)

 **my\_float = 3.14** → a decimal number (float)

 **my\_str = "Hello..."** → text (string)

 **my\_complex = 2+3j** → a complex number with real and imaginary parts

**11.Take any 3 datatypes examples and check type of each variable using the type() function.**

num = 100

print(type(num)) # → <class 'int'>

# Example 2: Float

pi = 3.14159

print(type(pi)) # → <class 'float'>

# Example 3: String

greeting = "Hello!"

print(type(greeting)) # → <class 'str'>

**12.What happens if you try to add a string and an integer?**

If you try to add (concatenate) a **string** and an **integer** in Python using the + operator—like this:

"Age: " + 25

—it will raise a TypeError:

TypeError: can only concatenate str (not "int") to str

Why? Because Python is strongly typed and won't automatically convert the integer to a string

**13.What is the output of:**

* 1. x = [1, 2, 3]
  2. y = (1, 2, 3)
  3. z = {1, 2, 3}
  4. print(type(x), type(y), type(z))

OUTPUT:

<class 'list'> <class 'tuple'> <class 'set'>

**14.Can you change a value in a tuple once it is defined? Why or why not?**

No, you **cannot** change a value in a tuple once it’s defined. That’s because tuples are **immutable**, meaning their contents are fixed and can’t be altered directly.

**15.Is reassignment possible for immutable datatypes?**

Yes, reassignment **is possible** for immutable data types—but it's **not modifying** the original object. You’re simply pointing the variable to a **new** object.