**Python Data Types**

Python Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data.

* **Numeric –**[int](https://www.geeksforgeeks.org/python-numbers/), [float](https://www.geeksforgeeks.org/python-float-type-and-its-methods/), [complex](https://www.geeksforgeeks.org/python-complex-function/)
* **Sequence Type –**[string](https://www.geeksforgeeks.org/python-string/), [list](https://www.geeksforgeeks.org/python-lists/), [tuple](https://www.geeksforgeeks.org/python-tuples/)
* **Mapping Type –**[dict](https://www.geeksforgeeks.org/python-dictionary/" \t "_blank)
* **Boolean –**[bool](https://www.geeksforgeeks.org/boolean-data-type-in-python/)
* **Set Type –**[set](https://www.geeksforgeeks.org/python-sets/), [frozenset](https://www.geeksforgeeks.org/frozenset-in-python/" \t "_blank)

**1. Numeric Data Types in Python**

a = 5 //Integer type

print(type(a))

b = 5.0

print(type(b)) //Float type

c = 2 + 4j //Complex number

print(type(c))

**2. Sequence Data Types in Python**

The sequence Data Type in Python is the ordered collection of similar or different Python data types. Sequences allow storing of multiple values in an organized and efficient fashion.

String

List

Tuple

**String Data Type**

Python[Strings](https://www.geeksforgeeks.org/python-strings/)are arrays of bytes representing Unicode characters. In Python, there is no character data type Python, a character is a string of length one. It is represented by str class.

s = 'Welcome to the Geeks World'

print(s)

*# check data type*

print(type(s)) =🡺datatype

*# access string with index*

print(s[1])

print(s[2])

print(s[-1])

**List Data Type**

[Lists](https://www.geeksforgeeks.org/python-list/)are just like arrays, declared in other languages which is an ordered collection of data. It is very flexible as the items in a list do not need to be of the same type.

*# Empty list*

a = []

*# list with int values*

a = [1, 2, 3]

print(a)

*# list with mixed int and string*

b = ["Hello", "For", "Remya", 4, 5]

print(b)

a = ["Hello", "For", "Revature"]

print("Accessing element from the list")

print(a[0])

print(a[2])

print("Accessing element using negative indexing")

print(a[-1]) =🡺 Revature

print(a[-3]) =🡺 Hello

print(a[-4])

**Tuple Data Type**

Just like a list, a [tuple](https://www.geeksforgeeks.org/python-tuples/) is also an ordered collection of Python objects. The only difference between a tuple and a list is that tuples are immutable. Tuples cannot be modified after it is created.

*# initiate empty tuple*

tup1 = ()

tup2 = ('Hello', 'For')

print("**\n**Tuple with the use of String: ", t2)

-----------------------------

tup1 = tuple([1, 2, 3, 4, 5])

*# access tuple items*

print(tup1[0])

print(tup1[-1])

print(tup1[-3])

**3. Boolean Data Type in Python**

**Valid=True**

**Invalid=False**

print(type(**True**))

print(type(**False**))

print(type(true))

**4. Set Data Type in Python**

In Python Data Types, [Set](https://www.geeksforgeeks.org/python-sets/)is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

*# initializing empty set*

s1 = set()

s1 = set("HelloWelcomeHello")

print("Set with the use of String: ", s1) ====🡺

s2 = set(["Hello”, "For", "Hello"])

print("Set with the use of List: ", s2) ====🡺

**🔹 1. Numeric Data Types**

**Python supports integers, floats, and complex numbers.**

**✅ Integer (int)**

**Used for whole numbers.**

**x = 10**

**y = -5**

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

**✅ Floating Point (float)**

**Used for decimal numbers.**

**pi = 3.14**

**gravity = 9.81**

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

**✅ Complex Numbers (complex)**

**Used for numbers with a real and imaginary part.**

**z = 3 + 4j**

**print(type(z)) # Output: <class 'complex'>**

**🔹 2. Sequence Data Types**

**These store multiple items in an ordered manner.**

**✅ Strings (str)**

**Used for text.**

**name = "Python"**

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

**➤ String Operations:**

**name=”Revature”**

**print(name[0]) # First character: P**

**print(name[1:4]) # Slicing: ---🡪 eva 1 3**

**print(name.upper())**

**Task : create a program to mention all the string function**

**✅ Lists (list)**

**Ordered, mutable (changeable) collection.**

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

**fruits.append("orange") # Add an item**

**print(fruits) # ['apple', 'banana', 'cherry', 'orange']**

**Create pgm to implement all the remaining functions in list**

**✅ Tuples (tuple)**

**Ordered, immutable collection.**

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

**print(colors[0]) # red**

**✅ Ranges (range)**

**Represents a sequence of numbers.**

**numbers = range(1, 10, 2) # (start, stop, step)**

**print(list(numbers)) # [1, 3, 5, 7, 9]**

**🔹 3. Set Data Types**

**Stores unique elements.**

**✅ Sets (set)**

**Unordered and does not allow duplicates.**

**unique\_numbers = {1, 2, 3, 4, 4, 5}**

**print(unique\_numbers) # {1, 2, 3, 4, 5}**

**✅ Frozen Set (frozenset)**

**An immutable version of set.**

**fs = frozenset({1, 2, 3})**

**# fs.add(4) # ❌ This will cause an error (immutable)**

**🔹 4. Dictionary Data Type (dict)**

**Stores key-value pairs.**

**student = {"name": "John", "age": 25, "city": "New York"}**

**print(student["name"]) # John**

**student[age]**

**student[city]**

**🔹 5. Boolean Data Type (bool)**

**Used for True or False values.**

**is\_python\_fun = True**

**print(type(is\_python\_fun)) # <class 'bool'>**

**🔹 6. Binary Data Types**

**Used to handle binary data.**

**b =[65, 66, 67]**

**print(b) =============🡺**

**c= bytes([65, 66, 67]) # A, B, C in ASCII**

**print(c) ======================🡺A, B,C**

**ba = bytearray(b)**

**print(ba) # bytearray(b'ABC')**

**🔹 7. NoneType**

**Represents the absence of a value.**

**x = None**

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

**y**

**print(y)**

**print(type(y)) ============🡺?**