Python Data Types

Built-in Data Types

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview
None Type:	NoneType

Getting the Data Type

We can get the data type of any object by using the type() function:

Example

Print the data type of the variable x:

x = 5
print(type(x))
Output:
<class 'int'>

Setting the Data Type

In Python, the data type is set when we assign a value to a variable:

Example	Data Type	
x = "Hello World"	str	x = "Hello World" #display x:
		print(x)
		#display the data type of x: print(type(x))
	Dy. U	Output: Hello World
•	O	<class 'str'=""></class>
x = 20	int x = 20	
	#display x:	
	print(x)	
	#display the data type of x:	

SECURE TO STATE AT NO. OF STATE OF THE SECURE AT THE SECUR	print(type(x))	
	Output:	
	20	
	<class 'int'=""></class>	
x = 20.5	float	x = 20.5
		#display x:
	,x*** /	print(x)
		#display the data type of x:
		print(type(x))
	A141	Output:
		20.5
		<class float'=""></class>
x = 1j	Complex	
	x = 1j	
	#display x:	
	print(x)	
	#display the data type of x:	
	<pre>print(type(x))</pre>	
	Output	

	lj <class 'complex'=""></class>	
x = ["apple", "banana", "cherry"]	List x = ["apple", "banana", "cherry"]	
	#display x: print(x)	
	#display the data type of x:	
	<pre>print(type(x)) Output:</pre>	
	['apple', 'banana', 'cherry']	
	<class 'list'=""></class>	
x = ("apple", "banana", "cherry")	tuple	
x = range(6)	range	

<pre>x = {"name" : "John", "age" : 36}</pre>	dict	
x = {"apple", "banana", "cherry"}	set	
<pre>x = frozenset({"apple", "banana", "cherry"})</pre>	frozenset	
x = True	bool x = True #display x: print(x) #display the data type of x: print(type(x)) Output: True <class 'bool'=""></class>	JM &
x = b"Hello"	bytes	*
x = bytearray(5)	bytearray	

x = memoryview(bytes(5))	memoryview	
x = None	NoneType	

Setting the Specific Data Type

If you want to specify the data type, you can use the following constructor functions:

Example	Data Type Try it
x = str("Hello World")	str
x = int(20)	int
x = float(20.5)	float
x = complex(1j)	complex
x = list(("apple", "banana", "cherry"))	list

<pre>x = tuple(("apple", "banana", "cherry"))</pre>	tuple
x = range(6)	range
x = dict(name="John", age=36)	dict
x = set(("apple", "banana", "cherry"))	set
x = frozenset(("apple", "banana", "cherry"))	frozenset
x = bool(5)	bool
x = bytes(5)	bytes
x = bytearray(5)	bytearray
x = memoryview(bytes(5))	memoryview

Python Numbers

Python Numbers

There are three numeric types in Python;

- int
- float
- complex

Variables of numeric types are created when you assign a value to them:

Example

```
x = 1  # int
y = 2.8  # float
z = 1j  # complex
```

To verify the type of any object in Python, use the type() function:

Example

```
y = 2.8
z = 1j
print(type
```

x = 1

print(type(x))
print(type(y))
print(type(z))

Output: <class 'int'> <class 'float'> <class 'complex'>

Int

 $_{\mbox{Int, or integer, is a whole number, positive or negative, without decimals, of unlimited length.}$

Example

Integers:

x = 1

y = 35656222554887711

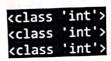
z = -3255522

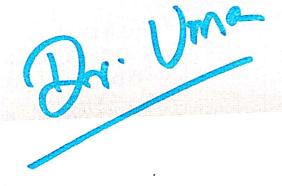
print(type(x))

print(type(y))

print(type(z))

Output:





Float

Float, or "floating point number" is a number, positive or negative, containing one or more decimals.

Example

Floats:

x = 1.10

y = 1.0

z = -35.59

print(type(x))

```
print(type(y))
print(type(z))
```

Float can also be scientific numbers with an "e" to indicate the power of 10.

Example

Floats:

```
x = 35e3
y = 12E4
z = -87.7e100

print(type(x))
print(type(y))
print(type(z))

Output:
<class 'float'>
<class 'float'>
<class 'float'>
```



Complex

Complex numbers are written with a "j" as the imaginary part:

Example

Complex:

```
x = 3+5j
y = 5j
z = -5j

print(type(x))
print(type(y))
print(type(z))
```

```
Output:

<class 'complex'>
<class 'complex'>
<class 'complex'>
```

Type Conversion

You can convert from one type to another with the int(), float(), and complex() methods:

z. Uma

Example

```
Convert from one type to another:
#convert from int to float:
x = float(1)

#convert from float to int:
y = int(2.8)

#convert from int to complex:
z = complex(1)

print(x)
print(y)
print(y)
print(type(x))
print(type(y))
print(type(z))
```

```
1.0
2
(1+0j)
<class 'float'>
<class 'int'>
<class 'complex'>
Note: You cannot convert complex numbers into another number type.
```

Random Number

Python does not have a random() function to make a random number, but Python has a built-in module called random that can be used to make random numbers:

Example

```
Import the random module, and display a random number between 1 and 9:
```

```
import random
print(random.randrange(1, 10))
```

Output:

1