

Data types in Python

A data type represents the type of data stored into a variable or memory.

Built-in data types

1) None type :- In None datatype represents an object that does not contains any value. in languages like java it is called as NULL.

2) Numeric type :- The numeric type represents numbers.

1) int

2) float

3) complex

int datatype :-

int represents integer numbers. An integer number is a number without any decimal point or fraction part. For example :- 200, 50, -50.
etc.

float datatype :- float represents

floating point values for example

Complex datatype :- A complex number is a number that is written in the form of $a + bj$ or $a + bJ$. Here a represents real part and b represents imaginary part. The suffix 'J' or 'j' after b indicates square root value of -1

for example :-

$$c = 3 + 5j \text{ etc.}$$

Converting the datatype Explicitly :-

① $x = 15.56.$

$$\text{int}(x) \rightarrow 15 \text{ Ans.}$$

② $\text{num} = 15$

$$\text{float}(\text{num}) \rightarrow 15.0 \text{ Ans.}$$

③ $n = 10$

$$\text{complex}(n) \rightarrow 10 + 0j$$

or

$$a = 10$$

$$b = -5$$

$$\text{complex}(a, b) \rightarrow 10 - 5j$$

bool datatype

The bool datatype in Python represents boolean values. (True or False) only

$$\text{True} \rightarrow 1$$

$$\text{False} \rightarrow 0$$

A blank string "" also represents as False.

3) Sequences in Python

- a) str
- b) bytes
- c) list
- d) tuple
- e) range

~~##~~ ~~###~~

str datatype :- In python str represents string datatype. A string is represented as group of characters. strings are enclosed in single quotes and double quotes.

for example

str = "welcome"

str = 'welcome'

we can also write string in the following format

str = """ welcome """

str = ''' welcome '''

string as a sequence of characters :

A python string is a sequence of characters and each character can be individually accessed using its index.

The individual elements in string are the characters contained in it (stored in contiguous memory location) and as mentioned the characters of a string are given two-way index for each location.

Let $sub = \text{"COMPUTERS"}$

	0	1	2	3	4	5	6	7	8	
sub =	C	O	M	P	U	T	E	R	S	→ forward indexing
	-9	-8	-7	-6	-5	-4	-3	-2	-1	

↙ Reverse indexing.

Thus $sub[0] = 'C'$ $sub[-9] = 'C'$
 $sub[2] = 'M'$ $sub[-7] = 'M'$

*Note :- You cannot change the individual letters of a string in place by assignment because strings are immutable and hence item assignments is not supported

$name = \text{"hello"}$

$name[0] = 'p'$ — (X) Not allowed.

$name = \text{"hello"}$

$name = \text{"new"}$

2) bytes Data type :-

The bytes datatype represents a group of byte numbers just like an array

A byte number is any positive

for Example

Elements = [10, 20, 0, 40, 15]

x = bytes(Elements)

print(x[0])

→ converts list
into bytes
array.

We can not modify or edit elements
in the byte type array.

for Example

x[0] = 55 gives error.

3) list data type

Lists in python are similar to arrays
in C or java. A list represents a group
of elements. The main difference is
list can store different types of elements
but an array can store only one type
of elements. Also list can grow dyna-
-mically in memory. But the size
of array is fixed it can not grow
dynamically.

list is represented by [] brackets

or
by using list()

for Example

```
list = [10, -20, 15.5, "Vijay"]  
print(list)
```

tuple data type :-

A tuple is similar to a list. A tuple contains a group of elements which can be different types. The tuple elements are separated by commas and enclosed in parentheses ().

```
tpl = (10, -20, 15, 16.5, 'Vijay')
```

Range datatype :- The range datatype represents a sequence of numbers. The numbers in the range are modifiable. Range is used for repeating a for loop for a specific number of times.

for Example

```
r = range(10)
```

```
for r in r : print(i)
```

Sets :- A set is unordered collection of elements much like a set in Mathematics. The order of elements is not maintained in the sets. It means the elements may not appear

in the same order. as they entered into the sets.

Mapping data type :-

A map represents a group of elements in the form of key and value pairs so that when the given key is given, we can retrieve the value associated with it. the dict is an example of a map.

$d = \{ 10 : 'Kamal', 11 : 'Pranav' \}$

Reserved keywords in Python

and	del	from	nonlocal	try
as	elif	global	not	while
assert	else	if	or	with
break	except	import	pass	yield
class	exec	in	print	False
continue	finally	is	raise	True
def	for	lambda	return.	