iNurture
Education Solutions
TOMORROW'S HERE

Subject: Python Programming

Module Number: 1.2

Module Name: Built-in Data Type in Python

Version Code:PP4, PP5
Released Date: 4-OCT-2019

Introduction to Python

Syllabus:

- Introduction: Introduction to Python, Setting up the environment, Installing Python, Running python program, Python's execution model, Guidelines on how to write good, The Python culture, A note on the IDEs
- **Built-in Data Types: Numbers, Immutable sequences, Mutable sequences, Set types**
- Mapping types – dictionaries, The collections module, Final considerations
- Iterating and Making Decisions: Conditional programming, Looping, Putting this all together.

Data Types in Python

Aim:

To discuss about the Data Types in Python

Data Types in Python

Objectives

The Objectives of this module are to:

To learn about different data types used in Python

To learn about different methods associated with data types

Data Types in Python

Outcomes

At the end of the module, you are expected to :

- Understand about the data types used in Python
- Student will able to understand about the methods associated with Data types

Data Types in Python

CONTENT :

- To discuss about the Mutable Vs Immutable Sequence
- To discuss about different data types used in Python
- To discuss about the different methods associated with the data types

Data Types in Python

Built-in Data types used in Python are listed below :-

- Numbers
- String
- Tuple
- Set
- List
- Dictionaries

Data Types in Python

Numeric Data Type

- Immutable Sequence: Everything in Python is an object. The value of Immutable objects cannot be changed once it is created.

Example: Numbers, Tuples, String, Set

- Mutable Sequence: The value of objects can be changed once it is created.

Example: List, Dictionary

Data Types in Python

Integer Data Type

Integers numbers can be positive, negative numbers or zero without any decimal point.

They support all the basic mathematical operation.

Example 1 :-

```
a = 2
b = 3278686496482564759453763
c = -7534765
print ( type ( a ) )
print ( type ( b ) )
print ( type ( c ) )
print ( b )
```

Output :-

```
<class ‘int’>
<class ‘int’>
<class ‘int’>
3278686496482564759453763
```

Data Types in Python

Arithmetic Operation

Integers numbers can be positive, negative numbers or zero without any decimal point. They support all the basic mathematical operation.

Example :-

```
>>>a = 3
>>>b =2
>>> a + b           # addition
5
>>> a – b           # subtraction
1
>>> a * b           # multiplication
6
 >>> a / b  # true division
1.5
>>> a // b   # integer division
 1
>>> a % b # modulo operation (reminder of division)
 1
 >>> a ** b         # power operation
9
```

Data Types in Python

Real Data Type

- Real numbers or “floating point numbers” is a number where float values are specified with a decimal point. These can be positive or negative.
- The character ‘e’ or ‘E’ followed by a positive or negative integer may be appended to specify scientific notations.

Example:- a = 4e7 , x = -5e8 , y = 0.7e3 , etc.

- E or e indicating the power of 10 (2.5e2 = 2.5 x 10^2 = 250).

Data Types in Python

Real Data Type

Example:-

```
a = .2
b = 4.5
c = 5.
d = 4e7
print ( type ( a ) )
print( a )
print( c )
print( d )
```

Output :-

```
<class ‘float’>
0.2
5.0
40000000
```

Data Types in Python

Boolean Data Type

It refers to Boolean whose value is either True or False (i.e., 1 or 0) only.

It is basically used for decision making as shown in the below example. This data type belongs to class ‘bool’.

Example :-

```
P = 5
Q = 7
xyz = P < Q
print(xyz)
print(type(xyz))
```

Output :-

```
True
<class ‘bool’>
```

Data Types in Python

Complex Data Type

Python supports complex number. Complex numbers are the numbers which are expressed in the form of a+jb where a, b are the real numbers and j is the imaginary number (j=square root of -1)

Example :-

```
a = 3 + 5j
print ( type ( a ) )
print ( a )
```

Output :-

```
<class ‘complex’>
3+5j
```

Data Types in Python

String Data Type

String literals in python are surrounded by either single quotation marks, or double quotation marks. ‘hello' is the same as "hello".

Strings are immutable sequences of Unicode code points. Unicode code points can represent a character, but can also have other meanings, such as formatting data. Example- Python, unlike other languages, does not have a char type, so a single character is rendered simply by a string of length 1.

Python treats single quotes the same as double quotes. Creating strings is as simple as assigning a value to a variable.

Data Types in Python

String Data Type

Example of String:	Output:
var1 = 'Hello World'	Hello World
var2="Programming Concepts"	Programming Concepts
print(var1)	<class 'str'>
print(var2)	
print(type(var1))	

Data Types in Python

String Methods

Some of the Strings functions are listed below :-

String Operation	Description	Example

len	The len() method returns the length of a string.	a = "hello world“
		print(len(a))
		Output : 11

lower()	The lower() method returns the string in lower case.	a = “Hello,world"
		print(a.lower())
		Output : hello,world

upper()	The upper() method returns the string in upper case.	a = “Hello,world“
		print(a.upper())
		Output:HELLO,WORLD

Data Types in Python

String Methods

String Operation	Description	Example

replace()	The replace() method replaces a string with another string.	a = “Hello,world“
		print(a.replace("H", "J"))
		Output: Jello,world

split()	The split() method splits the string into substrings if it finds instances of the separator.	a = “Hello,World“
		print(a.split(","))
		Output: Hello
		world

Data Types in Python

String Methods

String Operation	Description	Example

capitalize()	Capitalizes first letter of a string	a = "hello,world"
		print(a.capitalize())
		Output: Hello,world

strip()	Removes the white spaces from the beginning and the end.	a=" hello world “
		print(a.strip())
		Output:helloworld

Data Types in Python

String Methods

String Functions	Description	Example
lstrip	Removes all leading whitespace in string.	a=“ __hello world__” print(a.lstrip(“_”)) **Output**:helloworld__
rstrip	Removes all trailing whitespace of string.	a=" __hello world__” print(a.rstrip(“_”)) **Output**:__helloworld
find()	Displays the starting index of the substring to be found, if it is not present then it displays -1.	a=" hello world” print(a.find(“h”)) **Output**:1

Data Types in Python

String Formatting

Strings and numbers can be combined by using format() method as shown in the below example:

Example :

```
age = 36

txt = "My name is John, and I am {}"

print(txt.format(age))
```

Output: My name is John, and I am 36

Data Types in Python

Tuple Data Type

A tuple is a collection which is ordered, indexed and **unchangeable**. In Python tuples are written with round brackets.

Once a tuple is created, you cannot change its values. Tuples are **unchangeable**.

Example :

fruit =(“Mango”,Grapes”)

Access Tuple Items - You can access tuple items by referring to the index number, inside square brackets

Example: print(fruit[1])

Allows duplicate members.

Data Types in Python

Tuple Data Type

Example :-

```
fruit = ("apple", "banana", "cherry")
print (fruit)
print ( len ( fruit ))
print (fruit[1] )
```

Output :-

```
('apple' , 'banana' , 'cherry')
3
banana
```

Data Types in Python

Tuple Methods

Some of the tuple operation are listed below :-

Tuple Operation	Description	Example

len	The len() method returns the length of a string.	fruit = ("apple","banana","cherry")
		Print(len(fruit))
		Output: 3

count()	Returns the number of times a specified value occurs in a tuple.	fruit = ("apple","banana","cherry")
		print(fruit.count("apple"))
		Output: 1

index()	Searches the tuple for a specified value and returns the position of where it was found.	fruit = ("apple","banana","cherry")
		print(fruit.index("apple"))
		Output: 0

Data Types in Python

Lists Data Type

List is a collection which is ordered, indexed and changeable. Allows duplicate members.

Lists are represented using square brackets [ ].

It allows duplicate members.

String input should be in double quotes while output is in single quotes.

Example:- fruit=["apple","banana”,"cherry"]

To access the items from the list, index is used.

Example: print(fruit[1])

Data Types in Python

List Data Type Example

Example:-

```
List = [ 1 , 1 , “hello” , “world” ,34 ]
```

Example:-

```
List = [ 1 , 1 , “hello” , “world” ,34 ]
print(List)
```

Output:-

```
[ 1 , 1 , ‘hello’ , ‘world’ ,34 ]
```

Data Types in Python

List Methods

Some of the list methods are listed below :-

List Operation	Description	Example
len	The len() method returns the length of a set	fruit = {"apple","banana","cherry"} Print(len(fruit)) **Output**: 3
insert()	To add an item at the specified index	fruit.insert(1, "orange")
remove()	To remove the specified item	fruit.remove("orange")
pop()	To remove the specified index, (or the last item if index is not specified)	fruit.pop("orange") or fruit.pop()

Data Types in Python

List Methods

List Operation	Description	Example
del	The del keyword removes the specified index. The del keyword can also delete the list completely.	del fruit(“orange”) del fruit
clear()	The clear() method empties the list	fruit.clear()
extend()	This adds the contents of sequence to existing list	fruit.extend([“grapes”,”berry”])
index()	This returns the index of the specified element in the list	print(fruit.index("apple"))

reverse()	Reverses objects of list in place	fruit.reverse()
		print(fruit)

Data Types in Python

Set Data Type

A set is a collection which is unordered and unindexed. In Python, sets are written with curly {} brackets.

Sets are unordered, so the items will appear in a random order. No duplicate members are allowed in set.

Since it is unordered, so on popping elements of a set, any element can be removed from the set.

Example :- fruit={“mango”,”apple”}

Once a set is created, you cannot change its items, but you can add new items.

Data Types in Python

Set Data Type

You cannot access items in a set by referring to an index, since sets are unordered the items has no index.

But you can loop through the set items using a for loop, or ask if a specified value is present in a set, by using keyword.

Example :

```
fruit = {"apple", "banana", "cherry"}
for x in fruit:
    print(x)
```

Out put -> apple

```
banana
cherry
```

Data Types in Python

Set Methods

Set Operation	Description	Example

len	The len() method returns the length of a set.	fruit = {"apple","banana","cherry“}
		Print(len(fruit))
		Output: 3

add()	To add one item to a set use the add() method.	fruit.add(“orange”)
		Output : {'apple', 'banana', 'cherry', 'orange'}

update()	To add more than one item to a set use the update() method.	fruit.update([“Grapes“,”Mango”])
		Output :{'banana', 'Grapes', 'apple', 'Mango', 'cherry'}

Data Types in Python

Set Methods

Set Operation	Description	Example

remove()	This method removes the specified item from the set.	fruit = {"apple", "banana", "cherry"}
		fruit.remove(“orange”)
		Print(fruit)
		Output: {'apple', 'banana'}

pop()	This method removes the item from the set as set is unordered hence any item can be removed from the set.	fruit = {"apple", "banana", "cherry"}
		fruit.pop()
		Print(fruit)
		Output: {'banana', 'cherry'}
		Note: apple is removed

Data Types in Python

Set Methods

Set Operation	Description	Example

union()	Returns a set that contains all items from both sets, duplicates are excluded.	set1 ={1,2,3}
		set2={1,4,5,6}
		set3=set1.union(set2)
		print(set3)
		Output:{1, 2, 3, 4, 5, 6}

intersection()	Returns a set that contains the items that exist in both set x, and set y.	set1 ={1,2,3}
		set2={1,4,5,6}
		set3=set1.intersection(set2)
		print(set3)
		Output:{1}

Data Types in Python

Set Methods

Set Operation	Description	Example

difference	Returns a set that contains the items that only exist in set x, and not in set y.	set1 ={1,2,3}
		set2={1,4,5,6}
		set3=set1.difference(set2)
		print(set3)
		Output:{2,3}

Data Types in Python

Dictionary Data Type

- A dictionary is a collection which is unordered, changeable and indexed.
- In Python dictionaries are written with curly brackets, and they have keys and values.

Example: vehicle ={“brand”:”ford”,”model”:”Mustang”,”year”:1991}

- Items in the dictionaries can be accessed using key

Example: x=vehicle[“model”] or x =vehicle.get("model")

- You can change the value of a specific item by referring to its key name: **vehicle["year"] = 2018**

Data Types in Python

Dictionary Data Type

Example :-

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict)
print( thisdict[“model”] )
thisdict[“year”]=2019
print(len(thisdict))
thisdict.pop(“model”)
thisdict[“colour”]=“red”
print(thisdict)
```

Output :-

```
{‘brand’:’Ford’,’model’:’Mustang’,’year’:1964}

‘Mustang

3

{‘brand’:’Ford’,’year’:2019,’colour’:’red’}
```

Data Types in Python

Dictionary Methods

Dictionary Operation	Description	Example

len()	Determines how many items (key-value pairs) a dictionary has.	vehicle ={"brand":"ford","model":"Mustang","year":1991}
		print(len(vehicle))
		Output:3

Adding item to dictionaries	Adding an item to the dictionary is done by using a new index key and assigning a value to it.	vehicle ={"brand":"ford","model":"Mustang","year":1991}
		vehicle["color"]="red"
		print(vehicle)
		Output:{'brand': 'ford', 'model': 'Mustang', 'year': 1991, 'color': 'red'}

Data Types in Python

Dictionary Methods

Dictionary Operation	Description	Example

pop()	This method removes the item with the specified key name.	vehicle ={"brand":"ford","model":"Mustang","year":1991}
		vehicle.pop("model")
		print(vehicle)
		Output:{'brand': 'ford', 'year': 1991}

popitem()	This method removes the last inserted item.	vehicle ={"brand":"ford","model":"Mustang","year":1991}
		vehicle.popitem()
		print(vehicle)
		Output:{'brand': 'ford', 'model': 'Mustang‘}

Data Types in Python

Dictionary Methods

Dictionary Operation	Description	Example

del	The del keyword removes the item with the specified key name
	
	The del keyword can also delete the dictionary completely (del <variablename>)
		vehicle ={"brand":"ford","model":"Mustang","year":1991}
		del vehicle("model")
		print(vehicle)
		Output:{'brand': 'ford', 'year': 1991}

clear()	This method empties the dictionary	vehicle.clear()

Data Types in Python

Self Assessment Question

1. Which one of these in not a core data type?
    1. Lists
    2. Dictionary
    3. Tuples
    4. Class

Answer: Class

Data Types in Python

Self Assessment Question

Which one of the given options will be the oputput, if the following set of commands are executed in shell?

```
>>>str="hello"
>>>str[:2]
>>>
```

a. He

b. Lo

c. Olleh

d. hello

Answer: He

Data Types in Python

Assignment

1. Write a python code to find mean and standard deviation of a given list of numbers.
2. Write a python code to add and delete element from a dictionary using functions.

Data Types in Python

Summary

- Discussed about the different data types in Python
- Discussed about the methods in List, String, Set, Tuple, Dictionaries with example

Data Types in Python

Document Links

Topics	URL	Notes
Built in Data Type in Python	https://www.programiz.com/python-programming/variables-datatypes	This link explains data type used in python
Numbers	https://www.w3schools.com/python/python_numbers.asp	This link explains about the number data type in python
String	https://www.w3schools.com/python/python_strings.asp	This link explains about the string data type in python
List	https://www.w3schools.com/python/python_lists.asp	This link explain about the list data type in python
Tuple	https://www.w3schools.com/python/python_tuples.asp	This link explain about the tuple data type in python
Set	https://www.w3schools.com/python/python_sets.asp	This link explain about the set data type in python
Dictionaries	https://www.w3schools.com/python/python_dictionaries.asp	This link explains about the dictionaries data type in python

Data Types in Python

Video Links

Topics	URL	Notes
Data Type in python	https://www.youtube.com/watch?v=gCCVsvgR2KU	This link explains about the data types in Python