19/06/2020

PYTHON ASSIGNMENT

Assignment Name :→

321810304042 - Python - Assignment - 2

Name : Nanjundareddy Bhavyasree

year : 2nd year

Section : Section 'D' section

RNo. : 321810304042

19/06/2020

3218103040428 - Python - Assignment - 2

(1) What the Data types in Python ? Explain ?

Ans :-

DATA TYPES IN PYTHON :-

Variables can hold values of different data types. i.e., for example, a person's name must be stored as a string whereas its id must be stored as an integer.

Python provides various standard datatypes that define the storage method on each of them. some of the important types are listed below.

(1) Numbers
(2) string
(3) List
(4) Tuple
(5) Dictionary

(1) Numbers :->

* Number stores Numeric values.
* Integers, floating point numbers and complex numbers fall under Python Numbers category.

* they are defined as

→ int : signed integers like 10, 20, .. etc.

→ float : float is used to store floating point numbers like 0.9, 9.9, 15.5 etc.

→ Complex : complex numbers like 2.4j, 2.0 + 2.3j etc.

Example :-

① a = 50
print(a)
print(type(a))

output :-
50
<class 'int'>

② a = 2.0
print(a, type(a))

o/p :-
2.0 <class 'float'>

③ a = 1+2j
print(type(a), a)

o/p :-
<class 'complex'>(1+2j)

④ String :-

* String is sequence of Unicode characters which are represented in quotation marks i.e., "PYTHON".

* In python, we can use single (' '), double (" ") quotes.

* Multi-Line strings can be denoted using Triple quotes (''' ''') or (""" """).

Example :-

(1) s1 = " hello "
s2 = " good morning "
print (s1, s2)
print (type(s1), type(s2))

o/p :-
hello good morning
<class 'str'>
<class 'str'>

(2) s = "hello world"
print (s[0])
print (s[4])
print (s[:-1])
print (s[1:])

o/p :- "h"
"o"
"hello worl"
"ello world"

(3) List :-→

* List is an Ordered sequence of items.
* It is one of the most used datatype in python and is very flexible.
* All the items in a list do not need to be of the same data type.
* the items stored in the list are seperated by comma (,) and enclosed within square brackets [].
* we can use slice operators [:] to access the data of the list.

Example :-

a = [5, 10, 15, 20, 25, 30, 35, 40]
print (" a[2] = ", a[2])

```
print("a[0:3] = ", a[0:3])
print("a[5:] = ", a[5:])
```

o/p :- a[2] = 15
a[0:3] = [5, 10, 15]
a[5:] = [30, 35, 40]

④ Tuple :-

* Tuple is an ordered sequence of items same as a list.
* The only difference is that tuples are immutable.
* Tuples once created cannot be modified the size and value of the item.
* Tuple is a Read-Only data structure.
* It is defined within parentheses ( ) where items are separated by commas.

Example :-

```
t = ("hi", "python", 2)
print("t[1] = ", t[1])
print("t[1:] = ", t[1:])
print("t[0:1] = ", t[0:1])
print(type(t))
```

o/p :- t[1] = (python)
t[1:] = ('python', 2)
t[0:1] = ('hi',)
<type 'tuple'>

⑤ Dictionary :→

* Dictionary is an unordered collection of key-value pairs.
* Dictionaries are used / optimized for retrieving data.
* In Python, dictionaries are defined within braces { }

Example :-

```
d = {1: 'Jimmy', 2: 'Alex', 3: 'john',
     4: 'mike'};
print(d);
print(d.keys());
print(d.values());
```

o/p :- {1: 'Jimmy', 2: 'Alex', 3: 'john',
4: 'mike'}

[1, 2, 3, 4]

['Jimmy', 'Alex', 'john', 'mike']

⑥ Briefly explain History of Python ?

Ans :-

HISTORY OF PYTHON :-

Python is a widely used general-purpose, high-level programming language. It was initially designed by "GUIDO VAN ROSSUM" in 1991 and developed by Python software

foundation. It was mainly developed for Emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Let's dig deeper :→

In the late 1980s, history was about to be written. It was that time when working on python started. soon after that, Guido Van Rossum began doing its applications based work in december of 1989 by at Centrum Wiskunde and Informatica (CWI) which is situated in Netherland.

It was first started as a Hobby project because he was looking for an interesting project to keep him occumpied during Christmas. The Programming language which python is said to have succeeded is ABC programming Language, which had the interfacing with the Amoeba os and had the feature of exception Handling. He had already helped to create ABC earlier in his career and he had seen some issues with ABC but liked the most of the features.

The language was finally released in

1991. when it was released, it used a lot fewer codes to express the concepts, when the we compare it with java, c++ and c. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. when it was released it had more than enough capability to provide classes with inheritance, several core data types exception handling and functions.

Year and versions of PYTHON :-

1) PYTHON 1.0 in January, 1994
2) PYTHON 1.5 in December, 1997
3) PYTHON 1.6 in september, 2000
4) PYTHON 2.0 in October, 2000
5) PYTHON 2.2 in December, 2001
6) PYTHON 2.1 in April, 2001
7) PYTHON 2.3 in july, 2003
8) PYTHON 2.4 in November, 2004
9) PYTHON 2.5 in september, 2006.
10) PYTHON 2.6 in October, 2008
11) PYTHON 2.7 in July, 2010
12) PYTHON 3.0 in December, 2008.
13) PYTHON 3.1 in June, 2009
14) PYTHON 3.2 in february, 2011.

15) PYTHON 3.4 in March, 2014.
16) PYTHON 3.3 in September, 2012
17) PYTHON 3.5 in September, 2015
18) PYTHON 3.6 in December, 2016
19) PYTHON 3.7 in June, 2018
20) PYTHON 3.7.3 in March, 2019

These all are the illustrations of different versions of Python along with timeline.

(3) Explain all the Operators in Python?

Ans :-

BASIC OPERATORS IN PYTHON :-

1) Arithmetic operator
2) Bitwise operator
3) Comparision operator
4) Assignment operator
5) Logical operator.
6) Membership operator } special operators.
7) Identity operator.

(a) Arithmetic Operators :->

Arithmetic operators are used to perform mathematical operations like Addition, Division, Subtraction, Multiplication, floor division, Exponentiation.

Operator	Syntax
+ (Addition)	x+y
- (Subtraction)	x-y

* (multiplication)	x * y
/ (division)	x / y
// (floor division)	x // y
% (modulus)	x % y
** (exponent)	x ** y.

Example :-

```
a = 5
b = 5
add = a+b
print (add)
sub = a-b
print (sub)
mul = a*b
print (mul)
divl = a/b
print (a/b)
div2 = a//b
print (a//b)
mod = a%b
print ( mod)
p = a**b
print (p)
```

output :-

```
10
0
25
1.0
1
0
3125
```

(b) Relational Operator :→

Relational operators compares the values. It either returns True (or) False according

to the condition.

operator	Syntax
>	x > y
<	x < y
==	x == y.
!=	x != y
>=	x >= y
<=	x <= y.

Example :-

```
a = 13
b = 33
print (a>b)
print (a<b)
print (a==b)
print (a!=b)
print (a>=b)
print (a<=b)
```

output :-

```
false
True
false
True
false
True.
```

(c) Logical Operators :->

Logical operators perform Logical AND, Logical OR and Logical NOT Performs.

Operator	Syntax
and	x and y
or	x or y
not	not(x and y)

Example :-

```
a = True
b = False
print(a and b)
print(a or b)
print(not a)
```

o/p :-

```
False
True
False
```

(d) Bitwise Operator :-→

Bitwise operator acts on bits and perform bit by bit operation.

operator	syntax
&	x & y
|	x | y
~	x ~ y
^	x ^ y
>>	x >> y
<<	x << y

Example :-

```
a = 10
b = 4
print(a & b)
print(a | b)
print(~a)
print(a ^ b)
print(a >> 2)
print(a << 2)
```

o/p :-

```
0
14
-11
14
2
40
```

(e) Assignment Operator :-→

Assignment operators are used to assign

values to the variables.

operator	syntax
=	x = y + z
+=	a += b , a = a + b
-=	a = a - b
*=	a = a * b
/=	a = a / b
%=	a = a % b
//=	a = a // b
**=	a = a ** b
&=	a = a & b
|=	a = a | b
^=	a = a ^ b
>>=	a = a >> b
<<=	a = a << b

Example (f) Special Operators →

Identity operator :-

is and is not are the identity operators both are used to check if two values are located on the same part of the memory

Eg :- a1 = 3
b1 = 3
a2 = 'PYTHON'
b2 = 'PYTHON'
a3 = [1, 2]
b3 = [1, 2]

```
print (a1 is not b1)        o/p :-
print (a2 is b2)            False
print (a3 is b3)            True
                            False
```

Membership operator :-

in and not in are the membership operators used to test whether a value or variable is in a sequence.

Example :-

```
x = 'python'                o/p :-
y = {3: 'a', 4: 'b'}        True
print ('G' in x)            True
print ('python' not in x)   false
print ('PYTHON' not in x)   True
print (3 in y)              false.
print ('b' in y)
```

(4) Explain the features of python ?

Ans :-

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming.

* In python, we don't need to declare the type of variable because it is a dynamic typed language.

Eg, x = 10

here, x can be anything such as string, int etc.

Features in Python :-

(1) Easy to code :-

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, jara etc. It is very easy to code. It is also developer friendly language.

(2) Free and Open Source :-

Python language is freely available at the official website and you can download it from the given download link.

It is open-source, this means that the source code is also available to the public.

(3) Object-oriented Language :-

One of the key features of python is OOP. python supports oops and concepts of classes, object encapsulation etc.

(4) GUI programming Support :-

GUI can be made using a module such as PyQt5, PyQt4, wxPython or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.

(5) High-level Language :-

Python is a high-level language. When we write programs in python, we need to manage the memory and do not need to remember system architecture.

(6) Extensible feature :-

Python is a extensible language. We can write our some python code into C (or) C++ language and also can compile.

(7) Python is portable Language :-

Python language is also a portable language. We can run the code on any platform.

(8) Python is Integrated Language :-

Because, we can easily integrated python with other language like C, C++ etc.

(9) Interpreted Language :-

Python is an ~~Integra~~ interpreted language because python code is executed line by line at a time.

(10) Large standard Library :-

python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing.

(11) Dynamically typed Language :-

That means the type for a variable is decided at run time not in advance. because of this feature we don't need to specify the type of variable.

(5) Justify why python is interactive interpreted Language ?

Ans :-

Interpreted Language :->

unlike c/c++ etc, python is an interpreted object-oriented paogaamming language. By interpreted it is meant that each time a progaam is run the interpreter checks through code for errors and then interprets the instructions into machine-readable byte code.

Interactive Language :->

python is interactive. when a python statement is entered, and is followed by the return key, if appropriate, the result will be printed on the screen, immediately, in the next Line. In intera-ctive mode of operation, python is used

In a similar way as the Unix command line (or) the terminal. Interactive python is very useful and helpul for the debugging purpose. It simply returns the >>> prompt (or) the corresponding output of the statement if appropriate and returns error for incorrect statements.

Hence, python is Interactive - Interpreted Language.

Name : NanjundaReddy Bhavyasree
year : 2nd year
sec : 'D' sec
RNo. : 321810304042

Assignment :

321810304042-Python-Assignment-2.