

PYTHON ASSIGNMENTS

Q) what are data types in Python? Explain

A) Python is a dynamically typed language hence we need not define the type of variable by declaring it.

Python provides us the type() function which returns the type of variable passed.

The Python data types are as follows:

1) Numbers.

2) String

3) List

4) Tuple.

5) Dictionary

Q) Numbers :- Number stores numeric values. Python creates number objects when a number is assigned to a variable.

Python supports 4 types of numeric data.

1) int

2) long

3) float

4) complex.

Q) String :- the string can be defined as sequence of characters represented in quotation marks.

In Python, we can use single, double or triple quotes to define a string.

String handling in Python is a straight forward task since there are various built-in functions and operators provided.

In the case of string handling, the operator + is used to concatenate two strings as the operation "Hello" + "Python" returns "Hello Python". The operator * is known as repetition operator as the operation "Python" * 2 returns "Python Python".

i) List:-

Lists are similar to arrays in C, however; the list can contain data of different types. The items stored in the list are separated with comma(,) enclosed together in square brackets [].

We can use slice [:] operators to access the data of the list. The Concatenation operator (+) and repetition operator (*) works with the list in the same way as they were working with the string.

ii) Tuple:-

A tuple is similar to the list in many ways. Like lists, tuple also contains the collection of the items of different data types. The items of the tuple are separated with a comma(,) and enclosed in parentheses ().

A tuple is a read-only data structures as we can't modify the size and value of the items of a tuple.

Dictionary:-

Dictionary is an ordered set of a Key-Value pair of items. It is like an associative array or a hash table where each key stores a specific value. Key can hold any primitive datatype whereas Value is an arbitrary Python object.

The items in the dictionary are separated with the comma and enclosed in the curly braces.

2) What is the history of Python?

A) Python was conceived in late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica in the Netherlands as a successor to the ABC language; capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989.

Van Rossum shouldered sole responsibility for the project, as the lead developer until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator for Life.

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector & support for unicode.

Python 3.0 was released on 3 December 2008. It was major revision of language that is not completely backward-compatible.

Python 2.7's end-of-life date was initially set at 2015, then postponed to 2020 out of concern that a large body

of what

1) Explain all operators in Python?

2) Arithmetic operators:-

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication & division.

operator, meaning.

example

+ Add two operands of unary $x + y + z$
Plus

- Subtract right operand from $x - y - z$
the left or unary minus

*

* Multiply two operands $x * y$

/ Divide left operand by the x / y
right one.

% Modulus - remainder of the $x \% y$
division of left operand by right.

// Floor division - division that results $x // y$
into whole number adjusted to
the left in number line.

** Exponent - left operand wrapped $x ** y$
to the power of right

3) Comparison operators:-

Comparison operators are used to compare values.
It returns either **True** or **False** according to
the condition.

<u>Operator</u>	<u>Meaning</u>	<u>Example</u>
>	Greater than - True if left operand	$x > y$
<	Less than - True if left operand	$x < y$
	is less than the right	
$\geq =$	equal to - True if both operands	$x = z \vee y$
	are not equal	
\neq	Not equal to - True if operands	$x \neq y$
	are not equal	
\geq	Greater than or equal to - True if	$x \geq y$
	left operand is greater than or	
	equal to the right	
\leq	Less than or equal to - True if	$x \leq y$
	left operand is less than or equal	
	to the right	

- iii) Logical Operators :-

Logical operators are the and, or, not operators.

<u>Operator</u>	<u>Meaning</u>	<u>Example</u>
and	True if both the operands	x and y
	are true	
or	True if either of the	x or y
	operands is true	
not	True if operand is false	$\text{NOT } x$

- iv) Bitwise operators !-

Bitwise operators act on operands as if they were strings of binary digits. They operate bit by bit, hence the name.

<u>operator</u>	<u>Meaning</u>	<u>Example</u>
&	Bitwise AND	$x \& y = 0$.
	Bitwise OR	$x y = 14$.
~	Bitwise NOT	$-x = -11$
^	Bitwise XOR	$x ^ y = 14$
>>	Bitwise right shift	$x >> 2 = 2$
<<	Bitwise left shift	$x << 2 = 40$.

) Assignment Operators :-

Assignment operators are used in python to assign values to variables.

<u>operator</u>	<u>Example</u>	<u>Equivalent to</u>
=	$x = 5$	$x = 5$
+=	$x + = 5$	$x = x + 5$
-=	$x - = 5$	$x = x - 5$
*=	$x * = 5$	$x = x * 5$
/=	$x / = 5$	$x = x / 5$
%=	$x \% = 5$	$x = x \% 5$
//=	$x // = 5$	$x = x // 5$
**=	$x ** = 5$	$x = x ** 5$
&=	$x \& = 5$	$x = x \& 5$
=	$x = 5$	$x = x 5$
^=	$x ^ = 5$	$x = x ^ 5$
>>=	$x >> = 5$	$x = x >> 5$

) Special Operators:-

Python language offers some special types of operators like the identity operator or the membership operator. They are described below

Identity Operators - is and is not are Identity Operators

Operator Meaning Example

is

True if the operands are

x . is True

Identical

is not

True if the operands are ~~x~~ is not,
NOT Identical

Membership Operators :-

in and not in are the membership operators
in Python. They are used to test whether a value
is found in a sequence.

Operator

meaning

example

in

True if Value/variable is
found in Sequence

5 in x

not in

True if Value/variable is
not found in the Sequence

~~5 not in x~~

5 not in x

4) Explain features of Python?

A) Python provides following features.

1) Easy to learn and use:- It is developer friendly
and high level programming language.

2) Expressive language:- Python language is
more expressive means that it is more understandible
and readable.

3) Interpreted language:- Interpreted executes
the code line by line, at a time.

4) Cross-platform language:- Python can run on
different platforms such as windows, Linux,
Unix and macintosh etc.

5) Free and open source:- Python language is
freely available at official web address, the
source-code is also available.

6) Object - Oriented Language! - Python supports object oriented language and concepts of class and objects come into existence.

7) Extensible! - It implies that other languages such as C/C++ can be used to complete the code and thus it can be used further in our Python code.

8) Large Standard Library! - Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI Programming Support! - Graphical user interfaces can be developed using Python.

10) Integrated:-

It can be easily integrated with languages like C, C++, JAVA etc.

5) Justify why Python is interactive interpreted language.

1) 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. This is particularly advantageous in debugging process. In interactive mode of operation, Python is used in a similar way as the Unix command line or the terminal.

Interactive Python is very much helpful for the debugging process. It simply returns the prompt or the corresponding output of the statement if appropriate and returns error for incorrect statements.