

ASSIGNMENT - 1

1. What are the data types in python? Explain

Data types are the classification or categorization of data items. Data types represent a kind of value which determines what operations can be performed on that data.

Python contains the following data types.

• Numeric • Non Numeric • Boolean.

* ~~Numeric~~ The five standard Data types are.

• Numbers • Tuple
• String • Dictionary
• List

• Numbers → int ex: $a = 10$

long ex: $a = 345L$ [long integer]

float ex: $a = 10.56$

complex ex: $a = 2 + 10j$

• List → contains series of values. List variables are declared by using `[]`. ex: $A = [1, 2, 3, 'Hello']$.

• Tuple → It is a group of values like a list and are manipulated in similar ways. These are fixed size and defined by `()`. ex: $A = ('Hi', 'Hello', 'gm')$

• Dictionary. \rightarrow List of key: value pairs

They are created by $\{ \}$, separated by $,$ and values associated with :

ex: marks = $\{ \text{'John': 94, 'tom': 96} \}$.

② Briefly explain history of Python.

Python was conceived in the late 1980's by Guido Van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language capable of exception handling and interfacing with the Amoeba operating system. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. It is highly interactive language and high-level language.

Python is named after a TV show called "Monty Python's Flying Circus" and not after Python the snake.

Python is a successor of ABC language

To overcome disadvantage of ABC, the python language was developed.

③ Explain all the Operators in Python

Python operator is a symbol that performs an operation on one or more operands. It consists of 4 types.

1. Arithmetic Operator

consists of basic mathematical operators $+$, $-$, $*$, $/$, $++$, $//$, $\%$. ex: $a+b$, $a-b$, $a//b$.

2. Relational Operator:

carries out the comparison between operands

Less than ($<$) ex: $3 < 4$. %p True

greater than ($>$) ex: $4 > 8$ %p False

Less than or equal to ($<=$) ex: $7 <= 7$ %p True.

greater than or equal to ($>=$) ex: $0 >= 7$ %p False.

Equal to ($==$) ex: $1 == 1$ True

$0 == 1$ False

Not equal to ($!=$)

ex: $1 != 1$ %p False.

3. Assignment Operator.

assigns a value to a variable.

$=$, $+=$, $-=$, $/=$, $*=$, $\%=$, $**=$, $//=$.

4. Logical Operator.

used to combine more than one condition.

and, or, not. operators.

• and ex: $a = 7 > 7$ and $2 > -1$ o/p: False.
 print(a)

• or ex: $a = 7 > 7$ or $2 > -1$ o/p: True.
 print(a)

• not ex: $a = \text{not } (0)$ o/p: True.
 print(a)

5. Membership Operator.

tests whether a value is a member of a sequence. it may be a list, string or tuple.

• in → checks if a value is a member of sequence.

ex: pets = ['dog', 'cat']
 'goat' in pets. o/p: False.

• not in → checks if a value is not a member of sequence.

6. Identity Operator.

checks if the two operands share an identity. 'is' and 'is not'.

7. Bitwise Operator.

operates bit by bit.

• Binary AND (&), Binary OR (|), Binary XOR (^)

• Binary ones complement (~), left shift (<<), right shift (>>)

4. Explain the features of Python?

1. Easy to learn and use.

it is developer-friendly and high-level programming language.

2. Expressive language.

more understandable and readable.

3. Interpreted

interpreter executes the code by line at a time.

This makes debugging easy and suitable.

4. Cross-platform.

- can run equally on different platforms

such as Windows, Linux, Unix and Macintosh, hence.

it is portable.

5. Free and Open Source

freely available at official web address.

6. Object-Oriented.

supports concepts of class and objects.

7. Extensible

other language can be used to compile the code and can be further use in python code.

8. Large standard library.

provides rich set of modules and functions.

9. GUI Programming support

GUI 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.

python is an interpreted language because code is executed line by line at a time, there is no need to compile code, this makes it easier to debug code. The source code of python is converted into an immediate form called bytecode. and Python is interactive.

when a 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 the debugging process.