1. **What are the data types in python ? Explain ?**

Python Data Types:-

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

Following are the standard or built-in data type of Python:

* [Numeric](https://www.geeksforgeeks.org/python-data-types/#numeric)
* [Sequence Type](https://www.geeksforgeeks.org/python-data-types/#Sequence)
* [Boolean](https://www.geeksforgeeks.org/python-data-types/#boolean)
* [Set](https://www.geeksforgeeks.org/python-data-types/#set)
* [Dictionary](https://www.geeksforgeeks.org/python-data-types/#dictionary)



# Numeric:-

In Python, numeric data type represent the data which has numeric value. Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.

**Integers:**– This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.

**Float:**– This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.

**Complex Numbers:** – Complex number is represented by complex class. It is specified as *(real part) + (imaginary part)j*. For example – 2+3j

**Note:** – type() function is used to determine the type of data type.

# Sequence Type:-

# String:- In Python, Strings are arrays of bytes representing Unicode characters. A string is a collection of one or more characters put in a single quote, double-quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by str class.

## **List*:-***Lists are just like the arrays, declared in other languages. Lists need not be homogeneous always which makes it the most powerful tool in Python. A single list may contain Data Types like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation. List in Python are ordered and have a definite count. The elements in a list are indexed according to a definite sequence and the indexing of a list is done with 0 being the first index. Each element in the list has its definite place in the list, which allows duplicating of elements in the list, with each element having its own distinct place and credibility. It is represented by list class.

## **Tuple:-** Tuple is an ordered collection of Python objects much like a list. The sequence of values stored in a tuple can be of any type, and they are indexed by integers. The important difference between a list and a tuple is that tuples are immutable. Also, Tuples are hash able whereas lists are not. It is represented by tuple class.

* BOOLEAN:-

*Type with one of the two built-in values, True or False. Boolean objects that are equal Data to True are truthy (true), and those equal to False. But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class bool.*

**Note:**– True and False with capital ‘T’ and ‘F’ are valid Booleans otherwise python will throw an error.

# Set:-

# In Python, Set is an unordered collection of data type that is interact mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements. The major advantage of using a set, as opposed to a list, is that it has a highly optimized method for checking whether a specific element is contained in the set.

# Dictionary:-

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key: value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a ‘comma’.

**2) Briefly explain history of python?**

[Python](https://www.geeksforgeeks.org/python-programming-language/) 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.

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 application based work in December of 1989 by at Centrum Wiskunde & Informatica (CWI) which is situated in Netherland. It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas. The programming language which Python is said to have succeeded is ABC Programming Language, which had the interfacing with the Amoeba Operating System 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 most of the features. After that what he did as really very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and had created a good scripting language which had removed all the flaws. The inspiration for the name came from BBC’s TV Show – ‘Monty Python’s Flying Circus’, as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python! He was the “Benevolent dictator for life” (BDFL) until he stepped down from the position as the leader on 12th July 2018. For quite some time he used to work

for Google, but currently, he is working at Dropbox.

The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & 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.

3) Explain all the operators in python?

Python Operators:-

* **Arithmetic operators:-** Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication and division.

|  |  |  |
| --- | --- | --- |
| **OPERATOR** | **DESCRIPTION** | **SYNTAX** |
| + | Addition: adds two operands | x + y |
| - | Subtraction: subtracts two operands | x - y |
| \* | Multiplication: multiplies two operands | x \* y |
| / | Division (float): divides the first operand by the second | x / y |
| // | Division (floor): divides the first operand by the second | x // y |
| % | Modulus: returns the remainder when first operand is divided by the second | x % y |
| \*\* | Power : Returns first raised to power second | x \*\* y |

* **Relational Operators:**- Relational operators compares the values. It either returns **True** or **False** according to the condition.

|  |  |  |
| --- | --- | --- |
| **OPERATOR** | **DESCRIPTION** | **SYNTAX** |
| > | Greater than: True if left operand is greater than the right | x > y |
| < | Less than: True if left operand is less than the right | x < y |
| == | Equal to: True if both operands are equal | x == y |
| != | Not equal to - True if operands are not equal | x != y |
| >= | Greater than or equal to: True if left operand is greater than or equal to the right | x >= y |
| <= | Less than or equal to: True if left operand is less than or equal to the right | x <= y |

* **Logical operators:**- Logical operators perform **Logical AND**, **Logical OR** and**Logical NOT** operations.

|  |  |  |
| --- | --- | --- |
| **OPERATOR** | **DESCRIPTION** | **SYNTAX** |
| And | Logical AND: True if both the operands are true | x and y |
| Or | Logical OR: True if either of the operands is true | x or y |
| Not | Logical NOT: True if operand is false | not x |

* **Bitwise operators:** Bitwise operators acts on bits and performs bit by bit operation.

|  |  |  |
| --- | --- | --- |
| **OPERATOR** | **DESCRIPTION** | **SYNTAX** |
| & | Bitwise AND | x & y |
| | | Bitwise OR | x | y |
| ~ | Bitwise NOT | ~x |
| ^ | Bitwise XOR | x ^ y |
| >> | Bitwise right shift | x>> |
| << | Bitwise left shift | x<< |

* **Assignment operators:**- Assignment operators are used to assign values to the variables.

|  |  |  |
| --- | --- | --- |
| **OPERATOR** | **DESCRIPTION** | **SYNTAX** |
| = | Assign value of right side of expression to left side operand | x = y + z |
| += | Add AND: Add right side operand with left side operand and then assign to left operand | a+=b     a=a+b |
| -= | Subtract AND: Subtract right operand from left operand and then assign to left operand | a-=b       a=a-b |
| \*= | Multiply AND: Multiply right operand with left operand and then assign to left operand | a\*=b       a=a\*b |
| /= | Divide AND: Divide left operand with right operand and then assign to left operand | a/=b         a=a/b |
| %= | Modulus AND: Takes modulus using left and right operands and assign result to left operand | a%=b   a=a%b |
| //= | Divide(floor) AND: Divide left operand with right operand and then assign the value(floor) to left operand | a//=b       a=a//b |
| \*\*= | Exponent AND: Calculate exponent(raise power) value using operands and assign value to left operand | a\*\*=b     a=a\*\*b |
| &= | Performs Bitwise AND on operands and assign value to left operand | a&=b     a=a&b |
| |= | Performs Bitwise OR on operands and assign value to left operand | a|=b         a=a|b |
| ^= | Performs Bitwise xOR on operands and assign value to left operand | a^=b       a=a^b |
| >>= | Performs Bitwise right shift on operands and assign value to left operand | a>>=b     a=a>>b |
| <<= | Performs Bitwise left shift on operands and assign value to left operand | a <<= b                    a= a << b |

* **Special operators:-**There are some special type of operators like-
* **Identity operators:-**  
  **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. Two variables that are equal does not imply that they are identical.

**is** True if the operands are identical

**is not** True if the operands are not identical

* **Membership operators**:-  
  **in** and **not in** are the membership operators; used to test whether a value or variable is in a sequence.

**in** True if value is found in the sequence

**not in** True if value is not found in the sequence

4) Explain the features of python?

### Features in Python:

There are many features in Python, some of which are discussed below –

**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, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

**2.Free and Open Source:**  
Python language is freely available at official website and you can download it from the given download link below click on the **Download Python** keyword.

**3.Object-Oriented Language:-**

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

**4. GUI Programming Support:-**  
Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wx Python 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 do not need to remember the system architecture, nor do we need to manage the memory.

**6.Extensible feature:-**  
Python is a **Extensible** language.we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

**7. Python is Portable language:-**  
Python language is also a portable language.for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

**8. Python is Integrated language:-**  
Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

**9. Interpreted Language:--**  
Python is an Interpreted Language. because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called **bytecode**.

**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. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

**11. Dynamically Typed Language:-**  
Python is dynamically-typed language. That means the type (for example- int, double, long etc) 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?

## **Interpreted Python:-**

* Unlike C/C++ etc, Python is an **interpreted object-oriented programming language**. By interpreted it is meant that each time a program is run the interpreter checks through the code for errors and then interprets the instructions into machine-readable bytecode.
* An interpreter is a translator in computer's language which translates the given code line-by-line in machine readable bytecodes. And if any error is encountered it stops the translation until the error is fixed. Unlike C language, which is a **compiled programming language**. The compiler translates the whole code in one-go rather than line-by-line. This is the reason why in C language, all the errors are listed during compilation only.

## **Interactive Python:-**

* 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 the 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 purpose. It simply returns the >>> prompt or the corresponding output of the statement if appropriate and returns **error** for incorrect statements. In this way if you have any doubts like: whether a syntax is correct, whether the module you are importing exists or anything like that, you can be sure within seconds using Python interactive mode.