1. ***Why do we call Python as a general purpose and high-level programming language?***

Python is a dynamic, high-level, free open source, and interpreted programming language. It is considered to be high level programming language because, when we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory. It also 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 dynamically typed language. For example, x = 10 Here, x can be anything such as String, int, etc.

1. ***Why is Python called a dynamically typed language?***

Python is a dynamically typed language because, it doesn’t know about the type of the variable until the code is run. So, declaration is of no use. What it does is, it stores that value at some memory location and then binds that variable name to that memory container. And makes the contents of the container accessible through that variable name. So, the data type does not matter. As it will get to know the type of the value at run-time.

1. ***List some pros and cons of Python programming language*?**

**Pros:**

1. Simple and easy-to-understand syntax.
2. Object Oriented Programming-driven.
3. Supports imperative and functional programming.
4. Extensive library.
5. Supports multiple platforms (Web and mobile computing).
6. Python is easily extensible with C/C++/Java code.
7. Open Source and large community support.

**Cons:**

* 1. Python is slow.
  2. Weak in mobile computing.
  3. Has limitations with database access.
  4. Despite being open source, there is no commercial support point.
  5. Since Python is dynamic, more errors show at run-time.

1. ***In what all domains can we use Python?***

Python is one of the most readable languages in the world right now. This language is very popular among developers and is widely used by many programmers to create application and programs.

The implementation of this programming language is simple and at the same time, the language has a very clean structure as compared to other languages.

Python can be used in domains like,

1. Machine learning / Artificial intelligence
2. Desktop GUI
3. Data analytics and data visualization
4. Web development
5. Game development
6. Mobile app development
7. Embedded systems
8. ***What are variable and how can we declare them?***

Variables are the names you give to computer memory locations which are used to store values in a computer program. A variable in a python program gives data to the computer for processing. Declaring and re-declaring a variable in Python is as easy as writing names and assigning values to it. There is no binding in Python to declare variables before we use it. We also need not explicitly declare variables with their data type. When we assign a value to Python variables, they are automatically declared.

Example: -

* If we want to declare a variable that should hold the total number of chocolates we have, which is 10, we can do it in the following way –

**no\_of\_chocolates=10**

1. ***How can we take an input from the user in Python?***

Taking input is a way of interact with users, or get data to provide some result. Python provides two built-in methods to read the data from the keyboard. These methods are given below.

* input(prompt)
* raw\_input(prompt)

**input()** - The input function is used in all latest version of the Python. It takes the input from the user and then evaluates the expression. The Python interpreter automatically identifies the whether a user input a string, a number, or a list. Let's understand the following example.

***User Input* -**

1. ***# Python program showing***
2. ***# a use of input()***
3. ***name = input("Enter your name: ")   # String Input***
4. ***age = int(input("Enter your age: "))  # Integer Input***
5. ***marks = float(input("Enter your marks: "))  # Float Input***
6. ***print("The name is:", name)***
7. ***print("The age is:", age)***
8. ***print("The marks is:", marks)***

***Output –***

1. ***Enter your name: Johnson***
2. ***Enter your age: 21***
3. ***Enter your marks: 89***
4. ***The name is: Johnson***
5. ***The age is 21***
6. ***The marks is: 89.0***
7. ***What is the default datatype of the value that has been taken as an input using input() function?***

In Python, we implement the input() function to get user input. The input function translates whatever you give it as input into a string. Even if an integer value is entered, the input() method accepts it as a string.

Syntax: input(prompt)

Parameter:

Prompt: (optional) The string to write to standard output (typically the screen) without a newline.

By default, it returns a string object. **Hence, the input() function by default returns the value as string data type​.**

1. ***What is type casting?***

The conversion of one data type into the other data type is known as type casting in python or type conversion in python. Python supports a wide variety of functions or methods like: int(), float(), str(), ord(), hex(), oct(), tuple(), set(), list(), dict(), etc. for the type casting in python.

There can be two types of Type Casting in Python –

1. Implicit Type Casting - Python converts data type into another data type automatically. In this process, users don’t have to involve in this process.
2. Explicit Type Casting - Python need user involvement to convert the variable data type into certain data type in order to the operation required.
3. ***Can we take more than one input from the user using single input() function? If yes, how? If no, why?***

Yes, we can take multiple input from the user using single input() function.

Python user can take multiple values or inputs in one line by two methods.

* Using split() method

***Syntax :***

***input().split(separator, maxsplit)***

* Using List comprehension

***x = list(map(int, input("Enter multiple values: ").split()))***

***print("List of students: ", x)***

1. ***What are keywords?***

**Keywords in Python are reserved words that cannot be used as a variable name, function name, or any other identifier.**

***The list of keywords is:***

**['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']**

1. ***Can we use keywords as a variable? Support your answer with reason.***

**We cannot use a keyword as a variable name, function name, or any other identifier. They are used to define the syntax and structure of the Python language. All the keywords except True, False and None are in lowercase and they must be written as they are. But, we can use them by changing the case of any letter in the keyword.**

**For example,**

***We can't name our variable as follows, as print is a keyword in Python.***

***But we can name our variable by changing the case of any letter in the keyword print*.**

1. ***What is indentation? What's the use of indentation in Python?***

**Python indentation refers to adding white space before a statement to a particular block of code. In another word, all the statements with the same space to the right, belong to the same code. Python indentation is a way of telling a Python interpreter that the group of statements belongs to a particular block of code. A block is a combination of all these statements. Block can be regarded as the grouping of statements for a specific purpose.**

**Example -**

* **Statement (line 1), if condition (line 2), and statement (last line) belongs to the same block which means that after statement 1, if condition will be executed. and suppose the if condition becomes False then the Python will jump to the last statement for execution.**
* **The nested if-else belongs to block 2 which means that if nested if becomes False, then Python will execute the statements inside the else condition.**
* **Statements inside nested if-else belong to block 3 and only one statement will be executed depending on the if-else condition.**

***# Python program showing***

***# Indentation***

***site = 'ineuron'***

***if site == 'ineuron':***

***print('Logging on to ineuron.ai')***

***else:***

***print('retype the URL.')***

***print('All set !')***

***Output:***

***Logging on to ineuron.ai***

***All set !***

1. ***How can we throw some output in Python?***

Python provides the print() function to display output to the standard output devices. Returns: It returns output to the screen. The message can be a string, or any other object, the object will be converted into a string before written to the screen.

***Example: 1***

***User Input:***

***print("Hello World")***

***Output:***

***Hello World***

***Example: 2 (Print a tuple)***

***User Input:***

***x = ("apple", "banana", "cherry")***

***print(x)***

***Output:***

***('apple', 'banana', 'cherry')***

***Example: 3 (Print two messages, and specify the separator:)***

***User Input:***

***print("Hello", "how are you?", sep="---")***

***Output:***

***Hello---how are you?***

1. ***What are operators in Python?***

The operator can be defined as a symbol which is responsible for a particular operation between two operands. Operators are the pillars of a program on which the logic is built in a specific programming language. Python provides a variety of operators, which are described as follows.

* Arithmetic operators
* Comparison operators
* Assignment Operators
* Logical Operators
* Bitwise Operators
* Membership Operators
* Identity Operators

1. ***What is difference between / and // operators?***

/ Operator:

Divides left hand operand by right hand operand.

**// Operator:**

**The division of operands where the result is the quotient in which the digits after the decimal point are removed.**

**Example: */ operator***

***a=10/3***

***print("Value of a is ",a)***

***Output is:***

***Value of a is 3.3333333333333335***

***Example: // operator***

***b=10//3***

***print("Value of b is ",b)***

***Output is:***

***Value of b is 3***

1. ***Write a code that gives following as an output.***

***# To repeat strings n times.***

***string\_var = 'iNeuron'***

***print(string\_var \* 4)***

***Output –***

***iNeuroniNeuroniNeuroniNeuron***

1. ***Write a code to take a number as an input from the user and check if the number is odd or even.***

*# A number is even if division by 2 gives a remainder of 0.*

*# If the remainder is 1, it is an odd number.*

***num = int(input("Enter a number: "))***

***if (num % 2) == 0:***

***print("{0} is Even".format(num))***

***else:***

***print("{0} is Odd".format(num))***

1. ***What are boolean operator?***

Boolean Operators are simple words (AND, OR, NOT or AND NOT) used as conjunctions to combine or exclude keywords in a search, resulting in more focused and productive results. This should save time and effort by eliminating inappropriate hits that must be scanned before discarding.

1. ***What will the output of the following?***
   * 1 or 0 - **‘0’**
   * 0 and 0 - **`1`**
   * True and False and True – **‘True’**
   * 1 or 0 or 0 - **‘0’**
2. ***What are conditional statements in Python?***

Conditional Statement in Python perform different computations or actions depending on whether a specific Boolean constraint evaluates to true or false. Conditional statements are handled by IF statements in Python.

* if
* if else
* Nested if
* if-elif statements.

1. ***What is use of 'if', 'elif' and 'else' keywords?***

The if/elif/else structure is a common way to control the flow of a program, allowing you to execute specific blocks of code depending on the value of some data.

* If keyword– If the condition following the keyword if evaluates as true, the block of code will execute.
* elif keyword - Multiple conditions can be checked by including one or more elif checks after your initial if statement. Just keep in mind that only one condition will execute.
* else keyword - You can optionally add an else response that will execute if the condition is false.

1. ***Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".***

***Code -***

***age = int(input("Enter Your Age : "))***

***if age >= 18:***

***print(age, " I can vote.")***

***else:***

***print(age, " I can't vote.")***

1. ***Write a code that displays the sum of all the even numbers from the given list.nums = []***

***Input -***

***print("Enter 7 elements for the list: ")***

***for i in range(7):***

***val = int(input())***

***nums.append(val)***

***sum = 0***

***for i in range(7):***

***if nums[i]%2 == 0:***

***sum = sum + nums[i]***

***print("\nSum of Even Numbers is", sum)***

***Output -***

***Enter 7 elements for the list: [ 12, 75, 150, 180, 145, 525, 50]***

***Sum of Even Numbers is 392***

1. ***Write a code to take 3 numbers as an input from the user and display the greatest no. as output.***

***Input -***

***num1 = 10***

***num2 = 14***

***num3 = 12***

***if (num1 >= num2) and (num1 >= num3):***

***largest = num1***

***elif (num2 >= num1) and (num2 >= num3):***

***largest = num2***

***else:***

***largest = num3***

***print("The largest number is", largest)***

***Output –***

***The largest number is 14***

1. ***Write a program to display only those numbers from a list that satisfy the following***

***conditions;***

* ***The number must be divisible by five***
* ***If the number is greater than 150, then skip it and move to the next number***
* ***If the number is greater than 500, then stop the loop***

***numbers = [12, 75, 150, 180, 145, 525, 50 ]***

***Input -***

***numbers = [12, 75, 150, 180, 145, 525, 50]***

***# Iterate each item of a list***

***for item in numbers:***

***if item > 500:***

***break***

***elif item > 150:***

***continue***

***# Check if number is divisible by 5***

***elif item % 5 == 0:***

***print(item)***

***Output –***

***75***

***150***

***145***