

A-1- Python is considered a general-purpose programming language because it can be used for a wide range of applications, including web development, data analysis, scientific computing, artificial intelligence, machine learning, game development, and more. It is not limited to a specific domain or application area, and it provides a lot of flexibility in terms of the tasks it can accomplish.

Python is also a high-level programming language, which means that it abstracts away many of the low-level details of the computer's hardware and operating system.

This makes it easier to write code that is more concise, readable, and easier to maintain compared to low-level programming languages like Assembly or C. Python's syntax is simple and easy to understand, which reduces the time and effort required to write code and makes it more accessible to beginners.

A-2- Python is called a dynamically typed language because the data type of a variable is determined by the value it contains when it is created or assigned a value, and it can be changed later if needed. For example, in Python, you can assign a string value to a variable and then later assign an integer value to the same variable without having to explicitly specify the data type of the variable. This is because Python automatically determines the data type based on the value assigned to the variable.

A-3- Pros:

- \*Easy to learn and use: Python has a simple syntax and a large standard library, making it easy to learn and use for beginners.

- \*Versatile: Python can be used for a wide range of applications, from web development to scientific computing, artificial intelligence, machine learning, and more.

- \*Dynamically typed: Python is a dynamically typed language, which means that the data type of a variable is determined at runtime, making it more flexible and easier to use.

- \*Interpreted language: Python is an interpreted language, which means that the code can be run without the need for a separate compilation step, making it quicker to test and debug code.

- \*Large community and support: Python has a large and active community of developers and users, providing a wealth of resources, libraries, and frameworks to help solve problems and improve productivity.

Cons:

- \*Slower execution speed: Python is an interpreted language, which can result in slower execution speeds compared to compiled languages like C or Java.

- \*GIL limitation: Python's Global Interpreter Lock (GIL) can limit performance when running CPU-bound tasks, as it prevents multiple threads from executing Python code simultaneously.

- \*Weak on mobile and game development: Python is not as well-suited for mobile app development or game development compared to other programming languages like Swift, Java, or C++.

- \*Design restrictions: Python's design philosophy enforces some restrictions on the code style and organization, which can limit the programmer's creativity and flexibility.

- \*Memory management: Python's memory management can sometimes lead to issues with memory leaks and other performance issues if not managed carefully.

A-4-Python is a versatile programming language that can be used in a wide range of domains and applications. Here are some of the main domains in which Python is commonly used:

- \*Web development: Python has a range of web frameworks such as Django, Flask, and Pyramid that enable developers to build high-performance web applications with ease.
- \*Data science and machine learning: Python is a popular language for data science and machine learning tasks, as it provides powerful libraries such as NumPy, Pandas, and Scikit-learn, that simplify data analysis, visualization, and model building.
- \*Artificial intelligence and deep learning: Python has a rich set of libraries for artificial intelligence and deep learning, such as TensorFlow, PyTorch, and Keras, that enable developers to create complex models for image and speech recognition, natural language processing, and more.
- \*Scientific computing: Python is a popular choice for scientific computing tasks, as it provides libraries such as SciPy, Matplotlib, and SymPy that enable scientists and researchers to perform numerical calculations, data analysis, and simulations.
- \*Game development: Python can be used to develop 2D and 3D games with the help of libraries like Pygame, Panda3D, and PyOgre.
- \*Desktop applications: Python can be used to build cross-platform desktop applications using frameworks such as PyQt, wxPython, and PyGTK.
- \*DevOps: Python is widely used for automation, scripting, and configuration management tasks in DevOps, with popular tools such as Ansible, Fabric, and Salt.

A-5- A variable is a named reference to a value that can be used throughout the program. A variable allows you to store data of different types, such as strings, integers, floating-point numbers, and more. To declare a variable in Python, you simply choose a name for the variable and use the equal sign "=" to assign a value to it.

A-6- We can take the input from user by syntax:

```
a=input('ENTER THE INPUT')
print('YOU HAVE TYPE: ',a)
```

A-7- The default data type of the value returned by the input() function in Python is a string.

A-8- type casting is the process of converting a value of one data type to another data type.

A-9- Yes we can take multiple inputs from the user using a single input() function call by asking the user to enter multiple values separated by a delimiter, such as a space or a comma, and then splitting the input string into individual values using the split() function.

A-10- Keywords in Python are reserved words that have special meaning and purpose in the language. These words are used to define the syntax and structure of the language itself and cannot be used as identifiers (i.e., variable names, function names, etc.) in the program.

EX- and, as, assert, break, class, continue, def, del.

A-11- No, we cannot use keywords as a variable name in Python. Keywords are reserved words that have a predefined meaning and usage in the Python language. If we try to use a keyword as a variable name or any other identifier in our program, we will get a syntax error.

A-12- In Python, indentation refers to the whitespace (spaces or tabs) at the beginning of a line of code. Indentation is used to define the block of code, i.e., a group of statements that belong together.

The use of indentation in Python is significant because it defines the structure of the code and the logical flow of the program. Unlike other programming languages, where the braces or brackets are used to group the code, Python uses indentation for this purpose.

A-13-In Python, we can use the `print()` function to display output on the console. The `print()` function is a built-in function that takes one or more arguments, formats them as strings, and displays them on the console.

A-14-Operators in Python are special symbols or keywords that perform operations on one or more operands (variables, constants, expressions). Python supports a wide range of operators that can be used for arithmetic, comparison, logical, and bitwise operations.

A-15-In Python, the `/` operator performs floating-point division, while the `//` operator performs integer division (also known as floor division).

A-16-`print("iNeuroniNeuroniNeuroniNeuron")`

A-17-  
`a=int(input("ENTER YOUR NUMBER: "))`  
`if a%2==0:`  
    `print('the number is even')`  
`else:`  
    `print('the number is odd')`

A-18-Boolean operators are operators that operate on Boolean values (True and False) and return a Boolean result. Python provides three Boolean operators: `and`, `or`, and `not`.

A-19-  
`1 or 0 : 1`  
`0 and 0 : 0`  
`True and False and True : False`  
`1 or 0 or 0 : 1`

A-20-Conditional statements in Python are used to execute different code blocks based on whether a particular condition is True or False. The two main types of conditional statements in Python are the `if` statement and the `if-else` statement.

A-21-  
\*`if` statement: The `if` statement is used to execute a block of code if a particular condition is True.  
\*`if-else` statement: The `if-else` statement is used to execute one of two code blocks depending on whether a particular condition is True or False.  
\*`elif` statement: The `elif` statements to test multiple conditions.

A-22-  
`a=int(input("enter the age of a person: "))`  
`if a>=18:`  
    `print('I can vote')`  
`elif a<18:`  
    `print("I can't vote")`

```
A-23-  
sum=0  
numbers = [12, 75, 150, 180, 145, 525, 50]  
for i in numbers:  
    if i%2==0:  
        sum=sum+i  
  
print(sum)
```

```
A-24-  
a=int(input("ENTER FIRST NUMBER: "))  
b=int(input("ENTER SECOND NUMBER: "))  
c=int(input("ENTER THIRD NUMBER: "))  
if a>b and a>c:  
    print(a,"is greatest")  
elif b>a and b>c:  
    print(b,"is greatest")  
elif c>a and c>b:  
    print(c,"is greatest")
```