**Q1. Why do we call Python a general purpose and high-level programming language?**

A1. Python is a general purpose programming language as it can be used to build software in a wide variety of application domains such as data science, web development, software development, machine learning etc.   
Python is a high-level programming language as it is easily understood by humans but computers can not directly understand it. It has to be converted to machine code or low level language for it to be understood by a computer.

**Q2. Why is Python called a dynamically typed language?**

A2. Python is called a dynamically typed language because in Python an interpreter assigns variables a type at runtime based on the variable's value at the time & the type of a variable is allowed to change over its lifetime.

**Q3. List some pros and cons of the Python programming language?**

A3.

Pros -

* Free and open source
* Beginner friendly - It is easy to learn, understand and code as it does not include too many technicalities.
* Extensive Libraries
* Portable - Write once run anywhere.
* Highly scalable
* Dynamically typed - As the programmer, you don’t have to worry about the declaration of variables and the data types.

Cons -

* Poor Memory Efficiency
* Slow speed - It is slow primarily due to its dynamic nature and versatility.
* Weak in Mobile Computing
* Runtime Errors - A variable holding a string may contain an integer later, and this can lead to runtime errors.

**Q4. In what all domains can we use Python?**

A4. Python can be used in a wide variety of domains. Some of them are -

* Data science, Data analytics, data visualization.
* Web development.
* Game development.
* Machine learning / Artificial intelligence.
* Mobile app development.
* Desktop GUI.

**Q5. What are variables and how can we declare them?**

A5. Variable is a name given to a memory location.

Python has no command for declaring a variable. A variable is created the moment you first assign a value to it.

Eg. a = 5

b = “Tom”

**Q6. How can we take an input from the user in Python?**

A6. Input() function is used to take user input in python.

Syntax - input(prompt)

Eg. name = input("Enter name = ")

print("User name = ",name)

**Q7. What is the default datatype of the value that has been taken as an input using input() function?**

A7. String

**Q8. What is type casting?**

A8. Type casting is a method to convert datatype of a variable.

**Q9. Can we take more than one input from the user using a single input() function? If yes, how? If no, why?**

A9. Yes, split () can be used to take multiple inputs from a user using single input ().

Eg. x, y = input("Enter the number of boys and girls: ").split()

print("Number of boys: ", x)

print("Number of girls: ", y)

**Q10. What are keywords?**

A10. Python keywords are special reserved words that have specific meanings and purposes and can't be used for anything but those specific purposes.

Eg. break, continue, else, elif, for, if, in etc.

**Q11. Can we use keywords as a variable? Support your answer with reason.**

A11. No, keywords can’t be used as a variable in python because keywords have specific meaning and purpose in python.

Eg. while is a keyword in python used to create a loop.

while = 5

print (while)

This code would not work and throw a syntax error because python understands that while has a specific function and specific syntax and therefore it will try to execute it for looping and not as a variable holding a value.

**Q12. What is indentation? What's the use of indentation in Python?**

A12. Indentation is simply the spaces at the beginning of a code line. It is used for readability, but in Python, the indentation is an essential and mandatory.

Python uses indentation to indicate a block of code. Python treats the statements with the same indentation level (statements with an equal number of whitespaces before them) as a single code block.

Eg.

if x==y:

print("Yes, X is Equals to Y !!")

else:

print("No, X is not Equals to Y !!")

The print statement under the if statement is if code block similarly the print statement under else is the else code block.

**Q13. How can we throw some output in Python?**

A13. print () can be used to throw output in Python.

Eg.

name = "Tom"

print(name)

Output - Tom

**Q14. What are operators in Python?**

A14. Operators are used to perform operations on variables and values.

There are multiple operators in python -

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

Eg.

x = 5

y = 3

print("Addition of x + y = ", x+y)

Output - Addition of x + y = 8

**Q15. What is the difference between / and // operators?**

A15. / is for float division &

// is for integer division.

Eg.

x = 5

y = 3

print("Float Division of x / y = ", x/y)

print("Integer Division of x // y = ", x//y)

Output - Float Division of x / y = 1.6666666666666667

Integer Division of x // y = 1

**Q16. Write a code that gives the following as an output.**

**```**

**iNeuroniNeuroniNeuroniNeuron**

A16.

str1 = "iNeuron"\*4

print("Multiply str = ", str1)

**Q17. Write a code to take a number as an input from the user and check if the number is odd or even.**

A17.

num = int(input("Enter any number ="))

if num % 2 == 0 :

print("The provided number is even")

else :

print("The provided number is odd")

**Q18. What are boolean operators?**

A18. Boolean Operators are those that result in the Boolean values of True and False.

There are two types of operators in Python that return boolean values, i.e., Logical operators and Comparison operators. Since both these operators have return types as boolean, they are also termed Boolean operators.

**Q19. What will be the output of the following?**

**```**

**1 or 0**

**0 and 0**

**True and False and True**

**1 or 0 or 0**

A19.

| 1 or 0 | 1 |
| --- | --- |
| 0 and 0 | 0 |
| True and False and True | False |
| 1 or 0 or 0 | 1 |

**Q20. What are conditional statements in Python?**

A20. These statements guide the program while making decisions based on the conditions encountered by the program.

Using conditional statements, the program can be routed to different steps based on check of conditional statements.

Python has 3 key Conditional Statements:

If statement

If-else statement

Nested if-else statement

**Q21. What is the use of 'if', 'elif' and 'else' keywords?**

A21. If, else and elif keywords are used as conditional statements to determine whether a block of code will be executed or not.

Else - Is used to execute else code block when the “if” conditions fail.

Elif - Is used in nested if else statement i.e. to make multiple conditional statements.

**Q22. 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".**

A22.

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

if Age >=18:

print("I can vote")

else:

print("I can't vote")

**Q23. Write a code that displays the sum of all the even numbers from the given list.**

**```**

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

**```**

A23.

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

even = []

for num in numbers:

if num % 2 == 0:

even.append(num)

print(even)

total = 0

for el in range(0, len(even)):

total = total + even[el]

print(total)

**Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.**

A24.

x, y, z = [int(x) for x in input("Enter 3 numbers: ").split()]

if x>y and x>z:

print(x, "is the largest among the entered numbers")

elif y > z and y > x:

print(y, "is the largest among the entered numbers")

else:

print(z, "is the largest among the entered numbers")

**Q25. 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]**

A25.

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

for x in numbers:

if x > 500:

break

elif x > 150:

continue

elif x % 5 == 0:

print(x)