1. What is Python?

Python is a high-level, interpreted, general-purpose programming language. It's known for its emphasis on code readability, elegant syntax, and versatility.

2. Why is Python so popular?

Python is popular due to:

- Readability: Its syntax resembles English, making it easy to learn.
- Versatility: Used in web development, data science, machine learning, automation, and more.
- Strong community: Vast support, resources, and libraries.
- Beginner-friendly: A great language to start coding with.

3. What is the difference between interpreted and compiled languages?

- Interpreted languages (like Python) are translated into machine code line-by-line during execution. This allows for more flexibility and cross-platform compatibility.
- Compiled languages (like C++) are translated into machine code before execution, potentially leading to faster performance but less flexibility.

4. How do you run Python code?

Several ways to run Python code:

- Interactive shell: Type code directly into the Python interpreter.

- IDE: Use Integrated Development Environments (IDEs) like PyCharm or VS Code.
- Text editor and command line: Write code in a text editor and run it from the command line using the python command.

5. What is a variable in Python?

A variable is a named container that stores a value. You can use it to reference and manipulate data.

Read more...

Data Types

6. Name some basic data types in Python.

Core data types include:

- Numbers: Integers (e.g., 10), floating-point numbers (e.g., 3.14).
- Strings: Sequences of characters enclosed in quotes (e.g., “Hello, world!”).
- Booleans: Represent True or False.
- Lists: Ordered collections of items (e.g., [1, “apple”, 3.5]).
- Tuples: Immutable (unchangeable) ordered collections (e.g., (2024, “March”)).
- Dictionaries: Unordered sets of key-value pairs (e.g., {“name”: “Khushi”, “age”: 12}).

Operators

7. What are arithmetic operators in Python?

Arithmetic operators perform mathematical calculations:

- + Addition
- – Subtraction
- * Multiplication
- / Division
- % Modulo (remainder)
- // Floor division (integer result)
- ** Exponentiation

8. What are comparison operators in Python?

Comparison operators compare values, returning True or False:

- == Equal to
- != Not equal to
- < Less than
- > Greater than
- <= Less than or equal to
- >= Greater than or equal to

Control Flow

9. Explain the ‘if’ statement in Python.

The if statement allows conditional code execution. If a specified condition is True, a block of code is run:

```
if age >= 18:
    print("You are eligible to vote.")
```

10. What are ‘for’ loops used for in Python?

for loops iterate over sequences like lists, strings, etc. For each item, a block of code is executed:

```
fruits = ["apple", "banana", "orange"]
for fruit in fruits:
    print(fruit)
```

11. How does a ‘while’ loop work in Python?

while loops execute code repeatedly as long as a condition is True:

```
count = 0
while count < 5:
    print(count)
    count = count+1
```

12. What is a function in Python?

A function is a reusable block of code that performs a specific task. It helps organize code and promotes modularity. You define functions using the def keyword.

```
def greet(name):
    print("Hello,", name)

greet("Khushi")
```

13. Explain the difference between parameters and arguments.

- Parameters: Variables defined within the function's parentheses when it's being defined. They act as placeholders for values.
- Arguments: The actual values passed to the function when it's called.

14. What is the return statement used for?

The return statement sends a value back from a function to the caller. The function stops executing after the return.

```
def square(num):
  return num * num
```

Object-Oriented Programming (OOP)

15. What is a class in Python?

A class is a blueprint for creating objects. It defines attributes (data) and methods (behaviors) common to a type of object.

```
class Dog:
    def __init__(self, name, breed):
        self.name = name
        self.breed = breed

    def bark(self):
        print("Woof!")
```

16. What is an object?

An object is an instance of a class. It has state (data stored in attributes) and behavior (methods defined in its class).

17. Briefly explain the concept of inheritance in OOP.

Inheritance allows you to create new classes (child classes) that inherit properties and behaviors from existing classes (parent classes). This promotes code reuse and a hierarchical structure.

Libraries/Modules

18. What is a module in Python?

A module is a Python file (.py) containing reusable code like functions, classes, etc. You

import them using the import statement.

19. Name a few commonly used Python libraries.

- NumPy: For scientific computing and working with arrays.
- Pandas: For data analysis and manipulation.
- Matplotlib: For creating visualizations (plots, graphs).
- Requests: For making HTTP requests (web interaction).

20. How do you import a specific function from a module?

Use the from … import … syntax:

```
from math import sqrt  # Imports only the sqrt function
```

Related posts:

1. Does Python have double, short long data types
2. What are Interpreted Programming Languages?
3. What are General-Purpose Programming Languages?
4. What is a variable in Python?