Q: What is Python, and why is it popular?

A: Python is a high-level, interpreted programming language known for its readability and simplicity.

Popular because:

- Easy to learn and write
- Vast libraries (e.g., NumPy, Pandas, TensorFlow)
- Used in web development, data science, automation, AI, etc.
- Strong community support

Q: What is an interpreter in Python?

A: An interpreter reads and executes Python code line by line, converting it into machine code directly without compiling the whole program at once.

Q: What are pre-defined keywords in Python?

A: Keywords are reserved words that have special meaning in Python, like: if, else, for, while, def, class, return, True, False, etc. You cannot use them as variable names.

Q: Can keywords be used as variable names?

A: No, keywords cannot be used as variable names because they have predefined meanings and using them would confuse the interpreter.

Q: What is mutability in Python?

A: Mutability refers to whether an objects value can be changed after it is created.

- Mutable: Can be changed (e.g., lists, dictionaries)
- Immutable: Cannot be changed (e.g., strings, tuples)

Q: Why are lists mutable, but tuples are immutable?

A: Lists are designed for dynamic data handling, so they allow changes (adding, removing, updating items).

Tuples are meant for fixed data structures where safety and performance matter, so they are immutable.

Q: What is the difference between == and is operators in Python?

A: == checks if values of two variables are equal.

is checks if two variables point to the same object in memory.

Example:

a = [1, 2]

b = [1, 2]

a == b # True (values are same)

a is b # False (different memory locations)

Q: What are logical operators in Python?

A: Logical operators combine conditional statements:

- and: True if both conditions are true

- or: True if at least one condition is true

- not: Reverses the result

Q: What is type casting in Python?

A: Type casting is converting one data type to another.

Example: x = int("5") # Casting string to integer

Q: What is the difference between implicit and explicit type casting?

A: Implicit: Done automatically by Python.

Example: x = 10 + 5.5 # int + float = float

Explicit: Manually done by the programmer.

Example: x = int(5.9) # float to int = 5

Q: What is the purpose of conditional statements in Python?

A: Conditional statements control the flow of a program by executing different code depending on whether a condition is true or false.

Q: How does the elif statement work?

A: elif stands for 'else if'. It lets you check multiple conditions after an initial if condition.

Example:

```
x = 10
if x == 5:
    print("Five")
elif x == 10:
    print("Ten")
else:
    print("Other")
```

Q: What is the difference between for and while loops?

A: for loop: Used when the number of iterations is known.

while loop: Used when the loop should continue until a condition is false.

Q: Describe a scenario where a while loop is more suitable than a for loop.

A: When you dont know beforehand how many times the loop will run.

Example:

```
word = ""
while word != "stop":
    word = input("Type 'stop' to quit: ")
```