Q1. What is Python? Mention two of its key features.

Python is a high-level, interpreted, object-oriented programming language.
Two key features:
1. Easy and Readable Syntax – Python uses simple English-like commands and indentation, making it beginner-friendly.
2. Cross-Platform & Portable – Python code can run on different operating systems (Windows, Mac, Linux) without modification.

Q2. How do you write a single-line comment in Python?

Single-line comments begin with the # symbol.
Example:

```
# This is a single-line comment
print("Hello Python")
```

Q3. How do you write a multi-line comment in Python?

Python doesn't have a specific multi-line comment syntax, but we use triple quotes.
Example:

```
"""
This is a multi-line comment
spanning across several lines
"""
print("Hello World")
```

Q4. What is the difference between print() and return?

- print(): Displays output to the console.
- return: Sends a value back from a function to the caller.

Example:

```
def add(a, b):
    return a + b

result = add(5, 3)
print(result)
```

Q5. How do you get user input in Python?

Using the input() function.
Example:

```
name = input("Enter your name: ")
print("Hello", name)
```

Q6. How do you check the version of Python installed?

From command line:

```
python --version
```

Inside Python:

```
import sys
print(sys.version)
```

Q7. Is Python case-sensitive? Give an example.

Yes, Python is case-sensitive.

```
a = 10
A = 20
print(a) # 10
print(A) # 20
```

Q8. How do you run a Python script from the command line?

Save your file as program.py.
Run in terminal:

```
python program.py
```

Q9. What are keywords in Python? How can you list them?

Keywords are reserved words in Python with special meaning.
To list:

```
import keyword
print(keyword.kwlist)
```

Q10. How do you declare a variable in Python?

Just assign a value (no type needed).

```
x = 10
name = "Vinniii"
```

Q11. What's the difference between = and ==?

- = : Assignment operator.
- == : Comparison operator.

Example:

```
x = 5
print(x == 5) # True
```

Q12. How do you swap two variables without using a third variable?

Using tuple unpacking:

```
a, b = 10, 20
a, b = b, a
print(a, b)
```

Q13. How do you write a one-line if statement?

```
x = 10
if x > 5: print("x is greater than 5")
```

Conditional expression:

```
print("Even") if x % 2 == 0 else print("Odd")
```

Q14. What's the difference between None and 0 in Python?

- None: Represents no value/null.
- 0: Integer value.

Example:

```
a = None
b = 0
print(a == b) # False
```

Q15. What is indentation in Python and why is it important?

Indentation defines code blocks in Python.
Example:

```
if True:
    print("Indented correctly")
```

Q16. What are Python's built-in data types?

Numeric: int, float, complex
Sequence: list, tuple, range
Text: str
Set: set, frozenset
Mapping: dict
Boolean: bool
Binary: bytes, bytearray, memoryview

Q17. How do you check the type of a variable?

Using type() function.
Example:

```
x = 10
print(type(x)) # <class 'int'>
```

Q18. What's the difference between a list and a tuple?

List: Mutable, defined by []
Tuple: Immutable, defined by ()

Q19. How do you create a dictionary in Python?

```
student = {"name": "Sneha", "age": 22, "course": "MCA"}
print(student["name"])
```

Q20. What's the difference between append() and extend() for lists?

- append(): Adds a single element.
- extend(): Adds multiple elements.

Example:
```
a = [1,2]
a.append([3,4]) # [1,2,[3,4]]
a.extend([5,6]) # [1,2,[3,4],5,6]
```

Q21. How do you remove an item from a list?

Methods:
- remove(value): removes first occurrence.
- pop(index): removes at index (default last).
- del list[index].

Example:
```
a = [1,2,3]
a.remove(2) # [1,3]
```

Q22. How do you reverse a list in Python?

Using reverse() or slicing.
Example:
```
a = [1,2,3]
a.reverse() # [3,2,1]
```

Or:
```
print(a[::-1])
```

Q23. How do you sort a list in ascending order?

Using sort() or sorted().
Example:
```
numbers = [3,1,2]
numbers.sort()
print(numbers) # [1,2,3]
```

Q24. What is the difference between shallow copy and deep copy?

- Shallow copy: Copies references (changes in nested objects reflect).
- Deep copy: Fully copies all objects.

Example:
```
import copy
shallow = copy.copy(list1)
deep = copy.deepcopy(list1)
```

Q25. How do you convert a string to lowercase?

Using lower().
Example:
```
s = "HELLO"
print(s.lower())
```

Q26. How do you check if a string starts with a particular word?

Using startswith().
Example:
```
s = "Python is fun"
print(s.startswith("Python")) # True
```

Q27. What's the difference between is and ==?

- == : Compares values.
- is : Compares memory location (identity).

Example:
```
a = [1,2]
b = [1,2]
print(a == b) # True
print(a is b) # False
```

Q28. How do you merge two dictionaries in Python 3.9+?

Using | operator.
```
d1 = {"a":1}
d2 = {"b":2}
merged = d1 | d2
print(merged)
```

Q29. How do you find the length of a dictionary?

Using len().
```
d = {"a":1, "b":2}
print(len(d)) # 2
```

Q30. How do you create a set?

Using {} or set().
```
s1 = {1,2,3}
s2 = set([4,5])
```

Q31. What's the difference between set() and {} in Python?

- {} : Creates an empty dictionary.
- set() : Creates an empty set.

Example:

a = {} # dict
b = set() # set

Q32. How do you find the union of two sets?

Using union() or |.
A = {1,2}
B = {2,3}
print(A | B) # {1,2,3}

Q33. How do you find the intersection of two sets?

Using intersection() or &.
A = {1,2}
B = {2,3}
print(A & B) # {2}

Q34. What's the difference between remove() and discard() in sets?

- remove(): Removes element, raises error if not found.
- discard(): Removes element, no error if missing.

Q35. How do you convert a list into a tuple?

Using tuple().
l = [1,2,3]
t = tuple(l)

Q36. How does the if-elif-else structure work in Python?

It checks conditions in order:

```
if condition1:
  code
elif condition2:
  code
else:
  code
```

Q37. What is the difference between for and while loops?

- for: Iterates over sequence/range.
- while: Runs until condition false.

Example:
for i in range(3): print(i)
while i < 3: print(i); i+=1

Q38. How do you loop through a dictionary's keys and values?

Using items().
d = {"a":1, "b":2}

```
for k,v in d.items():
    print(k,v)
```

Q39. How do you break out of a loop?

Using break keyword.

```
for i in range(5):
    if i==3: break
```

Q40. How do you skip the current iteration in a loop?

Using continue.

```
for i in range(5):
    if i==2: continue
    print(i)
```