1. What is dynamic typing in Python?

Dynamic typing means you do not need to declare the data type of a variable.
The type is decided at runtime.

Example:

```
x = 10       # x is int
x = "Hello" # x is now string
```

2. Difference between mutable and immutable data types

Mutable: Can be changed after creation
Immutable: Cannot be changed after creation

Example:

```
# Mutable
lst = [1, 2, 3]
lst[0] = 10
print(lst)   # [10, 2, 3]

# Immutable
tup = (1, 2, 3)
# tup[0] = 10  ❌ Error
```

3. What are local and global variables?

Local variable: Defined inside a function.
Global variable: Defined outside a function.

Example:

```
x = 10  # global

def func():
    y = 5   # local
    print(y)

func()
print(x)
```

4. What is the purpose of the return statement?

The return statement sends a value back from a function.

Example:

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

result = add(3, 4)
print(result)   # 7
```

5. What is list comprehension?

List comprehension is a short and simple way to create a list.

Example:

```
squares = [x*x for x in range(5)]
print(squares)   # [0, 1, 4, 9, 16]
```

6. Difference between break, continue, and pass

break: Stops the loop
continue: Skips current iteration
pass: Does nothing

Example:

```
for i in range(5):
    if i == 2:
        continue
    if i == 4:
        break
    pass
    print(i)
```

7. What is recursion? Give a simple example.
Recursion is when a function calls itself.

Example (factorial):

```
def fact(n):
    if n == 1:
        return 1
    return n * fact(n-1)

print(fact(4))   # 24
```

8. What is a lambda function?

A lambda function is a small anonymous function written in one line.

Example:

```
add = lambda a, b: a + b
print(add(2, 3))   # 5
```

9. Difference between is and ==

== → checks value equality
is → checks memory location

Example:

```
a = [1, 2]
b = [1, 2]

print(a == b)  # True
print(a is b)  # False
```

10. What are docstrings and why are they important?

Docstrings are used to document functions, classes, or modules.

Example:

```
def add(a, b):
    """This function adds two numbers"""
    return a + b

print(add.__doc__)
```

Importance:

Improves readability
Helps in documentation
Useful for developers