Python

1. What is dynamic typing in Python?
Dynamic typing means the data type of a variable is decided at runtime, not in advance.
A variable can change its type during execution.
Eg:
```
 x = 10
x = "Hello”
```

2. What is the difference between mutable and immutable data types?
Mutable data types can be changed after creation.
Immutable data types cannot be changed
Eg:
```
# Mutable
lst = [1, 2, 3]
lst[0] = 10

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

3. What are local and global variables?
Local variables are defined inside a function and used only there.
Global variables are defined outside a function and can be accessed anywhere.
Eg:
```
x = 5  # global

def fun():
    y = 10  # local
    print(y)
```

4. What is the purpose of the return statement in a function?
The return statement sends a value back from a function and ends its execution**.** Eg:
```
def add(a, b):
    return a + b

print(add(2, 3))
```

5. What is list comprehension?
List comprehension is a short way to create a list using a single line of code**.**

Eg:
```
squares = [x*x for x in range(5)]
```

6. Difference between break, continue, and pass
break stops the loop completely.
continue skips the current iteration.
pass does nothing and acts as a placeholder.

Eg:
```
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 a function calling itself to solve a problem.
```
def factorial(n):
    if n == 1:
        return 1
    return n * factorial(n-1)

print(factorial(5))
```

8. What is a lambda function?
A lambda function is a small anonymous function written in one line.
Eg:
```
square = lambda x: x * x
print(square(4))
```

9. What is the difference between is and ==?
== compares values.
is checks whether both variables refer to the same object in memory.
Eg:
```
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 strings used to describe functions, classes, or modules.
They help in documentation and understanding the code.

Eg:

```
def add(a, b):
    """This function returns the sum of two numbers"""
    return a + b
```