**Python Functions**

**Q1. How do you define a function in Python?**  
Using the def keyword.

def greet(name):

return "Hello, " + name

**Q2. Difference between \*args and \*\*kwargs?**  
\*args is used to pass variable-length positional arguments, while \*\*kwargs passes variable-length keyword arguments.

**Q3. What are default arguments in Python functions?**  
 They allow parameters to have default values if no argument is passed.

**Q4. Can Python functions return multiple values?**  
 Yes, as a tuple.

def calc(a, b):

return a+b, a-b

**Q5. Explain lambda functions.**  
 Small, anonymous functions defined with lambda. Example:

square = lambda x: x\*x

**Q6. Can functions be assigned to variables?**  
Yes, functions are first-class objects in Python.

**Q7. What are keyword arguments?**  
 Arguments passed by name, e.g., greet(name="Nitin").

**Q8. What is recursion?**  
Function calling itself to solve smaller subproblems.

**Q9. What are anonymous functions?**  
 Functions created with lambda.

**Q10. Difference between return and yield?**  
 return exits function, yield makes it a generator.

**Q11. What is scope in Python?**  
 Local, Enclosing, Global, Built-in (LEGB rule).

**Q12. What are decorators?**  
 Functions that modify behavior of other functions/classes.

**Q13. Can functions be nested in Python?**  
Yes, functions can be defined inside other functions.

**Q14. What is closure in python?**

A closure is a function that:

1. Is defined inside another function (an inner function),

2. Captures variables from the enclosing (outer) function’s scope, even after that outer function has finished executing.

**Q15. Explain LEGB in Python.**

“LEGB is Python’s rule for resolving variable names. When Python sees a variable, it looks for it in a specific order: **Local → Enclosing → Global → Built-in**.”

1. **Local (L)**
   * Names defined inside the current function.
   * Example: variables assigned inside a function.
2. **Enclosing (E)**
   * Names in the scope of any **outer function** (for nested functions).
   * Example: inner function can access variables of its outer function.
3. **Global (G)**
   * Names defined at the top-level of the module/file.
   * Example: variables declared outside all functions.
4. **Built-in (B)**
   * Names that Python provides by default, like len, print, range.