Functions

## 1. What is the difference between a function and a method in Python?

A function is a block of code that performs a specific task and is defined using the `def` keyword. A method is similar but is associated with an object and called on that object.  
  
Example:  
Function: def greet(): print('Hello')  
Method: 'hello'.upper() # upper is a method

## 2. Explain the concept of function arguments and parameters in Python.

Parameters are variables listed in the function definition. Arguments are the values passed to the function when called.  
  
Example:  
def add(a, b): return a + b  
add(2, 3) # 2 and 3 are arguments

## 3. What are the different ways to define and call a function in Python?

Functions are defined using `def` or `lambda`, and called by using parentheses.  
  
Example:  
def greet(): print('Hi')  
greet()  
  
lambda: square = lambda x: x\*x

## 4. What is the purpose of the return statement in a Python function?

It sends back the result of the function to the caller.  
  
Example:  
def square(x): return x\*x

## 5. What are iterators in Python and how do they differ from iterables?

An iterator is an object with \_\_next\_\_() method. Iterable is an object which can be looped over.  
  
Example:  
iter\_obj = iter([1, 2, 3]) # Iterator  
[1, 2, 3] is iterable

## 6. Explain the concept of generators in Python and how they are defined.

Generators are functions that yield values one by one using the `yield` keyword.  
  
Example:  
def gen():  
 yield 1  
 yield 2

## 7. What are the advantages of using generators over regular functions?

Generators are memory-efficient and lazy evaluated.  
  
Example:  
def gen():  
 for i in range(10): yield i

## 8. What is a lambda function in Python and when is it typically used?

A lambda function is an anonymous function used for short operations.  
  
Example:  
square = lambda x: x\*x

## 9. Explain the purpose and usage of the map() function in Python.

map() applies a function to every element in an iterable.  
  
Example:  
list(map(lambda x: x\*2, [1, 2, 3]))

## 10. What is the difference between map(), reduce(), and filter() functions in Python?

- map(): applies a function to all items  
- filter(): filters items by condition  
- reduce(): reduces to a single value  
  
Example:  
from functools import reduce  
reduce(lambda x, y: x + y, [1, 2, 3])

## 11. Write the internal mechanism for sum operation using reduce list: [47,11,42,13]

Step 1: 47 + 11 = 58  
Step 2: 58 + 42 = 100  
Step 3: 100 + 13 = 113  
Final result = 113