**Exercise 1: Create a function in Python**

Write a program to create a function that takes two arguments, name and age, and print their value.

*# demo is the function name*def demo(name, age):  
 *# print value* print(name, age)  
  
*# call function*demo("Lakshmi", 20)

**OUTPUT:**

**Lakshmi 20**

**Exercise 2: Create a function with variable length of arguments**

Write a program to create function func1() to accept a variable length of arguments and print their value.

**Note**: Create a function in such a way that we can pass any number of arguments to this function, and the function should process them and display each argument’s value.

**Function call**

# call function with 3 arguments

func1(20, 40, 60)

# call function with 2 arguments

func1(80, 100)

**Expected Output**:

Printing values

20

40

60

Printing values

80

100

def func1(\*args):  
 for i in args:  
 print(i)  
  
func1(20, 40, 60)  
func1(80, 100)

### Exercise 3: Return multiple values from a function

Write a program to create function calculation() such that it can accept two variables and calculate addition and subtraction. Also, it must **return both addition and subtraction in a single return call**.

**Given**:

**def** calculation(a, b):

# Your Code

res = calculation(40, 10)

**print**(res)

**Expected Output**

50, 30

**Solution 1**:

def calculation(a, b):  
 addition = a + b  
 subtraction = a - b  
 *# return multiple values separated by comma* return addition, subtraction  
  
*# get result in tuple format*res = calculation(40, 10)  
print(res)

**Solution 2:**

def calculation(a, b):  
 return a + b, a - b  
  
*# get result in tuple format  
# unpack tuple*add, sub = calculation(40, 10)  
print(add, sub)

**Exercise 4: Create a function with a default argument**

Write a program to create a function show\_employee() using the following conditions.

* It should accept the employee’s name and salary and display both.
* If the salary is missing in the function call then assign default value 9000 to salary

**See**: [Default arguments in function](https://pynative.com/python-function-arguments/#h-default-arguments)

**Given**:

showEmployee("Ben", 12000)

showEmployee("Jessa")

**Expected output**:

Name: Ben salary: 12000

Name: Jessa salary: 9000

*# function with default argument*def show\_employee(name, salary=9000):  
 print("Name:", name, "salary:", salary)  
  
show\_employee("Ben", 12000)  
show\_employee("Jessa")

### Exercise 5: Create an inner function to calculate the addition in the following way

* Create an outer function that will accept two parameters, a and b
* Create an inner function inside an outer function that will calculate the addition of a and b
* At last, an outer function will add 5 into addition and return it
* def outer\_fun(a, b):  
   square = a \*\* 2  
     
   def addition(a, b):  
   return a + b  
     
   add = addition(a, b)  
     
   return add + 5  
  result = outer\_fun(5, 10)  
  print(result)

### Exercise 6: Create a recursive function

Write a program to create a **recursive function to calculate the sum of numbers** from 0 to 10.

A recursive function is a function that calls itself again and again.

**Expected Output**:

55

def addition(num):  
 if num:  
 # call same function by reducing number by 1return num + addition(num - 1)  
 else:  
 return 0  
  
res = addition(10)  
print(res)

### Exercise 7: Assign a different name to function and call it through the new name

Below is the function display\_student(name, age). Assign a new name show\_student(name, age) to it and call it using the new name.

**Given**:

**def** display\_student(name, age):

**print**(name, age)

display\_student("Emma", 26)

You should be able to call the same function using

show\_student(name, age)

def display\_student(name, age):  
 print(name, age)  
  
# call using original namedisplay\_student("Emma", 26)  
  
# assign new nameshow\_student = display\_student  
# call using new nameshow\_student("Emma", 26)

### Exercise 8: Generate a Python list of all the even numbers between 4 to 30

print(list(range(4, 30, 2)))

**OUTPUT:**

[4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28]

### Exercise 9: Find the largest item from a given list

x = [4, 6, 8, 24, 12, 2]

**Expected Output**:

24

x = [4, 6, 8, 24, 12, 2]  
print(max(x))

**Quiz:**

1. What is the output of the following display() function call

**def** display(\*\*kwargs):

**for** i **in** kwargs:

**print**(i)

display(emp="Kelly", salary=9000)

emp

salary

2. What is the output of the following function call

**def** outer\_fun(a, b):

**def** inner\_fun(c, d):

**return** c + d

**return** inner\_fun(a, b)

**return** a

result = outer\_fun(5, 10)

**print**(result)

15

3. Select which true for Python function

 A function is a code block that only executes when called and always returns a value.

 A function only executes when it is called and we can reuse it in a program

 Python doesn’t support nested function

a, b

4. What is the output of the following function call

**def** fun1(num):

**return** num + 25

fun1(5)

**print**(num)

NameError

5. Given the following function fun1() Please **select all the correct function calls**

**def** fun1(name, age):

**print**(name, age)

1. fun1("Emma", age=23)
2. fun1(age =23, name="Emma")
3. 6. What is the output of the following code?
4. **def** outer\_fun(a, b):
5. **def** inner\_fun(c, d):
6. **return** c + d
7. **return** inner\_fun(a, b)
8. res = outer\_fun(5, 10)
9. **print**(res)

15

7. What is the output of the following function call

**def** fun1(name, age=20):

**print**(name, age)

fun1('Emma', 25)

Emma 25

8. Select which is true for Python function

 A Python function can return only a single value

 A function can take an unlimited number of arguments.

 A Python function can return multiple values

 Python function doesn’t return anything unless and until you add a return statement

B, c

9. Choose the correct function declaration of  fun1() so that we can execute the following function call successfully

fun1(25, 75, 55)

fun1(10, 20)

def fun1(\*data)

10.Python function always returns a value

True

11. What is the output of the add() function call

**def** add(a, b):

**return** a+5, b+5

result = add(3, 2)

**print**(result)

(8, 7)

12. What is the output of the following display\_person() function call

**def** display\_person(\*args):

**for** i **in** args:

**print**(i)

display\_person(name="Emma", age="25")

TypeError