QB365 Question Bank Software Study Materials

Python Functions Important 2, 3 & 5 Marks Questions With Answers (Book Back and Creative)

12th Standard

Computer Science

Total Marks : 75

2 Marks

10 x 2 = 20

1) What is function?

Answer : (i) Functions are named blocks of code that are designed to do specific job.
(ii) If you need to perform that task multiple times throughout your program, you just call the function dedicated to handling that task.

2) Write the different types of function.

Answer : (i) User defined functions.
(ii) Built in functions.
(iii) Lambda functions.
(iv) Recursion functions.

3) What are the main advantages of function?

Answer : Main advantages of functions are
(i) It avoids repetition and makes a high degree of code reusing.
(ii) It provides better modularity for your application.

4) What is meant by scope of variable? Mention its types.

Answer : Scope of variable refers to the part of the program, where it is accessible, i.e., area where the variables can refer (use) it. The scope holds the current set of variables and their values. The two types of scopes - **local scope and global scope.**

5) Define global scope.

Answer : A variable with global scope can be used anywhere in the program. It can be created by defining a variable outside the scope of any function.

6) What is base condition in recursive function?

Answer : (i) A recursive function calls itself.
(ii) The condition that is applied in any recursive function is known as base condition.
(iii) A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

7) How to set the limit for recursive function? Give an example.

Answer : (i) Python stops calling recursive function after 1000 calls by default.
(ii) So, it also allows you to change the limit using sys.setrecursionlimit (limit_value).
Example:

```
import sys
sys.setrecursionlimit(3000)
def fact(n):
    if n == 0:
    return 1
    else:
    return n * fact(n-1)
print(fact (2000))
```

8) Write the output of the following program.

Answer : **Example:**

```
def hello():
print ("hello - Python")
return
print (hello())
```

Output:

```
hello - Python
None
```

9) What is local variable?

Answer : A variable declared inside the function's body or in the local scope is known as local variable.

10) What is anonymous function?

Answer : In Python, an anonymous function is a function that is defined without a name. While normal functions are defined using the def keyword, in Python anonymous functions are defined using the lambda keyword. Hence, anonymous functions are also called lambda functions.

3 Marks 10 x 3 = 30

11) Write the rules of local variable.

Answer : **Rules of local variable:**

(i) A variable with local scope can be accessed only within the function/block that it is created in.

(ii) When a variable is created inside the function the variable becomes local to it.

(iii) A local variable only exists while the function is executing.

(iv) The formal parameters are also local to function.

12) Write the basic rules for global keyword in python.

Answer : **Rules of global Keyword:** The basic rules for **globa**l keyword in Python are:

(i) When we define a variable outside a function, it's global by default. you don't have to use global keyword.

(ii) We use global keyword to modify the value of the global variable inside a function.

(iii) Use of global keyword outside a function has no effect.

13) What happens when we modify global variable inside the function?

Answer : If we modify the global variable, we can see the change on the global variable outside the function also.

Example:

```
x=0                                          # global variable
def add():
    global x
     x=x+5                                   # increment by 5
  print ("Inside add() function x value is:",x)
  add()
  print ("In main x value is:", x)
```

Output:

```
Inside add() function x value is: 5
In main x value is: 5                        #value of x changed outside the function
```

14) Differentiate ceil() and floor() function?

Answer :

S.NO	CEIL ( )	FLOOR ( )
(i)	Returns the smallest integer greater than or equal to x.	Returns the largest integer less than or equal to x.
(ii)	Syntax: math. ceil (x)	Syntax: math.floor (x)

15) Write a Python code to check whether a given year is leap year or not.

Answer : **Code:**

```
n=int(input("Enter the year"))
if(y%4==0):
        print ("Leap Year")
else:
        print ("Not a Leap Year")
```

Output:

Enter the year 2012

Leap Year

16) What is composition in functions?

Answer : (i) The value returned by a function may be used as an argument for another function in a nested manner.

(ii) This is called **composition**, For example, if we wish to take a numeric value or an expression as a input from the user, we take the input string from the user using the function **input()** and apply **eval()** function to evaluate its value.

17) How recursive function works?

Answer : (i) Recursive function is called by some external code.

(ii) If the base condition is met then the program gives meaningful output and exits.

(iii) Otherwise, function does some required processing and then calls itself to continue recursion.

18) What are the points to be noted while defining a function?

Answer : When defining functions there are multiple things that need to be noted;

(i) Function blocks begin with the keyword "**def**" followed by function name and parenthesis ().

(ii) If any input parameters are present should be placed within these parentheses when you define a function.

(iii) The code block always comes after colon(:) and is indented.

(iv) The statement "**return [expression]**" exits a function, optionally passing back an expression to the caller. A "return" with no arguments is the same as return None.

19) How python takes a default value in the function call? Explain with an example.

Answer : (i) In Python the default argument is an argument that takes a default value if no value is provided in the function call.

(ii) The following example uses default arguments, that prints default salary when no argument is passed.

(iii) **Example:**

```
def printinfo( name, salary = 3500):
print ("Name: ", name)
print ("Salary: ", salary)
return
printinfo("Mani")
```

When the above code is executed, it produces the following output

Output:

Name: Mani

Salary: 3500

When the above code is changed as print info("Ram", 2000) it produces the following output:

Output:

Name: Ram

Salary: 2000

20) Write a note on format () with an example.

Answer :

format ()	Returns the output based on the given format. 1. Binary format. Outputs the number in base 2. 2. Octal format. Outputs the number in base 8. 3. Fixed-point notation. Displays the number as a fixed-point number. The default precision is 6.	format (value [, format_spec])	x=14 y=25 print ('x value in binary :',format(x,'b')) print ('y value in octal :',format(y,'o')) print('y value in Fixed-point no ',format(y,'f')) **Output:** x value in binary: 1110 y value in octal: 31 y value in Fixed-point no: 25.000000

5 Marks 5 x 5 = 25

21) Explain the different types of function with an example.

Answer : Functions are named blocks of code that are designed to do one specific job.

Types of Functions :

(i) User-defined Functions

(ii) Built-in Functions

(iii) Lambda Functions

(iv) Recursion Function

User defined functions:

1. Functions defined by the users themselves are called user defined function.

2. Functions must be defined, to create and use certain functionality.

3.Function blocks begin with the keyword "def" followed by function name and parenthesis ().

Syntax:

```
def :


return
```

Example:

```
def hello ():
        print ("hello Python")
        return
```

Built-in function: Functions which are using Python libraries are called Built-in function.

```
x = 20
y = 23.2
print('x=',abs(x))
print ('y=', abs(y))
```

Output:

```
x = 20
y = 23.2
```

Lambda function :

1. In Python, anonymous function is a function that is defined without a name.

2. While normal functions are defined using the def keyword, in Python anonymous functions are defined using the **lambda** keyword.

3.Hence anonymous functions are also called as **lambda** functions.

Example:

```
    Sum =lambda arg1, arg 2: arg 1 + arg 2
     print ("The sum is:", sum (30,40))
     print ("The sum is:", sum (-30,40))
```

Output:

```
    The sum is: 70
    The sum is: 10
```

Recursion Function: Function that calls itself is known as recursive.

Overview of how recursive function works :

(a) Recursive function is called by some external code.

(b) If the base condition is met then the program gives meaningful output and exits.

(c) Otherwise, function does some required processing and then calls itself to continue recursion.

Example:

```
def fact(n):
        if n==0:
                return 1
else:
                return n* fact (n-1)
print (fact (0))
print (fact (5))
```

Output:

1

120

22) Explain the scope of variables with an example.

Answer : Scope of variable refers to the part of the program, where it is accessible, i.e., area where the variables refer (use). The scope holds the current set of variables and their values. The two types of scopes-local scope and global scope.

Local Scope: A variable declared inside the function's body is known as local variable.

Rules of local variable:

(i) A variable with local scope can be accessed only within the function that it is created in.

(ii) When a variable is created inside the function the variable becomes local to it.

(iii) A local variable only exists while the function is executing.

(iv) The formal parameters are also local to function.

(v) **Example:** Create a Local Variable

```
def loc():
        y=0 # local scope
        print(y)
loc()
```

Output:

0

Global Scope: A variable, with global scope can be used anywhere in the program. It can be created by defining a variable outside the scope of any function.

Rules of global Keyword:

The basic rules for global keyword in Python are:

(i) When we define a variable outside a function, it's global by default. You don't have to use global keyword.

(li) We use global keyword to modify the value of the global variable inside a function.

(iii) Use of global keyword outside a function has no effect.

Example: Accessing global Variable From Inside a Function

```
c = 1                          # global variable
def add():
        print(c)
        add()
```

Output:

1

23) Explain the following built-in functions.

(a) id ()

(b) chr ()

(c) round ()

(d) type ()

(e) pow ()

Answer :

Function	Description	Syntax	Example
(a) id()	id () Return the "identity" of an objects i.e. The address of the object in memory. **Note:** The address of x and y may differ in your system.	id(object)	x = 15 y = 'a' print('address of x is :',id(x)) print ('address of y is :',id (y)) **Output:** address of x is: 1357486752 address of y is: 13480736
(b) Chr()	Returns the Unicode character for the given ASCII value. This function is inverse of ord() function.	Chr(i)	c=65 d=43 print(chr(c)) print(chr(d)) **Output:** A +
(c) round()	Returns the nearest integer to its input. 1. First argument(number) is used to specify the value to be rounded.	round (number, [ndigits])	x = 17.9 y = 22.2 z = -18.3 print('x value is rounded to', round(x)) print('y value ic rounded to', round(y)) print ('z value is rounded to', round(z))
(d) type()	Returns the type of object for the given single object. **Note:** This function used with single object parameter.	type(object)	x = 15.2 y = 'a' s = true print(type(x)) print(type(y)) print(type(s)) **Output:** < class 'float' > < class 'str' > < class 'bool' >
(e) pow()	Returns the computation of ab i.e. (a**b) a raised to the power of b.	pow(a, b)	a = 5 b = 2 c = 3.0 print(pow(a, b)) print(pow(a, c)) print(pow(a + b, 3)) **Output:** 25 125.0 343

24) Write a Python code to find the L.C.M. of two numbers.

Answer : Program:

```
def lcm(x, y):
if x > y:
        greater = x
else:
        greater = y
while (True):
         if ((greater % x == 0) and (greater % y ==0))
         lcm = greater
         break
         greater + = 1
         return lcm
a = int (input ("Enter the first number:"))
b = int (input ("Enter the second number: "))
print ("The L.C.M of", a, "and", b, "is", lcm (a, b))
```

[OR]

Method II: (without using functions)

```
a = int (input ("Enter the first number:"))
b = int (input ("Enter the second number:"))
if a > b:
        mini a
else:
        min 1 = b
        while (1):
if (min 1% a 0 and mini 1% b=0):
print ("LCM is:", mini)
break
mini = min 1+1
```

Output:

```
Enter the first number: 15
Enter the second number: 20
LCM is: 60
```

25) Explain recursive function with an example.

Answer : (i) When a function calls itself is known as recursion.

(ii) Recursion works like loop but sometimes it makes more sense to use recursion than loop.

(iii) You can convert any loop to recursion. A recursive function calls itself.

(iv) Imagine a process would iterate indefinitely if not stopped by some condition is known as infinite iteration.

(v) The condition that is applied in any recursive function is known as base condition.

(vi) A base condition is must in every recursive function otherwise it will continue to execute like an infinite loop.

Overview of how recursive function works:

(i) Recursive function is called by some external code.

(ii) If the base condition is met then the program gives meaningful output and exits.

(iii) Otherwise, function does some required processing and then calls itself to continue recursion.

Here is an exanmple of recursive function used to calculate factorial.

Example:

```
def fact(n):
if n == 0:
            return 1
else:
            return n * fact (n-1)
print (fact (0))
print (fact (5))
```

Output:

```
120
```