

Functions

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

In Python a function is defined using the `def` keyword.

```
def <function_name>():  
    <your code here>
```

To call a function, use the function name followed by bracket.

```
def <function_name>():  
    print("Hello from my function")  
my_function()
```

Functions - Parameters

Information can be passed to functions as parameter. Parameters are specified after the function name, inside the brackets.

```
def hello(name):  
    print("Hello, " + name + "!")
```

```
hello("Alice")  
hello("Bob")  
hello("Charlie")
```

You can add as many parameters as you want, just separate them with a comma. You can send any data types of parameter to a function (string, number, list, dictionary etc.), and it will be treated as the same data type inside the function.

```
def <function_name>(<param1>,  
    <param2>...):  
    <your code here>
```

Functions - Return

To let a function return a value, use the `return` statement.

```
def <function_name>(<params>):  
    <your code here>  
    return <value>
```

You can have your function return a value, unless stated this will not be printed.

You can use the return data value from your function to input into other code where required.

Functions - Recursions

Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

```
def calc_factorial(x):  
    if x == 1:  
        return 1  
    else:  
        return (x * calc_factorial(x - 1))  
  
# calc_factorial(4)           # 1st call with 4  
# 4 * calc_factorial(3)       # 2nd call with 3  
# 4 * 3 * calc_factorial(2)    # 3rd call with 2  
# 4 * 3 * 2 * calc_factorial(1) # 4th call with 1  
# 4 * 3 * 2 * 1               # return from 4th call  
as number=1  
# 4 * 3 * 2                   # return from 3rd call  
# 4 * 6                       # return from 2nd call  
# 24                          # return from 1st call
```

In the above function: the first function is called, it takes the number 4 as a parameter, as this does not equal 1, it goes to the next code to execute which is multiplied by 4 multiplied by calc factorial 4 minus 3 and this keeps looping until x equals 1 and stops the loop.

This is then input into the print statement, which would print, "The factorial is 4 is 24".

CODING EXAMPLES

SECTION A

```

1. def hello_world():
    print("Hello World!")

2. def hello_world():
    print("Hello World!")
    hello_world()

3. def hello(name):
    print("Hello, " + name + "!")
    hello("Alice")
    hello("Bob")
    hello("Charlie")

4. def hello(name, age):
    print("Hello my name is " + name)
    print("I'm " + str(age) + " years old")

    age_in_10_years = age + 10
    print("In 10 years time I will be " +
          str(age_in_10_years))
    hello("Alice", 22)
    hello("Bob", 34)
    hello("Charlie", 17)

5. def area(x, y, z):
    print("The area is " + str(x * y * z))
    area(12, 3, 4)
    area(6, 14, 10)

6. def area(x, y, z):
    return x * y * z
    cube1 = area(12, 3, 4)
    cube2 = area(6, 14, 10)

7. def calc_factorial(x):
    if x == 1:
        return 1
    else:
        return (x * calc_factorial(x - 1))
    num = 4
    print("The factorial of " + num + " is " +
          str(calc_factorial(num)))

```

QUESTIONS

SECTION A

1. Write a function that prints your name.
2. Write a function that accepts a name as a parameter and prints "Hello, <name>".
3. Loop through the list ["Alice", "Bob", "Charlie"] and call the function you just wrote so it will print "Hello, <name>" for each name in the list.
4. Write a function that prints the area of two passed in parameters.
5. Write a function called 'print_list' that accepts a list as a parameter and then prints out each item of the list.
6. Put the following into a function:
 - i. If they are younger than 11, print "You're too young to go to this school".
 - ii. If they are between 11 and 16, print "You can can come to this school".
 - iii. If they are over 16, print 'You're too old for school'.
 - iv. If they are 0, print "You're not born yet!".

SECTION B

1. Write a function called is_odd that will return True or False if the integer passed as a parameter is odd (hint: $x \% 2$ will return true for all odd numbers).
2. Write a function that accepts a word and returns it backwards, e.g. 'hello' -> 'olleh'.
3. Write a recursive function that accepts a number and prints that number of stars, followed by ever decreasing stars on each line, E.g:

```
*****
****
***
**
*
```
4. Create a padlock function. You need to be able to pass in a passcode and if its correct it should return "Unlock", else "Locked".
5. Write a function that returns the sum of multiples of 3 and 5 between 0 and limit (parameter). For example, if limit is 20, it should return the sum of 3, 5, 6, 9, 10, 12, 15, 18, 20.
6. Write a function called is_prime() that accepts a number and return True or False if the number of prime or not.