# **Python 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.

## **Creating a Function**

In Python a function is defined using the def keyword:

**Example**

def my\_function():

print("Hello from a function")

## **Calling a Function**

To call a function, use the function name followed by parenthesis:

**Example**

def my\_function():

print("Hello from a function")

my\_function()

Arguments

* Information can be passed into functions as arguments.
* Arguments are specified after the function name, inside the parentheses.
* You can add as many arguments as you want, just separate them with a comma.
* The following example has a function with one argument (fname). When the function is called, we pass along a first name, which is used inside the function to print the full name:

### 

### 

### **Example:**

def my\_function(fname):

print(fname + " Refsnes")

my\_function("Emil")

my\_function("Tobias")

my\_function("Linus")

### **Example Program : Addition of two numbers :**

### add()

### def add():

### a=int(input("enter first no"))

### b=int(input ("enter sec no"))

### print("sum is",a+b)

### **Output:**

enter first no5

enter sec no3

sum is 8

**TASK 1a: Write a python program to find factorial of a given number.**

**AIM: To develop a Python program to find factorial of a given number.**

**PROGRAM:**

### n=int(input("enter the value of n"))

### fact=1;

### if(n==0):

### print("fact is",fact);

### else:

### for i in range(1,n+1):

### fact=fact\*i;

### print("fact is",fact);

**Output :**

### enter the value of n5

### fact is 120

**Factorial with recursion :**

n=int(input("enter the value of n"))

def fact(n):

if(n==0):

return 1

else:

return n\*fact(n-1)

fact(n)

**output:**

enter the value of n6

720

**Factorial without recursion or with function :**

n=int(input("enter the value of n"))

def fact(n):

fact=1;

if(n==0):

print("fact is",fact);

else:

for i in range(1,n+1):

fact=fact\*i;

print("fact is",fact);

fact(n)

**output:**

enter the value of n5

fact is 120

**TASK 1b: Write a Python program to find GCD of two numbers.**

**AIM: To write a Python program to find GCD of two numbers.**

**PROGRAM**

**Without function**

a=int(input("enter first no"));

b=int(input ("enter sec no"));

while a!=b:

if a>b:

a=a-b;

else:

b=b-a;

print("gcd is",a);

**output:**

### enter first no4

### enter sec no6

### gcd is 2

### Gcd with function :

a=int(input("enter first no"));

b=int(input ("enter sec no"));

def gcd(a,b):

while a!=b:

if a>b :

a=a-b;

else:

b=b-a;

print("gcd is",a);

gcd(a,b);

**output:**

enter first no1

enter sec no6

gcd is 1

**Gcd with recursion**

a=int(input("enter first no"));

b=int(input ("enter sec no"));

def gcd(a,b):

if a!=b:

if a>b :

return gcd(a-b,b);

else:

return gcd(a,b-a);

print("gcd is",a);

gcd(a,b);

**Output:**

**enter first no8**

**enter sec no9**

# Python Lists

**mylist = ["apple", "banana", "cherry"]**

## List

**Lists are used to store multiple items in a single variable.**

**Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are** [**Tuple**](https://www.w3schools.com/python/python_tuples.asp)**,** [**Set**](https://www.w3schools.com/python/python_sets.asp)**, and** [**Dictionary**](https://www.w3schools.com/python/python_dictionaries.asp)**, all with different qualities and usage.**

**Lists are created using square brackets:**

### Example

### **Cr**

### 

### **eate a List:**

**thislist = ["apple", "banana", "cherry"]**

**print(thislist)**

## List Items

**List items are ordered, changeable, and allow duplicate values.**

**List items are indexed, the first item has index [0], the second item has index [1] etc.**

## Ordered

**When we say that lists are ordered, it means that the items have a defined order, and that order will not change.**

**If you add new items to a list, the new items will be placed at the end of the list.**

**Note: There are some** [**list methods**](https://www.w3schools.com/python/python_lists_methods.asp) **that will change the order, but in general: the order of the items will not change.**

## Changeable

**The list is changeable, meaning that we can change, add, and remove items in a list after it has been created.**

## Allow Duplicates

**Since lists are indexed, lists can have items with the same value:**

### Example

**Lists allow duplicate values:**

**thislist = ["apple", "banana", "cherry", "apple", "cherry"]**

**print(thislist)**

**Example:**

list=[("abc",65),("xyz",45)];

print("list is",list);

**Output :**

list is [('abc', 65), ('xyz', 45)]

# Python Lambda

A lambda function is a small anonymous function.

A lambda function can take any number of arguments, but can only have one expression.

## Syntax

lambda *arguments* : *expression*

The expression is executed and the result is returned:

### Example

Add 10 to argument a, and return the result:

x = lambda a : a + 10

print(x(5))

Lambda functions can take any number of arguments:

### Example

Multiply argument a with argument b and return the result:

x = lambda a, b : a \* b

print(x(5, 6))

### Example

Summarize argument a, b, and c and return the result:

x = lambda a, b, c : a + b + c

print(x(5, 6, 2))

**TASK 1c: Write a Python program that reads a list of names and ages,then prints the list sorted by age**

**AIM: Write a Python program that reads a list of names and ages,then prints the list sorted by age**

**PROGRAM:**

**n=int(input("enter the value of n"))**

**list=[];**

**for i in range (n):**

**name=input("entr name");**

**age=int(input("enter age"))**

**list.append((name,age));**

**list.sort(key=lambda t:t[1])**

**print("list after sort",list);**

**Output :enter the value of n2**

**enter name xyz**

**enter age45**

**entr nameabc**

**enter age56**

**list after sort [('abc', 56), ('xyz', 45)]**

**List without lambda function**

def sec(ele):

return ele[1]

n=int(input("enter the value of n"))

list1=[];

for i in range (n):

name=input("entr name");

age=int(input("enter age"))

list1.append((name,age));

list1.sort(key=sec)

print("list after sort",list1);

**Out put:**

enter the value of n2

entr namewe

enter age45

entr namert

enter age12

list after sort [('rt', 12), ('we', 45)]

**List with lambda function**

n=int(input("enter the value of n"))

list1=[];

for i in range (n):

name=input("entr name");

age=int(input("enter age"))

list1.append((name,age));

list1.sort(key=lambda t:t[1])

print("list after sort",list1);

**Output:enter the value of n2**

**entr nameert**

**enter age43**

**entr nametry**

**enter age12**

**list after sort [('try', 12), ('ert', 43)]**