Computational Analysis of Physical Systems (Lecture 5)

Functions in Python

Functions - 1

```
def functionname( parameters ):
   "function_docstring"
   function_suite
   return [expression]
```

```
def printme( str ):
   "This prints a passed string into this function"
   print str
   return
```

```
# Now you can call printme function
printme("I'm first call to user defined function!");
printme("Again second call to the same function");
```

Functions - 2

```
# Function definition is here
def changeme( mylist ):
   "This changes a passed list into this function"
   mylist = [1,2,3,4]; # This would assign new reference in mylist
   print "Values inside the function: ", mylist
   return

# Now you can call changeme function
mylist = [10,20,30];
changeme( mylist );
print "Values outside the function: ", mylist
```

```
Values inside the function:  [1, 2, 3, 4]
Values outside the function:  [10, 20, 30]
```

Functions - 3

```
# Function definition is here
def printme( str ):
   "This prints a passed string into this function"
   print str;
   return;

# Now you can call printme function
printme( str = "My string");
```

```
My string
```

Functions - 4

```
# Function definition is here
def printinfo( name, age ):
   "This prints a passed info into this function"
   print "Name: ", name;
   print "Age ", age;
   return;

# Now you can call printinfo function
printinfo( age=50, name="miki" );
```

```
Name:  miki
Age  50
```

Functions - 5

```
# Function definition is here
def printinfo( name, age = 35 ):
   "This prints a passed info into this function"
   print "Name: ", name;
   print "Age ", age;
   return;

# Now you can call printinfo function
printinfo( age=50, name="miki" );
printinfo( name="miki" );
```

```
Name:  miki
Age  50
Name:  miki
Age  35
```

Functions - 6

```
# Function definition is here
def sum( arg1, arg2 ):
   # Add both the parameters and return them."
   total = arg1 + arg2
   print "Inside the function : ", total
   return total;

# Now you can call sum function
total = sum( 10, 20 );
print "Outside the function : ", total
```

```
Inside the function :  30
Outside the function :  30
```

Functions - 7

```
total = 0; # This is global variable.
# Function definition is here
def sum( arg1, arg2 ):
   # Add both the parameters and return them."
   total = arg1 + arg2; # Here total is local variable.
   print "Inside the function local total : ", total
   return total;

# Now you can call sum function
sum( 10, 20 );
print "Outside the function global total : ", total
```

```
Inside the function local total :  30
Outside the function global total :  0
```

Functions - 8

```
from numpy import *

def sum(mat1,mat2):
    # Add two matrices
    mat3=mat1+mat2
    return mat3;

m1=array([[1,2],[3,4]])
m2=array([[4,3],[2,1]])
print sum(m1,m2)
```

Functions - 9

- Now try this to “append” two matrices:

```
def sum(mat1,mat2):
    # Add two matrices
    mat3=mat1+mat2
    return mat3;

m1=[[1,2],[3,4]]
m2=[[4,3],[2,1]]
print sum(m1,m2)
```

Exercise - 1

- In the *main program*:
  - Ask the user to enter two numbers (x and y). (These numbers are two sides of a right triangle other than the hypotenuse.)
  - If x or y is zero or negative, print "No triangle" to the screen.
  - If x and y are nonzero or positive, call a *function* that finds the hypotenuse of this triangle if x and y are nonzero and positive.
  - Print the result on the screen.

- In the *function*:
  - The inputs are two numbers x and y.
  - The output is the hypotenuse of the triangle.

Exercise - 2

***main program*:**

- Enter a 3x3 matrix M
- Send this matrix to the function "minmaxvectors"
- Send the output vectors of the function "minmaxvectors" to the function "maxminofall"
- Print the results on the screen

"minmaxvectors" function:

- INPUT: A 3x3 matrix

- OUTPUTS: Two vectors (vmax and vmin) of three elements where,

  --The first element of vmax is the maximum value of the first row of the input matrix.
  --The second element of vmax is the maximum value of the second row of the input matrix.
  --The third element of vmax is the maximum value of the third row of the input matrix.

  --The first element of vmin is the minimum value of the first row of the input matrix.
  --The second element of vmin is the minimum value of the second row of the input matrix.
  --The third element of vmin is the minimum value of the third row of the input matrix.

"maxminofall" function:

-INPUTS: Two vectors of three elements. One containing the maximum values(vmax) and the other containing the minimum values (vmin)

- OUTPUTS: Two numbers maxofall,minofall where,
  -- maxofall is the maximum value in vector vmax (i.e. the maximum element of the matrix).
  -- minofall is the minimum value in vector vmin (i.e. the minimum element of the matrix).