Python

{ Functions

Violet Cullors

Why Functions?

- DEFINITION: A function is a block of organized, reusable code that is used to perform a single, related action.
- Group statements
- Improves clarity
- Eliminate repetitive code
- Divides a complex program into simpler parts
- A useful function can be updated and reused

Type of Functions - 3 Types

- Imported from a module:

```
from math import sqrt

print sqrt(81)
```

- Built in functions part of the program core
  - bin() – convert an integer into binary
  - print() – output string
  - range(start,stop,step) – use in for loops

- User Defined:

```
def cube(x):
    return x * x * x
```

Define a Function

SYNTAX
def functionname(parameters):
    statement 1
    statement N
    return [expression]

Returns a result :

```
def addUp (a, b):
        total  = a + b
        return total
```

Execute code - returns no results

```
def read_fineprint ():
        print ("This is the fine print of the contract…….")
```

A function can take in parameters or not

getTotal = addUp(7,8) *getTotal = 15*

read_fineprint()
"this is the fine print of the contract…."

Function parameters are passed in by Reference – changes done to the parameter reflect back into the calling program

```
string1 = "This is the initial string"

Def  addOn(test_string):
        test_string = test_string + " & add this part."

##MAIN PROGRAM
addOn(string1)
print(string1)
```

OUTPUT

This is the initial string & add this part

Scope of Functions – variables created in a function are local.

```
total = 0; # This is global variable

def sum( arg1, arg2 ):
        total = arg1 + arg2;
        print ("Inside the function local total : " total)
        return total

##MAIN PROGRAM
sum( 10, 20 )
print ("Outside the function global total : " total )
```

OUTPUT

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

Try this:

```
def tempConverter(temp, num):

    def celsius_to_fahrenheit(c_temp):
        return 9.0 / 5.0 * c_temp + 32

    def fahrenheit_to_celsius(f_temp):
        return (f_temp - 32.0) * 5.0 / 9.0

    if num == 1:
        result = celsius_to_fahrenheit(temp)
        print (str(temp) + " celsius to fahrenheit:")
    elif num == 2:
        result = fahrenheit_to_celsius(temp)
        print (str(temp) + " fahrenheit to celsius:")
    else:
        result = 0
        print("Temp type not valid:")

    print(result)

##MAIN PROGRAM

tempConverter(12.78,1)
tempConverter(100,2)
tempConverter(100,3)
```

OUTPUT:

12.78 celsius to fahrenheit:
55.004
100 fahrenheit to celsius:
37.77777777777778
Temp type not valid:
0