Nicole Angel B. Lopena

CS3C

Functions  
  
● Defining a Function

> 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.

Code Example:

def my\_function():

print("Hello from a function")

● Reasons of Using Functions

> Using functions, we may avoid writing the same logic or code in software many times.

In a single application, we can call Python functions everywhere and more than once.

We can easily trace a Python programme that has been divided up into numerous functions.

● Types of Functions in Python

> Built-in library function: These are Standard functions in Python that are available to use.

User-defined function: We can create our own functions based on our requirements.

● Advantages of User – Defined Function

> We can build definitions that are not included in the built-in Python functions by using a user-defined function. It can assist in meeting our needs. There are some repetitive code parts in most programs. We can write the code for these sections of the code using a user-defined function.

● Rules in Declaring a Function in Python

> In Python, the def keyword is used to define a function. The function name is then written, enclosed in parentheses, and followed by a colon. After making sure you indent with a tab or four spaces, you need to tell the function what you want it to do for you.

● Python Function Syntax

> A function is a block of code that performs a specific task.

Code Example:

def greet():

print('Hello World!')

# call the function

greet()

print('Outside function')

● Function Argument and Parameter

> Information sent into a function is what both the phrases parameter and argument refer to. From the viewpoint of a function, a parameter is the variable specified in the function definition inside parenthesis. The value that is sent to the function when it is called is called an argument.

Code Example:

def my\_function(fname, lname):  
  print(fname + " " + lname)  
  
my\_function("Emil", "Refsnes")

● The Return Statement

> The function call is terminated by using a return statement, which also "returns" to the caller the result (the value of the expression that comes after the return keyword). There is no execution of the statements that follow the return statements. The special value None is returned in the event that there is no expression in the return statement. In general, a function is called using a return statement to allow the passed statements to be executed.

Code Example:

def cube(x):

r=x\*\*3

return r