

Defining and Using Functions

Using a function consists of two parts: 1) *Defining it* and 2) *Calling it*.

To define a function, you start with the keyword `def` and then the name of the function, open and close parentheses and end with a `:`

The body of the function contains the Python statements that will be executed when the function is called. Note that the body should be indented, just like in the case of the if statement and the while and for loops. If the function requires any inputs, the variable names for these should be listed (separated by commas) inside the parentheses. This list of variable names is referred to as the parameters of the function.

Note some functions return a value while others do not. A function which returns a value is called a `fruitful` function.

In [1]:

```
'''
Write a function definition that will accept an integer value as an argument
and return the cubed value.
'''
#start the function definition with the def keyword. End with a ':'. Include the required inputs
inside parentheses
def cube_int(i):
    result = i ** 3
    return result
```

To call the function we pass it the appropriate inputs. The term argument(s) refers to the value or variable name provided as input(s) when calling the function. These arguments must be listed in the same order as the parameters in the function definition.

If the function being called is a fruitful function, we must make sure that we capture the return value by assigning it to a variable. Otherwise, the value returned by the function is lost.

Some points to note (test out):

- 1: What happens if the function does not have a return statement
- 2: What happens when you call the `cube_int()` function by itself (without the print statement)
- 3: What happens if you try to assign the return value from a function which has no return statement?

In [3]:

```
'''
To call a function, we write the name of the function and provide the argument values inside parentheses
'''
def cube_int(i):
    result = i ** 3
    return result

n = int(input('Enter an integer: '))
print(cube_int(n))
```

Enter an integer: 3
27

In [4]:

```
'''
To call a function, we write the name of the function and provide the argument values inside parentheses. Note that
the arguments must be in the same order as the parameters in the function definition.
'''
def divide_int(i, j):
    result = i / j
```

```
    return result

n1 = int(input('Enter an integer: '))
n2 = int(input('Enter another integer: '))
print(divide_int(n1, n2)) ##Note that switching the order of n1 and n2 will change the results
```

```
Enter an integer: 16
Enter another integer: 2
8.0
```

In [5]:

```
'''
Any function can be called (invoked) multiple times.
'''
def cube_int(i):
    result = i ** 3
    return result

# Print the cube of all integers between 1 and 10
for i in range(1,11):
    print(cube_int(i))
```

```
1
8
27
64
125
216
343
512
729
1000
```