

User-Defined Single Result (or Pure) Functions Review

A function can be thought of as a self-contained “mini-program” that does a specific task. A **single result function** (also called a **pure function**) usually accepts one or more values called paramters or arguments and are enclosed in brackets (). Single result functions always only return one result.

A single result (pure) function is not **used on its own** - it is most often used with another function (ie. the **print** function) or assigned to a variable.

Two Python functions which are **never** found in single result (pure) functions are input() and print().

Local Variables vs. Global Variables

Global variables

- Declared as soon as it is required in the main program.
- Its scope (ie. where it can be used), is the entire program after it has been declared.
- If a global variable is assigned a value, that value can be used throughout the program.
- The value stored in a global variable can be also changed anywhere in the program.

Local variables

- Declared within a function.
- Its scope is only within the function in which it has been declared.
- The computer "forgets" that a local variable has been declared and "forgets" about any value stored in a local variable after the function in which it is declared is executed.

Formal Parameters vs Actual Parameters

- When a function is created, it is given **formal parameters** (or arguments).

Example: `def aFunction (formalParameter)`

- When the function is called, the values passed to the function are called **actual parameters** (or arguments).

Example: `print (aFunction (actualParameter))`

- Note that the order of formal parameters and actual parameters must correspond exactly or incorrect results may occur.

Programming Examples Explaining Single Result Functions

'''A program that will use a function to calculate the volume of a series of shapes with a given radius and height '''

Version ONE

```
import math    # Imports the math module to use the math.pi() function
```

```
def cylinderVolume (radius, height) :
```

```
    # Returns the volume of a cylinder
```

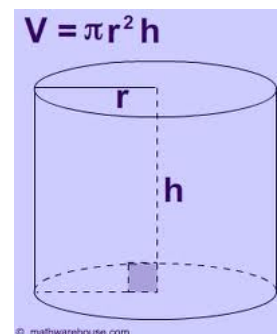
```
    # radius and height are the radius and height of the cylinder. They are formal parameters
```

```
    volume = math.pi * radius ** 2 * height    # The volume of the cylinder
```

```
    # volume is a local variable
```

```
    # Note that using math.pi () will result in an error
```

```
    return volume
```



Main program

```
cylRadius = float(input("Enter radius of the cylinder ")) # Stores the height of the cylinder.  
                                                    # This is a global variable  
cylHeight = float(input("Enter height of the cylinder ")) # Stores the height of the cylinder.  
                                                    # This is a global variable  
  
rightVolume = cylinderVolume(cylRadius, cylHeight) # This would return the correct answer  
                                                    # cylHeight and cylRadius are actual parameters  
                                                    # rightVolume is a global variable  
  
wrongVolume = cylinderVolume(cylHeight, cylRadius) # This would return the wrong answer as the  
                                                    # order of the actual parameters does not match  
                                                    # the order of the formal parameters  
  
print("The cylinder volume with a height of ", cylHeight, "cm and a radius of ", cylRadius, "cm is", rightVolume, "cm**3")
```

'''A program that will use a function to calculate the volume of a series of shapes with a given radius and height '''

Version Two

```
import math # Imports the math module to use the math.pi() function  
  
def cylinderVolume(radius, height):  
    # Returns the volume of a cylinder  
    # radius and height are the radius and height of the cylinder. They are formal parameters  
  
    volume = math.pi * radius ** 2 * height # The volume of the cylinder  
                                                    # volume is a local variable  
                                                    # Note that using math.pi () will result in an error  
  
    return volume
```

Main program

```
radius = float(input("Enter radius of the cylinder ")) # Stores the radius of the cylinder.  
                                                    # This is a global variable  
height = float(input("Enter height of the cylinder ")) # Stores the height of the cylinder.  
                                                    # This is a global variable  
  
volume = cylinderVolume(radius, height) # volume is a global variable - it is a different variable than volume above  
                                                    # radius and height are actual parameters - different from radius and height above  
  
print("The volume of a cylinder with a height of ", height, "cm and a radius of ", radius, "cm is", volume, "cm**3")  
  
otherVolume = cylinderVolume(10, 20) # volume is a global variable  
                                                    # 10 and 20 are actual parameters (but not variables)  
print("The volume of a cylinder with a height of 20 cm and a radius of 10 cm is", otherVolume, "cm**3")
```