

Calling the Function

The Role of a Function

If we look at the `IsEven` function again:

```

IsEven FUNCTION

def IsEven (InputNumber) :

    if (InputNumber % 2) == 0:
        ReturnValue = True  #it's even
    else:
        ReturnValue = False #it's odd
    # EndIf;

    return ReturnValue

# END IsEven.
```

We note that the `IsEven` function only does one thing, it takes in a value and checks if it's even or not, it doesn't deal with asking the user for a number or printing a message out to the user. This is the key philosophy behind the design of functions, *they should do one thing, and do it well*. So a function shouldn't try to do two or three things, or even half a job, it should always strive to do just one thing, as well as possible.

Calling the Function

When the main program uses the name of the function, we say "*the program is calling the function*", and we saw an example of that, where we say the following:

```
print (IsEven (4))
```

We can also call the function in a way that includes getting input from the user, and prints an answer back to them. We do this using the `Input` function to get a value from the user, then we call the `IsEven` function as part of the condition of an `IF` statement, and if the `IsEven` function returns `True` we print out the number with the message that the number is even, otherwise we print out the number is odd:

```
GetNumber = int(input("Input number:\n"))

if (IsEven(GetNumber) == True):
    print(GetNumber, "is an even number")
else:
    print(GetNumber, "is an odd number")
# Endif;
```

Note that the name of the variable being passed into the function from the main program is called `GetNumber`, but when the value for that variable is received by the function, it is stored in a new variable with a different name `InputNumber`. This is normal programming practice, and it means the programmer who writes the main program doesn't need to know anything about how the function is written.

#PythonMonday © Damian Gordon