

Prime Number Function

Prime Number Function

We've already seen how to check if a number is prime or not, we just divide it by all the numbers less than it and greater than one, and if any of them divide evenly into the number (i.e. gives no remainder) then we know that the number isn't prime. To take the program we have and make it into a function, the first thing we do is to get the number under investigation, passed into the function as a parameter, in this case called `InputNumber`. We do the computation exactly the same way as before, and then we return an answer back to the main program, in this case a Boolean value, where `True` means the value is prime, and `False` means that the value is not prime:

IsPrime FUNCTION

```
def IsPrime(InputNumber):
    Countdown = InputNumber - 1
    ReturnValue = True
    while (Countdown > 1):
        if (InputNumber % Countdown == 0):
            ReturnValue = False
        # EndIf;
        Countdown = Countdown - 1
    # EndWhile;
    return ReturnValue

# END IsPrime.
```

The code in the main part of the program is almost exactly the same as previous examples, except that this time it calls the function `IsPrime`. So the main program deals with the Input and Output to the users (sometimes called "I/O"), and the calculations and computations are done by the function:

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

if (IsPrime(GetNumber) == True):
    print("It's a prime number")
else:
    print("It's not a prime number")

# Endif;
```

And if the number inputted is 11, then we will get the following output:

```
It's a prime number
```

We note again that it is alright if the name of the variable that passes the value into the function is different from the variable that receives that value in the module.

#PythonMonday © Damian Gordon