VIDEO 10 : FUNCTIONS 2

In this video I'll show you how to return and receive multiple values. We'll also calculate primes, calculate areas for different shapes and talk about main().

To return multiple values just separate values returned with commas.

CODE

```
def mult_divide(num1, num2):
    return (num1 * num2), (num1 / num2)

mult, divide = mult_divide(5, 4)

print("5 * 4 =", mult)
print("5 / 4 =", divide)
```

Return a List of Primes

A prime can only be divided by 1 and itself. 5 is prime because 1 and 5 are its only positive factors. 6 is a composite because it is divisible by 1, 2, 3 and 6.

We'll receive a request for primes up to the input value. We'll then use a for loop and check if modulus == 0 for every value up to the number to check. If modulus == 0 that means the number isn't prime.

CODE

```
def is_prime(num):
    # This for loop cycles through primes from 2 to
    # the value to check
    for i in range(2, num):

        # If any division has no remainder we know it
        # isn't a prime number
        if (num % i) == 0:
            return False
    return True


def get_primes(max_number):

    # Create a list to hold primes
    list_of_primes = []

    # This for loop cycles through primes from 2 to
    # the maximum value requested
    for num1 in range(2, max_number):

        if is_prime(num1):
            list_of_primes.append(num1)

    return list_of_primes
```

```
max_num_to_check = int(input("Search for Primes up to : "))

list_of_primes = get_primes(max_num_to_check)

for prime in list_of_primes:
    print(prime)
```

Unknown Number of Arguments

We can receive an unknown number of arguments using the splat (*) operator

CODE

```
def sumAll(*args):
    sum = 0

    for i in args:
        sum += i
        return sum

print("Sum :", sumAll(1,2,3,4))
```

Route to Different Functions & Main()

Here we will route to different functions depending on what type of shape we want to get the area for. We will also use a main function for the first time.

CODE

```
import math

# This routes to the correct area function
# The name of the value passed doesn't have to match
def get_area(shape):

    # Switch to lowercase for easy comparison
    shape = shape.lower()

    if shape == "rectangle":
        rectangle_area()
    elif shape == "circle":
        circle_area()
    else:
        print("Please enter rectangle or circle")

# Create function that calculates the rectangle area
def rectangle_area():
    length = float(input("Enter the length : "))
    width = float(input("Enter the width : "))

    area = length * width
```

```
    print("The area of the rectangle is", area)


# Create function that calculates the circle area
def circle_area():
    radius = float(input("Enter the radius : "))

    area = math.pi * (math.pow(radius, 2))

    # Format the output to 2 decimal places
    print("The area of the circle is {:.2f}".format(area))

# We often place our main programming logic in a function called main
# We create it this way

def main():

    # Our program will calculate the area for rectangles or circles

    # Without functions we'd have to create a giant list of ifs, elifs

    # Ask the user what shape they have
    shape_type = input("Get area for what shape : ")

    # Call a function that will route to the correct function
    get_area(shape_type)

    # Because of functions it is very easy to see what is happening
    # For more detail just refer to the very short specific functions

# All of the function definitions are ignored and this calls for main()
# to execute when the program starts

main()
```