More Fun With Functions

How does a function actually function? Let's take a look...

Functions allow us to repeat process again and again, whenever we need. We do this by calling them - using their name (or identifier) and passing in any data that might be required. The job of the function is to process this data, and **return** to the function caller.

Return

Look at the code to the right - we define a function called **add_up()**, which requires us to give it two arguments when we call it - **num1** and **num2**. The job of the function is to add up **num1** and **num2**. Its job isn't to show us the result - nowhere in the function have we asked to see the result using **print()**. We might want to do many different things with the result of that sum.....so how do we get that result?

```
1   def add_up(num1,num2):
2       num1+num2
3
4   add_up(7,3)
```

Return

We call a function using its name - and we can do this anywhere. We can call a function as part of an if or elif statement, even inside another function - like **print()**!
On our right, we call the function within a print statement, so we're asking the print statement to print out the result of **add_up(7,3)**

```
4   print(add_up(7,3))
```

Return

In our function, we've asked for the variables **num1** and **num2** to be added together - but we haven't done anything with the result. The function **returns** data to its caller - but without us telling us what data to **return**, it can only **return None**, or empty data. So we use return to tell the function what it should give back to us. In this case return **num1+num2** - bring back to the function caller the result of **num1+num2**

```
1   def add_up(num1,num2):
2       return num1+num2
3
4   print(add_up(7,3))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
10
```

Return

We can ask **return** to return anything - even a completely irrelevant string, so be very precise with what you're returning! **return** also has a second job - it ends the function there, so if you write out anything after **return** in your defintion, those lines won't run! We can see Line 3 on our right is greyed out!

```
1   def add_up(num1,num2):
2       return num1+num2
3       print("Hello")
4
```

How is this helpful?

It might seem REALLY OBVIOUS to us that if we're adding up two numbers, we want the result of that - but code doesn't make assumptions! It only does what we ask it to. There are lots of ways we could have got the result of add_up() - but using return is very efficient! All functions return to their caller - that's their job, so be aware of any paths you might create by having functions call functions call functions.....

{cn}®