

# **KIT100**

# **Programming**

# **Preparation**

Tutorial Seven – Week 8



# • **Today's Flow**

- Walk through Portfolio Tasks PP8.1
- One-on-one session with me
  - to help you to resolve the programming concerns from the previous weeks



# Functions

def : keyword to define a function.

Function name

Function parameters

Function body

```
def welcome(first_name):  
    print("Welcome!", first_name)
```

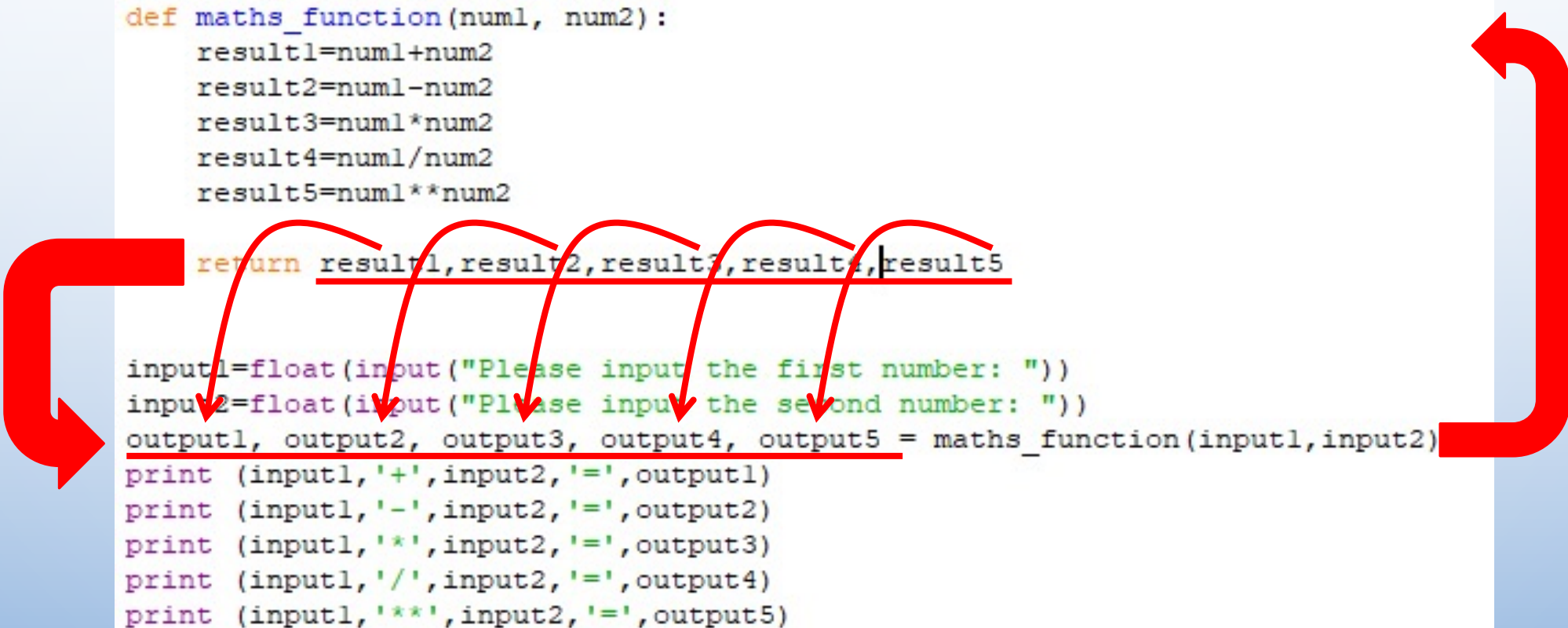
indentation

```
def functionName():  
    .....  
    .....  
    return result
```

```
a = functionName()
```

# Functions

```
def maths_function(num1, num2):  
    result1=num1+num2  
    result2=num1-num2  
    result3=num1*num2  
    result4=num1/num2  
    result5=num1**num2  
  
    return result1,result2,result3,result4,result5  
  
input1=float(input("Please input the first number: "))  
input2=float(input("Please input the second number: "))  
output1, output2, output3, output4, output5 = maths_function(input1,input2)  
print (input1,'+',input2,'=',output1)  
print (input1,'-',input2,'=',output2)  
print (input1,'*',input2,'=',output3)  
print (input1,'/',input2,'=',output4)  
print (input1,'**',input2,'=',output5)
```



The diagram illustrates the flow of data between the function definition and its call. A large red arrow on the left points from the function call line to the function definition. A large red arrow on the right points from the function definition back to the function call line. Five smaller red arrows originate from the return statement in the function definition and point to the five output variables in the function call, showing how the function's return values are assigned to the calling code's variables.

# Python String Functions

- Python provides rich amount of operations for you to work on your string.
  - `islower()`
  - `isdigit()`
  - `capitalize()`
  - `count()`
  - ....
- More details @ **Week 7 Lecture – Working with Strings and Lists**, or...
- Official documentation:  
<https://docs.python.org/3/library/stdtypes.html#string-methods>

Method	Description
<code>capitalize()</code>	Converts the first character to upper case
<code>casefold()</code>	Converts string into lower case
<code>center()</code>	Returns a centered string
<code>count()</code>	Returns the number of times a specified value occurs in a string
<code>encode()</code>	Returns an encoded version of the string
<code>endswith()</code>	Returns true if the string ends with the specified value
<code>expandtabs()</code>	Sets the tab size of the string
<code>find()</code>	Searches the string for a specified value and returns the position of where it was found
<code>format()</code>	Formats specified values in a string
<code>format_map()</code>	Formats specified values in a string
<code>index()</code>	Searches the string for a specified value and returns the position of where it was found
<code>isalnum()</code>	Returns True if all characters in the string are alphanumeric
<code>isalpha()</code>	Returns True if all characters in the string are in the alphabet
<code>isdecimal()</code>	Returns True if all characters in the string are decimals
<code>isdigit()</code>	Returns True if all characters in the string are digits
<code>isidentifier()</code>	Returns True if the string is an identifier
<code>islower()</code>	Returns True if all characters in the string are lower case
<code>isnumeric()</code>	Returns True if all characters in the string are numeric



# Portfolio Task

## • 8.1PP More Complex Functions

### Description:

The pharmacist from task 6.1PP now has a more difficult problem. They have a collection of medicine bar codes that contains a mixture of numbers and letters. There's a secret 'checksum' that can be calculated by adding all the digits in a bar code together (ignoring the letters). They need your help to write a program to do this - the user will enter all the characters (letters and numbers) from a bar code and your program should print the checksum along with how many digits were found.

### Task:

Write a function called `getChecksum` that takes one named parameter `barcode`. The function should then calculate the checksum by adding all the digits together that are found in the barcode, and it should return the checksum value to the caller as well as how many digits were found in the barcode. For example, if the input barcode is "abc456", the `getChecksum` will return the checksum value "15" and digits "3". **NOTE the function `getChecksum` does not print anything itself!**

In the main code, the user should be asked for a value for the requested barcode, and the `getChecksum` function is called with this data. The data **returned** by the function should then be printed to the screen with a suitable message. The user should then be asked if they want to run the program again (answer 'y') and the program should repeat the entire process until they answer 'n' to the *run again* prompt.

### Hint

You **must** define the function `getChecksum` and it must have one parameter, `barcode`. Your main code will need a *while* loop that calls the `getChecksum` function repeatedly until the user decides they don't want to run it any more.

- Look at some of the String functions defined in the week 7 lecture - one in particular is very useful to identify individual numbers in a string..
- A function can return more than one data item by separating each item by a comma in the return statement. To *receive* the data, the caller can define a comma-separated list of variables on the left hand side of the assignment (=) operator  
e.g. `a,b,c = myFunction()` (if `myFunction` returns three items)
- The `getChecksum` function will need a *for* loop to examine each character in the supplied barcode

### Example output:

```
Enter the barcode:abc
The checksum is 0 and 0 digits were entered

Do you want to run this again? (y/n):y

Enter the barcode:abc456
The checksum is 15 and 3 digits were entered

Do you want to run this again? (y/n):y

Enter the barcode:ab123def
The checksum is 6 and 3 digits were entered

Do you want to run this again? (y/n):y

Enter the barcode:aab11109x
The checksum is 12 and 5 digits were entered

Do you want to run this again? (y/n):n
Goodbye!
```