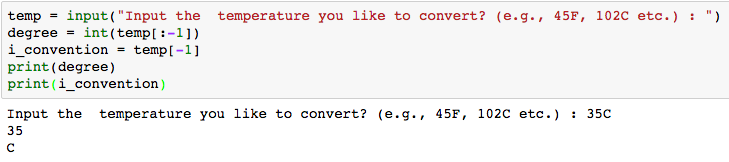
|  |
| --- |
| Worksheet 7 |

1) The following Python code takes temperature from a user and put all numerical characters in variable degree and character in variable i\_convention.



* Write a Python program to convert temperatures to and from celsius, fahrenheit.

[ Formula : c/5 = f-32/9 [ where c = temperature in celsius and f = temperature in fahrenheit ]

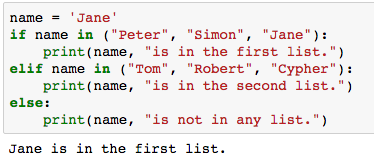
Expected Output:

60C is 140 in Fahrenheit

45F is 7 in Celsius

2) Write a Python code to take a distance and convert the distance to and from kilometer (km) and mile (mi). For example, if 10km is entered, the output is 16mi. If 32mi is entered, the output is 20km.

3) The following Python code checks if variable name (‘Jane’) appears in which list.



The following Python code takes name of month (3-char abbreviation, eg. Jan, Feb, etc.) as an input and print the number of days of the entered month



* Complete the Python code for other months.

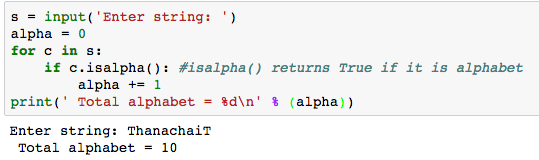
Expected Output:

List of months: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

Enter month (e.g., Jan, Feb, Mar): Aug

No. of days: 30 days

4) The following Python code counts the number of alphabets from the string you enter.



Note:

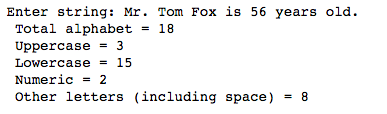
isalpha( ) returns True if it is an alphabet.

isupper( ) returns True if it is an uppercase letter

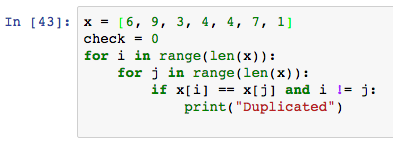
islower() returns True if it is a lowercase letter

isdigit() returns True it is a numeric

* Modify the code to print the total number of alphabets, uppercase letters, lowercase letters, numeric and other letters. An example of input and output is as follows.



5) The following Python code prints duplicated if any of the values in the list *x* is duplicated.



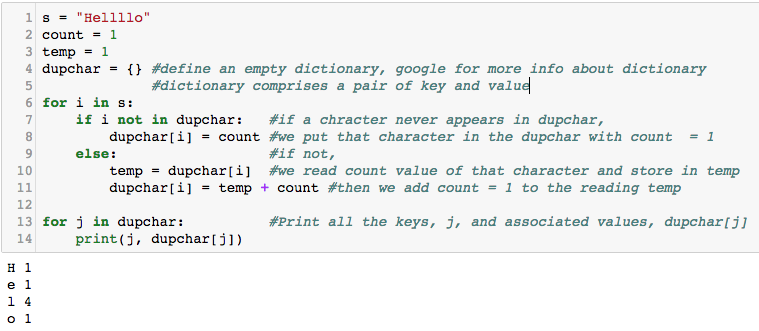
* Write a Python code that requests five integer values (on the same input line using .split( )) from the user. It then prints one of two things: if any of the values entered are duplicated, it prints "DUPLICATED"; otherwise, it prints "ALL UNIQUE".

6) Based on exercise 5, write a Python code that takes a word as an input and checks if there are duplicated characters. The expected outputs are:

../../../../Desktop/Screen%20Shot%202018-02-18%20at%206.54.13%

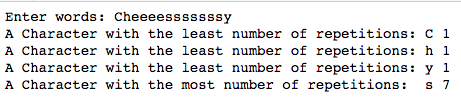
../../../../Desktop/Screen%20Shot%202018-02-18%20at%206.54.36%

The following Python code example illustrates how to count the number of duplicated characters.



|  |
| --- |
| **If you carefully observe the codes in exercises 5 and 6, you can see that there are now two approaches that can be used to determine the duplicated characters. In exercise 5, nested loop is used to run through all characters to see if there is duplicated character. In exercise 6, we have created a dictionary that we will use to store all characters that have been found and we need just one for loop. In exercise 6, at line 9 (else), we know that the character also exists. You can print a statement and break from the loop as done in exercise 5.** |

7) Write a Python code to take a word as an input and checks for duplicated characters and their associated repetitions. Print duplicated character(s) that has the minimum and maximum number of repetitions. For example,



|  |
| --- |
| Introduction to Functions in Python |

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

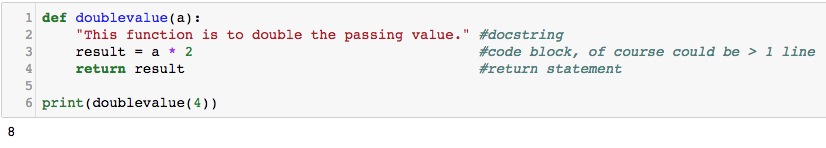
As you already know, Python gives you many built-in functions like print(), input(), min(), max(), etc. but you can also create your own functions. These functions are called *user-defined functions.*

## Defining a Function

You can define functions to provide the required functionality. Here are simple rules to define a function in Python.

* Function blocks begin with the keyword **def** followed by the function name and parentheses “( )”.
* Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
* The first statement of a function can be an optional statement - the documentation string of the function or *docstring*.
* The code block within every function starts with a colon (:) and is indented.
* The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

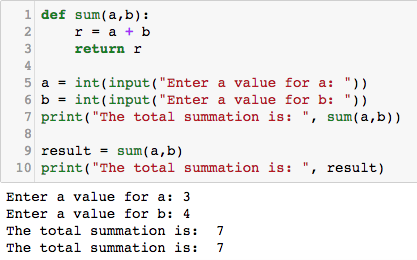
For example,



Calling a Function

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code.

Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python notebook. Following is the example to call sum() function −

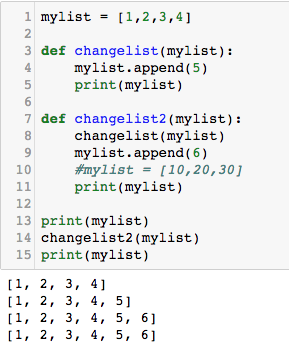


You can call a function directly by a function name as shown in code line# 7. Another possibility is to have a variable to take the value that is to be returned from the function as shown in code line# 9.

8) Write a Python function to take three inputs and return the average value.

## Pass by reference vs value

All parameters (arguments) in the Python language are passed by reference. It means if you change what a parameter refers to within a function, the change also reflects back in the calling function. For example −

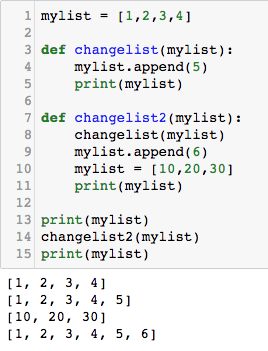


There are two functions, namely chagelist() and changelist2(). The first one appends 5 to the list and the second one calls the first function and appends 6 to the list.

Observe carefully!!! At the beginning, mylist contains 4 numbers which are 1,2,3 and 4. Then, at line# 13, we call print(mylist), which prints [1,2,3,4] on the screen. After that we call changelist2() and we pass mylist as an argument to the function. Hence, mylist is now passed by reference. Whatever you have made changes to mylist will reflect back. Changelist2(mylist) firstly calls changelist(mylist) at line# 8. Again, mylist is passed by reference inside changelist().

In chagelist(), we append 5 to the list and print(mylist). That is why you get [1,2,3,4,5] for the second output. Back to changelist2(), at line# 9, we append 6 to the list and print(mylist). That is the third output which is [1,2,3,4,5,6]. Finally, you call print(mylist) again at line# 15, which is the list output on the screen.

However, if you have created a new object (a new variable) inside a function, although the variable name is the same as an argument that has been passed to the function. Changes made to that object will not reflect back to the argument that has been passed to the function. For example,

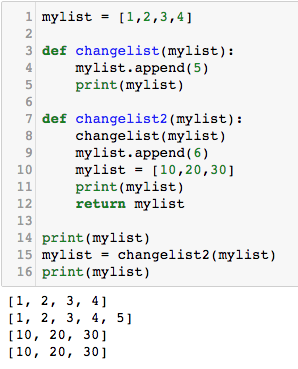


The above code is similar to the previous, except line# 10 where we have created a new object (parameter) namely, mylist (the same as mylist that has been passed into the function.

If you follow the code carefully, the third output now is different from what is shown in the previous example. The output is the value from mylist which is created inside changelist2(). This mylist is local to changelist2() which will note reflect back to mylist at line#1.

Hence, print(mylist) at line# 11 and 15 now give you different output.

What to do if you want a newly created object inside the function to reflect back to the variable outside the function? The answer is that you need to return that object out as shown in the following example:



The difference is at line#15. In the previous example, we simply call changelist2(mylist). We pass mylist by reference. Whatever happened to mylist inside the function will reflect back.

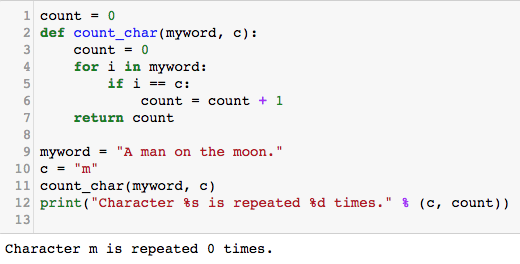
But at this line#15, we have mylist to take the return value from changelist2(mylist). The returned value is from a local mylist which contains 10,20,30.

That is why you get the different fourth output when compared to the previous example.

9) The following Python code is expected to count the number of a character “m”, however the output is not as expected. Why does the value of count is equal to 0?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What should be fixed to get the correct output?



10) Convert exercise 7 to a function.

1) Write a function

2) Take a word as an input to a variable *myword*.

2) Call a function and pass *myword* as an argument. The function shall do its intended job and return the dictionary *dupchar.*

3) Print characters with min and max no. of repetitions.