Python Bootcamp {2019}

EXERCISES WEEK 2

For each exercises, you will create a function that return a value. Also, before returning the value, you will have to print it inside your function and you might have some extra value to print for some of the exercises. The program name and the function name will be different. For example: the first program will be named (21_fun_calc.py but the function you have to write will be named fun_calc) You have to check the output every time. Finally, the output will show you how the function must be called.
Example: fun_calc(5, 15)

EXERCISE 21

Description :

> In this program you will create a function that take 2 parameters in value (A and B) and return the total value. It will print the result and also the calculation.

Requirements :
- Program must be named : **21_fun_calc.py** and saved into **week02/ex** folder

Hint :
- ❖ function
- ❖ return

Output :

```
fun_calc(5, 15)
>> 20
>> 5 + 15 = 20
```

Python Bootcamp {2019}

EXERCISE 22

Description :

In this program you will create a function take a string and return a
list. The function must split the string at every space character. If
the string is empty you will return an empty list.

Requirements :
- Program must be named : **22_fun_split.py** and saved into **week02/ex** folder

Hint :
❖ Function
❖ list
❖ split

Output :

```
fun_split("Hello! It's me again!")
>> ['Hello!', 'It's', 'me', 'again!']
```

EXERCISE 23

Description :

You will write a function that take 2 parameters and return a Tuple.

Requirements :
- Program must be named : **23_fun_tuple.py** and saved into **week02/ex** folder

Hint :
❖ function
❖ tuple

Python Bootcamp {2019}

Output :

```
fun_tuple('abc', 123)
>> ('abc', 123)
```

EXERCISE 24

Description :

    You will write a function that take a Tuple in parameters and return
    the first value

Requirements :
  ● Program must be named : **24_tuple_first.py** and saved into **week02/ex**
    folder

Hint :
  ❖ function
  ❖ tuple

Output :

```
tuple_first(('abc', 123))
>> abc
```

EXERCISE 25

Description :

    You will write a function that take a Tuple in parameters and return
    the second value

Requirements :
  ● Program must be named : **25_tuple_second.py** and saved into **week02/ex**
    folder

Python Bootcamp {2019}

Hint :
   ❖ function
   ❖ tuple

Output :

tuple_second(('abc', 123))
>> 123

EXERCISE 26

Description :

   You will write a function that take a list in parameters and return a
   List with unique values only.

Requirements :
   ● Program must be named : **26_list_set.py** and saved into **week02/ex** folder

Hint :
   ❖ function
   ❖ list
   ❖ set

Output :

list_set(['456', '123', '789', '123', 'abc', 'abc', 'def'])
>> ['456', '123', '789', 'abc', 'def']

EXERCISE 27

Description :

   You will write a function that take a list in parameters and return a
   sorted list.

Python Bootcamp {2019}

Requirements :
- Program must be named : **27_list_sort.py** and saved into **week02/ex** folder

Hint :
- ❖ function
- ❖ list
- ❖ sort

Output :

```
list_sort([4, 2, 19, 50, 49, 48, 1, 2, 3, 4, 5])
>> [1, 2, 2, 3, 4, 4, 5, 19, 48, 49, 50]
```

EXERCISE 28

Description :

You will write a function that take a list in parameters and return a
sorted list with unique values only.

Requirements :
- Program must be named: **28_list_sort_set.py** and saved into **week02/ex**
folder

Hint :
- ❖ function
- ❖ list
- ❖ sort

Output :

```
list_sort_set([4, 2, 19, 50, 49, 48, 1, 2, 3, 4, 5])
>> [1, 2, 3, 4, 5, 19, 48, 49, 50]
```

Python Bootcamp {2019}

EXERCISE 29

Description :

   You will write a function that take a list in parameters and return a
   sorted list with unique values only with non numeric values.

Requirements :
   ● Program must be named : **29_list_sort_str.py** and saved into **week02/ex**
      folder

Hint :
   ❖ function
   ❖ list
   ❖ sort

Output :

```
list_sort_str(["abc", "4", "2", "3", "dza", "def"])
>> ['abc', 'def', 'dza']
```

EXERCISE 30

Description :

   You will write a function that take a List in parameters and return a
   sorted List with unique values only that are valid numbers. also the
   returned list elements must be INTEGER (not string)

Requirements :

   ● Program must be named : **30_list_sort_int.py** and saved into **week02/ex**
      folder

Hint :
- ❖ function
- ❖ list
- ❖ sort

Output :

```
list_sort_int(["abc", "4", "4", "4", "4", "2", "3", "dza", "def"])
>> [2, 3, 4]
```

EXERCISE 31

Description :

You will write a function that take a list in parameters and return a dictionary with the element of the list as key, and the number of occurrences as value.

Requirements :
- Program must be named : **31_dict_count.py** and saved into **week02/ex** folder

Hint :
- ❖ function
- ❖ list
- ❖ dictionary
- ❖ occurrences

Output :

```
dict_count(["a, "a", "a", "b", "c", "d", "c", "b", "c", "d", "c", "e", "e", "e"])
>> {a: 3, b: 2, c: 4, d: 2, e: 3}
```

EXERCISE 32

Description :

> You will write a function that take a List in parameters and return a list of tuples with the element of the List as Key, and the number of occurrences as value. **The list output must be sorted by value**. Then the function will display the sum of all the items. If the list is empty you will return an empty list and display **"Your string is empty."**. Before returning the value you will also print the TOTAL.

Requirements :
- Program must be named : **32_dict_count2.py** and saved into **week02/ex** folder

Hint :
- ❖ Function
- ❖ list
- ❖ dictionary
- ❖ Occurrences
- ❖ values

Output :

dict_count2(["z, "z", "z", "z", "b", "b", "b", "b", "a", "a", "a", "a", "a", "a"])

>> [('a' 6), ('b' 4), ('z', 4)]
>> TOTAL: 14

EXERCISE 33

Description :

You will write a function that take a List in parameters and return a
list of tuples with the element of the List as Key, and the number of
occurrences as value. **The list output must be sorted by key**. Then the
function will display the sum of all the items. If the list is empty
you will return an empty list and display **"Your string is empty.".**
Before returning the value you will also print the TOTAL.

Requirements :
● Program must be named : **33_dict_count3.py** and saved into **week02/ex**
   folder

Hint :
❖ Function
❖ list
❖ dictionary
❖ Occurrences
❖ values

Output :

dict_count3(["a", "b", "b", "c", "c", "c", "c", "d", "d", "e", "e", "e"])
>> [('a', 1), ('b', 2), ('c', 4), ('d', 2), ('e', 3)]
>> TOTAL: 12

EXERCISE 34

Description :

You will write a function that return the current date with the
following format: YYYY-MM-DD. The return value must be a string.

Python Bootcamp {2019}

Requirements :
- Program must be named : **34_current_date.py** and saved into **week02/ex** folder

Hint :
- ❖ function
- ❖ datetime

Output :

current_date()
>> 2021-06-14

EXERCISE 35

Description :

You will write a function that return the current time with the following format: hh:mm:ss
The return value must be a string.

Requirements :
- Program must be named : **35_current_time.py** and saved into **week02/ex** folder

Hint :
- ❖ function
- ❖ datetime

Output :

current_time()
>> 04:59:40

Python Bootcamp {2019}

EXERCISE 36

Description :

> You will write a function that return the current date and time with the following format: YYYY-MM-DD hh:mm:ss
> The return value must be a string.

Requirements :
- Program must be named : **36_date_time.py** and saved into **week02/ex** folder

Hint :
- ❖ function
- ❖ datetime

Output :

```
date_time()
>> 2021-06-14 04:59:40
```

EXERCISE 37

Description :

> You will write a function that take a timestamp string and convert to readable date and time with the following format: YYYY-MM-DD hh:mm:ss
> If the timestamp is not valid, you function will return 0 and display **"Your timestamp is not valid."**

Requirements :
- Program must be named : **37_timestamp_to_str.py** and saved into **week02/ex** folder

Hint :
- ❖ function
- ❖ datetime
- ❖ timestamp

Python Bootcamp {2019}

Output :

```
timestamp_to_str(1623646780)
>> 2021-06-14 04:59:40

timestamp_to_str("abc")
>> Your timestamp is not valid
```

You can find timestamp on this link: https://www.epochconverter.com/

EXERCISE 38

Description :
    You will write a function that take an integer as parameter that represent a number of second and return a list of time.
    For example: if the current time is 04:30:12 and the integer = 3
    ==> [04:30:12, 04:30:13, 04:30:14]
    If the integer is negative or not valid, you will return an empty list and display **"Invalid integer."**

Requirements :
- Program must be named : **38_time_list.py** and saved into **week02/ex** folder

Hint :
- ❖ loop
- ❖ list
- ❖ function
- ❖ datetime
- ❖ sleep

Output :

```
time_list(5)
>> ['04:30:12', '04:30:13', '04:30:14', '04:30:15', '04:30:16']

time_list(-2)
>> []
>> Invalid integer.
```

Python Bootcamp {2019}

EXERCISE 39

Description :
        You will write a function that take a list and return a list with
        uppercase format. If the list is empty, your function will return an
        empty list.

Requirements :
   ● Program must be named : **39_acronym.py** and saved into **week02/ex** folder

Hint :
   ❖ list
   ❖ string
   ❖ loop

Output :

acronym(["world", "wide", "web"])
>> ['W', 'W', 'W']

acronym([])
>> []

EXERCISE 40

Description :
        You will write a function that take a list of string and return the
        number of vowels. Also it will display the number of vowels, then
        display a concatenate string with vowels only in lowercase. Finally
        display a concatenate string with all the characters that are not
        vowels in uppercase. If not vowels are given, you function will return
        0 and print "NO VOWELS"

Requirements :
   ● Program must be named : **40_vowels.py** and saved into **week02/ex** folder

Output :

```
vowels("what is that ?")
3
aia
WHTSTHT?
```

⚠ Warning ⚠

MAKE SURE THAT ALL YOUR PROJECTS AND EXERCISES ARE WELL TESTED AND YOU DID NOT FORGET EVEN A SINGLE CHARACTER. MAKE SURE THAT YOUR FILENAMES ARE CORRECT. MAKE SURE THAT YOUR PROJECTS AND EXERCISES RESPECTS ALL THE REQUIREMENTS.