Python programming exercises

Part 3

1. Simple functions

1.1. Hello World

- Create simple_functions.py file
- Write a function that prints “Hello World”: `greet()`
- Call this function several times from your script

1.2. Arithmetic operations

- Create another function that takes two numbers as parameters and prints sum of those numbers: `add(x, y)`
- Create similar functions for
    - Subtraction: `sub(x, y)`
    - Multiplication: `mul(x, y)`
    - Division: `divi(x, y)`

1.3. Return value

- Now, instead of printing results, your functions shall return result (using return statement)
- Call your functions several time from the script, assign returned value to a variable and print it

2. Calculator

2.1. Calling functions from a function

- Continue working with simple_functions.py file.
- Create a new function which takes 3 parameters: 2 numbers and an operator (operator can be one of the following: `‘+’, ‘-’, ‘*’, ‘/’`)
- Based on the operator, perform arithmetic operation on the numbers and return the result.
    - If operator is equal to ‘+’ then perform addition
    - If operator is equal to ‘-’ then perform subtraction
    - If operator is equal to ‘*’ then perform multiplication
    - If operator is equal to ‘/’ then perform division
    - If operator is something else - print a message “Unknown operator” and return `None`
    - Use four functions which you implemented in the previous exercise

3. Functions with default parameters

3.1. Hello World, revisited

- Create default.py file

- Write (or copy) “Hello World” function from the first exercise
- Modify this function, so it takes a parameter `name` and prints the name instead of “`World`”
- If user doesn’t specify parameter `name` then function shall print “`Hello World`”. Examples:

  `greet(“John”)` prints `“Hello John”`

  `greet()` prints `“Hello World”`

3.2. List of pets

- Write a function that takes 2 parameters: a pet (i.e. ‘cat’, dog’) and a list
- Function shall add this pet to the list and return this list
- If user doesn’t specify pet parameter, then function shall add ‘cat’
- If user doesn’t specify list parameter then function shall create a new list

4. Various exercises

4.1. Unique values in the list

- Create list_functions.py file
- Write a function that takes a list and returns a new list containing unique elements from the first list

4.2. Even numbers

- Write a function that takes a list of numbers and prints only even numbers from that list

4.3. Word frequency

- Open sherlock.py file. Inside, you’ll find a short part of one of the stories about Sherlock Holmes.
- Write a function that takes one parameter (text) and calculates how many times each word appears in the text:
  - First you need to split text into list of words
  - Then you need to calculate how many times each word appears in the list (hint: use a dictionary type)