Exercises Python Advanced

# Turtle

Draw a polygon with turtle.

* Check out the different methods that you can use with turtle
  + in particular: forward and left
* Import turtle
* End the program with turtle.done()
* Draw a square
* Draw a polygon
  + Calculate the angle of each corner <https://nl.wikipedia.org/wiki/Regelmatige_veelhoek>

# BankAccount

Create a Bankaccount class, create serveral Bankaccount objects and demonstrate that you can deposit and withdraw amount to the account.

Tips:

* Create a class Bankaccount
* Add attributes in the **\_\_init\_\_** method. Attributes should be \_\_balance and \_\_holder.
* Add the methods: **deposit** and **withdraw** that take an amount argument and a third method **info** that returns information about the account.

Instantiate serveral bankaccount objects and demonstrate the working of the class

# Class Car

* Create a class named **Car**
* Add the \_\_init\_\_ method and set several attributes like \_**make**, \_**type** and \_**color**
* Set the \_**mileage** attribute to 0
* Create a method **info** that describes the car and the mileage
* Create a method **drive** that takes an amount of kilometers and adds that to the mileage.

Test you class by instantiating a car and calling the methods

# Vector Class

Create a 2d-Vector class. Also add operator overloading for the + sign to add two vectors together.

Tips:

* Build a class called Vector
* Add two attributes: x and y
* Implement the \_\_init\_\_ method that takes two arguments: x and y
* Implement the \_\_str\_\_ and \_\_repr\_\_ methods.
* Implement the \_\_add\_\_ method the define the adding of two vectors.
* Test your class by creating two vectors and adding these together.

# sys library

* Get the current version of Python
* Return the message you are currently running Python version …

# os library

* Use the os library to get the contents of a directory in a list.

# datetime library

* input a date and print the date in another format

# re library

* Go to the website [https://rubular.com](https://rubular.com/)
* Build and test a regular expression to match and e-mail address
* Use the same regular expression in python with the search method in the re library

# pickle library

* Create a datastructure and store this in a pickle file. Create a second python script that reads pickle file and restores the data in the data structure.

# xml library

Read and parse the Macbeth xml file and generate several overviews.

* The name of the play
* The names of all the personas
* The names of all the scenes

# statistics library

Create a function that calculates and returns the mean, median and mode of a list of numbers.

Tips:

* Define a function as **def central\_measures(numbers)**
* Calculate the measures:
  + The **mean** is the sum of the values divided by the number of values
  + The **median** is middle value of the sorted list of values
  + The **mode** is the most frequently occuring value
* Return the measures as a tuple with **return mean, median, mode**
* Call the function with a list of arbitrary numbers
* Print the result

# doctest library

* Create a function and add a docstring with doctests for the function.

# requests library

Use requests to query openweathermap.org for the weather in a specified city.

Tips:

* import **requests**
* build the url (see <https://openweathermap.org/current>)  
   use: **appid=d1526a9039658a6f76950cff21823aff**
* use the following code to get the response:   
   **response = requests.get(url)**
* use json to decode the response into a Python dictionary **response.json()**
* get and print the temperature