

Test.

Note: Run and test all your code before submitting. If your code outputs unexpected results or does not compile, it means that something is not right in the code and you need to fix it.

Tasks:

1. Using `time.time()`, measure the time it takes Python to execute the following.

Using list comprehension create a list of squares of values from -20, to 20. Your list should look like this:

```
[400, 361 ... 4, 1, 0, 1,4 ... 361, 400]
```

Use `matplotlib.pyplot.plot` to plot this data.

Use `time.time()` again to find out how much time it took to execute these commands.

2. Repeat the same but this time use numpy array. Compare the time measures. Which way is faster?

3. Using `numpy.random.randint()` create an array with 20 random value in a range from 0 to 100. Write a filter statement with lambda function, to output only the values that are odd.

4. Write a money conversion function. The function should take the following parameters: amount, conversion rate, original currency, converted currency. Set default values, so if the user specifies only amount your function presumes that the original currency is US Dollars, the converted currency is Euros, and conversion rate is 0.82. The function should return a message saying that how much money has the user converted. For example:

“You have converted 100 US Dollars to 82 Euros”

Note: don't forget that the user can specify other parameters for the original currency, converted currency, and rate. Make sure you properly adjust your function for these cases.

5. Create a class `Vehicle`, with instance variables: name, year, public. Where name is a brand of the Vehicle, year is the year the vehicle was made in. Public is a boolean value, that is True, if the vehicle is public (e.g bus) or False, if the vehicle is private (e.g. car). Create an instance method to describe the object. The method should output the values for the variables from the `__init__` in a well-formed sentence.

Note: you might want to use `self.__class__.__name__` in your method.

6. Create a child class `Bus` that inherits from `Vehicle` class. Keep all the values from the `__init__()` function. For the class `Bus` add instance variables `capacity` and `rate`, to store the number of people the bus can take and the rate with which this bus goes (e.g. every 15 minutes).

Create a method `is_full()` with an extra parameter number of people waiting, that is specified by the user. If the number of people waiting is greater than `capacity`, return True, else return False.

Create a method `next_bus()` with extra parameter `day`. The method should alert the user that the next bus will come in `self.rate` minutes or less.

If `day` is "Saturday" or "Sunday" double the rate.

7. Create a child class `Car`, that inherits from class `Vehicle`. Keep all the values from the `__init__()` function. Add two instance variables: `mileage` and `max_speed`.

Create a method `is_a_race_car()`. Output `yes`, if `max_speed` is greater than 150 mph and `mileage` is less than 100000 miles.

8. Take a Plain UTF-8 copy of *The Wonderful Wizard of Oz* by L. Frank Baum from: <https://www.gutenberg.org/ebooks/55>

Open the file and read it. Count the number of characters, words, and sentences. Write it down in a new file that you created.

9. Create a sorted frequency list, excluding the stopwords and converting text to a lower case.

10. Plot the results of a frequency list using `matplotlib.pyplot.plot`.