

# The one with electric cars

---

**Due** 20 Oct by 23:59      **Points** 3      **Submitting** an external tool  
**Available** until 20 Oct at 23:59

---

Successfully solving this problem will award the student **3 point** towards Examination 1

## Requirements

- The script must pass all the automatic tests
- The script must pass a manual inspection
- **The student must write the script on his or her own**, any suspicion about cheating, cooperation, and/or plagiarism will be reported to the Discipline Board (see the [Study Guide](https://hkr.instructure.com/courses/5190/files/1027789/download?download_frd=1) ↓ ([https://hkr.instructure.com/courses/5190/files/1027789/download?download\\_frd=1](https://hkr.instructure.com/courses/5190/files/1027789/download?download_frd=1)) for more information). The students script will be checked both manually and using anti-plagiarism software

## Instructions

1. Analyse the problem below using [Polya's Problem solving Technique](https://math.berkeley.edu/~gmelvin/polya.pdf) ↗ (<https://math.berkeley.edu/~gmelvin/polya.pdf>)
2. Create a script named **Lab2\_3.py** that solves the problem according to the given instructions
3. Pay special attention when formatting the output as it must match the specified format **exactly** (character by character) to pass the tests
4. Upload the script using the link below

## Task description

*“Why spend five minutes doing something when you can spend five hours automating it?”*  
(Unknown)

The quote above definitely fits this task, but as you really like programming and you want to show your friends and family what you have learned, you have decided to write a small script that can help you when considering what electric car to buy. After lengthy discussions with everyone that wanted to listen, you have come to the conclusion that to you, **the range you can drive without recharging the car is the most important.**

With this in mind, **you want your script to ask you how far you want to drive, then the script shall display a list of electric cars and a percentage showing how much of the cars maximum range the specified distance is.** As an example, imagine that you want to drive 150km, then one line from the output might be:

```
Renault Kangoo ZE          --> 75%
```

This car has a max range of 200km, so the specified distance of 150km will be 75% of the maximum distance we can travel with this car.

When running the script, a command line argument shall be given that specifies what file contains all the information about the electric cars. The script shall implement the following functions:

### **read\_lines(filename)**

This function shall read all the lines from a file, strip them of the new line character, and place them in a list that is returned. If the *filename* parameter is not a valid path, the function shall raise an `FileNotFoundError`.

### **parse\_cars(list\_of\_strings)**

This function shall take a list as a parameter, each element in the list shall be a string containing the name of the car model and the max range for that model. The name and range shall be separated by a colon. Here is an example: *Renault Kangoo ZE:200*. The function shall parse each line into a **tuple consisting of the name of the car model and the max range**. An example would be `('Renault Kangoo ZE', 200)`. Note that the max range shall be an integer number. All tuples shall be placed in a list that is returned from the function.

### **calculate\_percentage(distance, cars)**

The first parameter is the distance to travel and the second parameter is a list of tuples where each tuple consists of the name of the car model and the max range for that model. The function shall create a new list that is later returned. The new list shall contain tuples consisting of the name of the car model and a calculated percentage. The percentage shall represent how many percent of the max range the distance parameter is. Store the percentage as a `Float`. An example to clarify, if we call the function with the distance 150

and a list of cars containing the Renault used as example before, one of the new entries would be ('Renault Kangoo ZE', 75.0).

### **display\_result(percentages)**

Takes as a parameter a list of tuples where each tuple contains the name of the car model and a percentage (as a Float). The function shall display a header and a list on the screen. The percentage shall not have any decimals (formatted using .0f). It is important that the format is **exactly** as follows:

```
To drive the specified distance would correspond to this many
percent of each cars specified max range.
Tesla Model S Long Range          --> 90%
Tesla Model S Performance         --> 93%
Volkswagen ID.3 77 kWh Tour       --> 96%
Tesla Model 3 LR 4WD              --> 98%
Tesla Model Y LR 2WD              --> Distance exceeds max range (102%)
```

Note that if the distance to travel is larger than the max range of the car model, this shall be indicated in the list.

### **main()**



This function shall ask the user for the distance to travel, and use the other functions as needed to read the file specified by the command line argument, parse the content, and display the list with percentages. This function shall also handle the *FileNotFoundError* raised by the **read\_lines** function. Should an error occur, show the error message:



```
An error occurred while trying to read the file.
```

To avoid problems with the testing framework, and to build a good habit, call the main function using the following construct:

```
if __name__ == '__main__':
    main()
```

## Comments and additional resources

- Students are highly recommended to first solve all the tasks from the corresponding lectures and exercises (including the extra tasks) before attempting to solve this problem. Having a solid plan before starting to code will also help
- The student can upload as many times as he or she likes to before the deadline
- The student is recommended to not use any rounding during the calculation of the percentages
- This [file](https://gist.github.com/andreas-hkr/259afc843fa474193f65ac1b8c26f850)  (<https://gist.github.com/andreas-hkr/259afc843fa474193f65ac1b8c26f850>) that contain some cars can be used to try the script out
- Watch [this](https://youtu.be/IOeIDvyRUQs)  (<https://youtu.be/IOeIDvyRUQs>) video if you want to learn more about why we call main the way we do

- This [video](https://youtu.be/o1Rxrtiqml8)  [\\_\(https://youtu.be/o1Rxrtiqml8\)\\_](https://youtu.be/o1Rxrtiqml8) covering how to use command line arguments might be useful
- This [video](https://youtu.be/oDCNZa7MJqA)  [\\_\(https://youtu.be/oDCNZa7MJqA\)\\_](https://youtu.be/oDCNZa7MJqA) recorded while running the finished script showcases how the script shall look and work when finished

This tool needs to be loaded in a new browser window

Load The one with electric cars in a new window