University of St. Gallen

Institute of Computer Science

Fundamentals of Computer Science (3,125), HS2019
T. Huber, C. Niklaus, B. Bermeitinger, J. Sales, S. Handschuh
thomas.huber@unisg.ch

Assignment 6: Data Science: Basics

Deadline: Nov 19, 2019; 12:00 CET

Introduction In this assignment you will learn the basic tools of the trade for *Data Science*. You will learn how to use list comprehensions, which allow you to modify lists with less code than by using *for* loops and how to use anonymous functions, called *lambda functions*[1]. You will learn the basics of functional programming in Python by using *map*[2] and *reduce*[3]. For the second half of the assignment you will learn how store data in a more structured way by getting to know *pandas DataFrames*[4]. After that you will learn how to visualize your data with *matplotlib*[5].

- Part 1: List comprehension and functional programming (Tasks 1-2)
- Part 2: DataFrames (Tasks 3-4)
- Part 3: Visualization (Task 5)

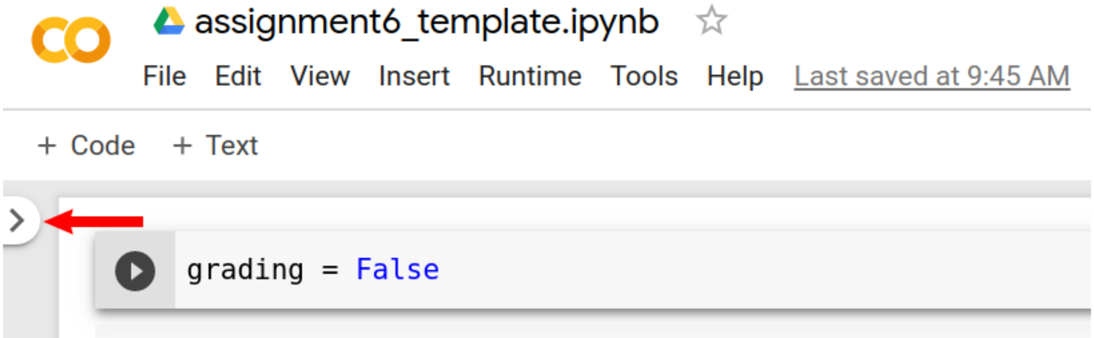
Rationale Data Science is useful to everyone. You will be able to apply your new skills to more easily analyze data and visualize it.

Jupyter Notebooks For this and the next two assignments you will be using *Google Colab*[6] instead of Thonny. Google Colab is a Jupyter Notebook environment that you can access in your browser. You can create *cells* that let you run only certain parts of your code. You can also get output for those specific cells, making it much easier to understand what your code does.

Prerequisites Create a Google account if you do not already have. Then upload the notebook for this assignment on Google Colab: `https://colab.research.google.com/`.

- Click `File` in the top-right menu
- Click `Upload notebook...`
- In the menu select the template file from the .zip file that you downloaded from Canvas `assignment6_template.ipynb`
- Click the arrow on the right and upload the other files from the .zip file in the `Files` tab

[1] https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions
[2] https://docs.python.org/3/library/functions.html#map
[3] https://docs.python.org/3/library/functools.html#functools.reduce
[4] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html
[5] https://matplotlib.org/
[6] https://colab.research.google.com



```
grading = False
```

Figure 1: How to open the sidebar with the `Files` tab

For this assignment, you **ONLY** need to edit the notebook!
To run your code, click into one of the cells (the blocks of text that you can edit) and then click the *Run* button on the side of the cell.
When the code in a cell depends on a previous cell (for example one cell has a class *Villain*) then you need to run the previous cell before running the cell that depends on it! This also applies when you change code in a cell, you need to run that cell again and also any cells that depend on it (for example when they use a function defined in that cell).
You can run multiple cells by using the `Runtime` menu at the top.

1 List Comprehension (1 Point)

In this task you will learn how to use list comprehensions.

(a) Edit the `Villain` class in such a way that a `Villain` object can be interpreted as as a *boolean value*. Use the `captured_villains` function to test your implementation.

(b) Edit the `Villain` class in such a way that it implements the six rich comparison methods[7][8]. Use the `villains_by_rank` function to test your implementation.

2 List Comprehension and Functional Programming (1.5 Points)

In this task you will encounter more advanced list comprehensions, basic functional programming with `reduce` and *lambda functions*.
Implement all methods in this function with one line of code!
Do NOT create helper functions in other cells and call them. If you feel like you need an additional function, use a *lambda function* instead of defining a function.

(a) Implement the `clean` function. Use the cell below it to test your implementation.

(b) Implement the `check_todo` function. Use the cell below it to test your implementation.

(c) Implement the `check_bills` function. Use the cell below it to test your implementation.

3 DataFrame: Basics (2.5 Points)

For this task you need to convert text data into a set of coordinates. To structure your data you will use a *pandas DataFrame*.

(a) Implement the `convert2float` function. Use the cell below it to test your implementation.

[7] https://docs.python.org/3/reference/datamodel.html#special-method-names
[8] https://docs.python.org/3/reference/datamodel.html#object.__lt__

(b) Implement the `load_coordinates` function. Use the cell below it to test your implementation.

④ DataFrame: Advanced (2 Points)

For this task you must modify the DataFrame from task 3b). If you were unable to solve task 3b) you can load the correct data like this:

```
correct_3b = pd.read_csv("3b_data.csv")
```

There are cells provided that do this for you. If your solution is correct you can also use the return value of your `load_coordinates` function.
You do NOT lose points if you use the provided solution from 3b) to solve task 4!

- Implement the `fix_symbol` function. Use the cells below to draw an image of the coordinates. There are two cells: One that draws the image based on the corrupted data, and one that draws the image after calling your `fix_symbol` method. If your solution is correct the second image will show you a certain symbol.

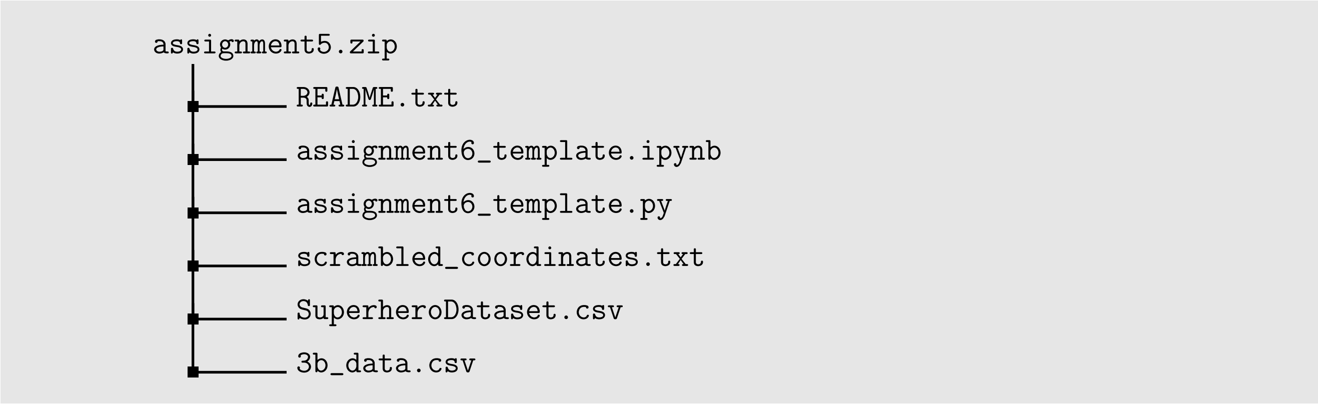
⑤ Visualization with Matplotlib and pandas (3 Points)

- Implement the `superhero_data` function. You can use the the cell below to test your implementation.

Hand-in Instructions Create a file `README.txt` that includes 1.) answers to the questions, 2.) any reference that you used to complete this assignment, 3.) pitfalls you encountered and 4.) short explanations of your solution if necessary. Fill in your name and student ID in the comment header of files you edited.

Download your notebook by clicking `File` at the top and then select `Download .ipynb` and `Download .py`. Include both of these files in your `zip` file.

Compress the whole folder with the Python files and the `README.txt` to a `zip` file named "`assignment6.zip`". Upload the zip file to your exercise group's Course page on Canvas. See Figure.2 for the list of files that should be included in the zip file.



```
assignment5.zip
    README.txt
    assignment6_template.ipynb
    assignment6_template.py
    scrambled_coordinates.txt
    SuperheroDataset.csv
    3b_data.csv
```

Figure 2: The required files for submission.