

Basic Programming - Seminar 5

Alexandra Ciobotaru

17 December 2021

The scope of this seminar is to get familiar with reading and writing files in Python, as well as manipulating dataframes and plotting. Also, to get to practice functions and loops again.

Exercise 1

Write a Python program to convert `j_str` Python dictionary object to JSON data. Print the object members with indent level 4 and sorted indexes, and save the result to a json file.

```
j_str = {'4': 5, '6': 7, '1': 3, '2': 4}
```

Hint: you can sort keys using `sort_keys=True` and indent the json using `indent=4` like in this example: `print(json.dumps(j_str, sort_keys=True, indent=4))`

Exercise 2

You have `sampleJSON` JSON string. Load it into a JSON object using `json.loads(sampleJSON)`, and put in a list all the vales of the key 'name'.

```
sampleJson = """[
    {
        "id":1,
        "name":"name1",
        "color":[
            "red",
            "green"
        ]
    },
    {
        "id":2,
        "name":"name2",
        "color":[
            "pink",
```

```

        "yellow"
    ]
}
]"""

```

Hint: after you load sampleJSON into a JSON object, create a list accumulator, and for each item in your JSON object, put in the accumulator the value existing in key "name", using `item.get('name')`.

Exercise 3

Write a python function that takes as an input a filename, opens the filename and then counts and displays the total number of words in the text file .

Remember how to write a Python function:

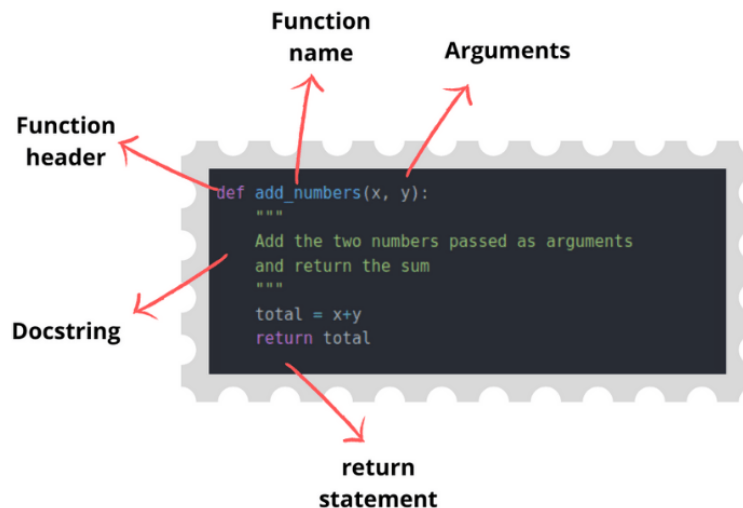


Figure 1: A function in Python

Recommendations:

- Name your function "count_words";
- your function should have only one input argument, filename;
- inside the function: open the filename in reading mode
(`file = open(filename, "r")`);
- read its contents using the read method and put it into a variable called data;
- think of a method to split data into words and generate a list of words (did you think about the `split()` method?);

- create an accumulator to count words called `count` and increment it in a simple for loop or by using list comprehension;
- put inside the function: `print("Number of words: ", count)`
- don't forget to close the file, if you didn't use the `with open` statement.

Test your function by calling it on a text file you create separately.

Modify previous function so that it returns only the `count` number. Write a docstring that explains the argument of the function and what it returns.

Further, call your function in a print statement.

Exercise 4

Write a function that counts the words 'this' and 'these' present in the text file created earlier.

Recommendations:

- Name your function "count_special_words";
- your function should have only one input argument, `filename`;
- inside the function: open the `filename` in reading mode
(`file = open(filename, "r")`);
- read its contents using the `read` method and put it into a variable called `data`;
- think of a method to split `data` into words and generate a list of words (did you think about the `split()` method?);
- create an accumulator to count words called `count` and increment it in a simple for loop or by using list comprehension, if the word is 'this' or 'these';
- don't forget to close the file, if you didn't use the `with open` statement;
- return the `count` value.

Exercise 5

Download the young people survey dataset from: <https://www.kaggle.com/miroslavsabo/young-people-survey/>. Open `responses.csv` with `.read_csv` method and analyze the dataframe:

- how many rows and columns does it have? (use `df.shape`);
- create a list with all the columns the dataset has;
- print it's head (first 5 rows).

Let's find out what music enjoys the youngest people who completed the survey.

- Select column Age and print it;
- find out the maximum and minimum age;
- select only the rows with minimum value in "Age" column and put those in a new dataframe (`new_df = df[df["Age"] == 15]`);
- from this dataframe select only these columns: ['Slow songs or fast songs', 'Dance', 'Folk', 'Country', 'Classical music', 'Pop', 'Rock', 'Metal or Hardrock', 'Punk', 'Hiphop, Rap', 'Reggae, Ska', 'Swing, Jazz', 'Rock n roll', 'Alternative', 'Latino', 'Techno, Trance'] representing data regarding their music preferences (5 means they enjoy that style a lot, 1 means they do not) and put this data into another dataframe;
- delete rows containing NaN values using `.dropna()` method;
- compute mean using `texttt.mean()` method on the created dataframe;
- What is the biggest value? What genre wins the hearts of 15 years old's?
- Now do the same for 30 years old youngsters. What genre is the winner?

Exercise 6

Write a script to create and display a DataFrame from the specified dictionary data with the specified index labels.

```
exam_data = {'name': ['Anastasia', 'Dima', 'Katherine',
                     'James', 'Emily', 'Michael',
                     'Matthew', 'Laura', 'Kevin',
                     'Jonas'],
             'score': [12.5, 9, 16.5, np.nan, 9, 20,
                      14.5, np.nan, 8, 19],
             'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
             'qualify': ['yes', 'no', 'yes',
                        'no', 'no', 'yes', 'yes',
                        'no', 'no', 'yes']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Next, remove rows with empty values and plot a bar chart that shows the name of the students on the horizontal axis and the scores they got on the vertical axis.

Hint: the bar plotting formula is `df.plot.bar(x = 'name', y = 'score')`. Can you modify this formula to sort values in 'score', using `sort_values` method?

Write the dataframe to a csv file.

Congratulations, you have finished your fifth seminar!