CODELABS

Author:

Kevin Ochieng'| linglabafrica@gmail.com [startup]

What are Codelabs

Codelabs are a type of interactive and hands-on learning resource designed to help individuals learn how to program, develop software, or work with specific technologies or tools. They are commonly used in the field of computer science and software development as a practical way to teach concepts, skills, and best practices.

Grooking Python 101 - Solve the Problems described below.

Data Source:

- Test Files

Libraries:

- Pandas
- Numpy
- LaBSE - stretch
- Tensorflow - requires a python downgrade
- Google API - for cloud storage - last step of the solution

Questions:

- [ ] Create a new environment and import the Test Files into the Project. You will need to download the file as a .xlsx file
- [ ] Generate the email addresses of all the students in the format below.
    - [ ] Student Name: `John, Doe Hussein` - Email address: `jhussein@gmail.com`
    - [ ] Points to Note
        - [ ] Email addresses should be unique

- [ ] If the student has two names then you need to use what is available
- [ ] Email addresses should not have special characters

- [ ] This project should be pushed into GitHub for collaboration
  - [ ] This project must have a proper readme file
  - [ ] All output files must be labeled properly
    - [ ] `Files with all functions - functions.py`
    - [ ] `Files with constraints - constraints.py`
    - [ ] `Main program file - main.py`
  - [ ] Use log files to save all the computations
- [ ] Save the files as TSV and CSV respectively
  - [ ] For all information formatting
- [ ] Generate separate lists of Male and Female students
  - [ ] In the logs, you need to show the number of all male students and female students
  - [ ] List the names of students with special characters for example `Orony, Charis Ng'wono - regex solution`
- [ ] It is common that some names are similar, using LaBSE, run a similarity matrix on all male names vs female names and show the results in a json file. Pick the results that have at least 50% similarity. - stretch
- [ ] Merge all the documents into one file then:
  - [ ] Using Pandas, run a one time shuffle of the names and save the file as a json file.
  - [ ] Save another copy as a Jsonl file with the format below.

```
{
  "id": "0",
  "student_number": "123449",
  "additional_details": [
    {
      "dob": "2000-2-27",
      "gender": "m",
      "special_character": "['yes']",
      "name_similar": "['no']"
    }
  ]
}
```


- [ ] Generate your Google API key and save all your files in Google Drive for backup