2021

Makhalfeh, Osaid

AXSOS AG

11/9/2021

Students Shuffling Tool



SOFTWARE MANUAL

Contents

[About the SST: 0](#_Toc87369232)

[System requirements 0](#_Toc87369233)

[Usage 1](#_Toc87369234)

[launching the tool 1](#_Toc87369235)

[read custom columns from excel file 3](#_Toc87369236)

[change generated column names in the excel file 3](#_Toc87369237)

[change output file name 3](#_Toc87369238)

[Change the shuffling type: 4](#_Toc87369239)

[Table of commands 5](#_Toc87369240)

# About the SST:

Students Shuffling Tool

    A tool made to read an excel file and generate an excel file with multiple sheets based on the user input where the user can choose the excel file to read from, number of sections, number of stacks that he wants the shuffling for.

   The tool reads and write excel files with extension of "**xlsx**" and does not support the old "**xls**" version.

# System requirements

Please ensure that your computer meets or exceeds the following system requirements before installing the **SST** application.

|  |  |
| --- | --- |
| Available disk space 200 MB | Available disk space 1 MB |
| Operating system | Operating system Windows, Linux, Mac |
| Environment Variables | Python v3.5+.  Pip list:- et-xmlfile==1.1.0  numpy==1.19.5  openpyxl==3.0.9  pandas==1.1.5  python-dateutil==2.8.2  pytz==2021.3  six==1.16.0  xlrd==2.0.1  XlsxWriter==3.0.2 |

# Usage

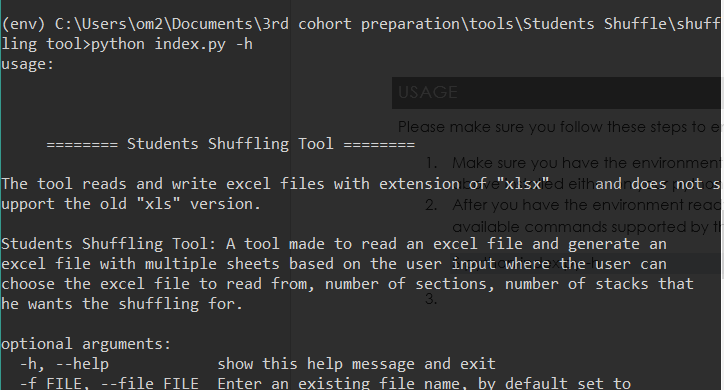
Please make sure you follow these steps to ensure that the tool runs perfectly on your system:

1. Make sure you have the environmental variables mentioned in the [requirements](#_System_requirements) section above installed either on your python local environment or the global one.

## launching the tool

1. After you have the environment ready you can launch the tool in help mode to see the available commands supported by the tool

$ python index.py -h



After you run the command $ python index.py –h you will see the description of the too in addition to optional arguments as a list contains all available commands you can perform with the tool

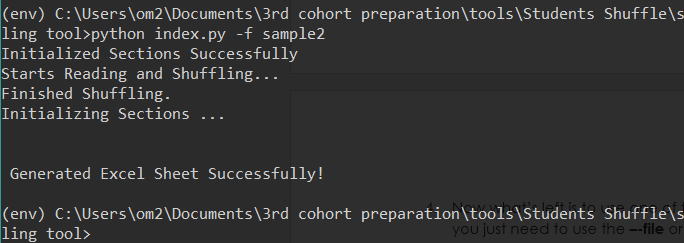
1. Inside the same folder I attached 2 samples of excel files, where you can get a clue about the type of formatting you should adapt your excel file on.

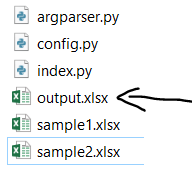
|  |  |
| --- | --- |
| Sample1: | Sample2: |
|  |  |

1. Now what’s left is to use one of these samples or your own excel file to get shuffled output of the file, to do that you just need to use the **–-file** or **–f** command to enter a file name in the same directory of the app in order to be able to read it

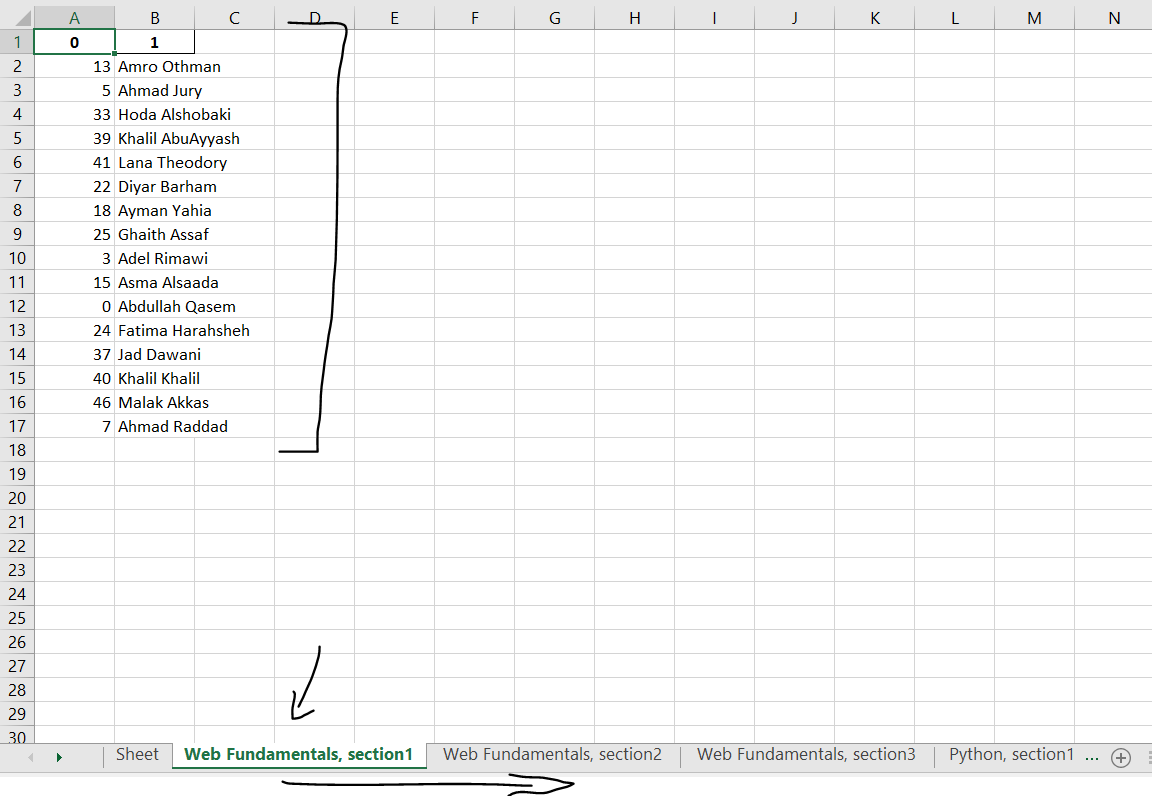
$ python index.py --file sample2

1. If you hit enter now, you will get a file called “**output.xlsx**” which is the default name and shuffling type if the user didn’t specify the type.





1. If we opened the generated file “**output.xlsx**”:



1. As you can see in the image in the 6th step we have multiple sheets and inside each sheet there are the students we shuffled, the shuffling is based on number of sections and number of stacks, where each sheet represents a **section per stack** example:
   1. Web Fundamentals, section1, Web Fundamentals, section2, Python, section1…
2. In the command in step 5, we shuffled the students in “**sample2.xlsx**” using shuffling type1, which is set by default if we didn’t specify the type of shuffling we want from the tool, as for now there are 2 shuffling types supported by the tool, (**shuffling type 1**, **shuffling type 2**).

## read custom columns from excel file

If you want to add your own excel file and read specific columns you can us this command

“-rclsl, --rcolslist” which stands for write columns list, for example ID, Name

$ python index.py -f sample2 – rcolslist ID, Name

## change generated column names in the excel file

If you want to add your own excel file and rename the generated columns you can us this command

“-wclsl, --wcolsl” which stands for write columns list, for example ID, Name

$ python index.py -f sample2 –wcolsl ID Name

By default the column names are indexed like 0, 1 for the shuffling type 1,

And for the shuffling type 2, column names are based on the section number like section1, section2…. .

## change output file name

If you want to add your own excel file and rename the generated columns you can us this command

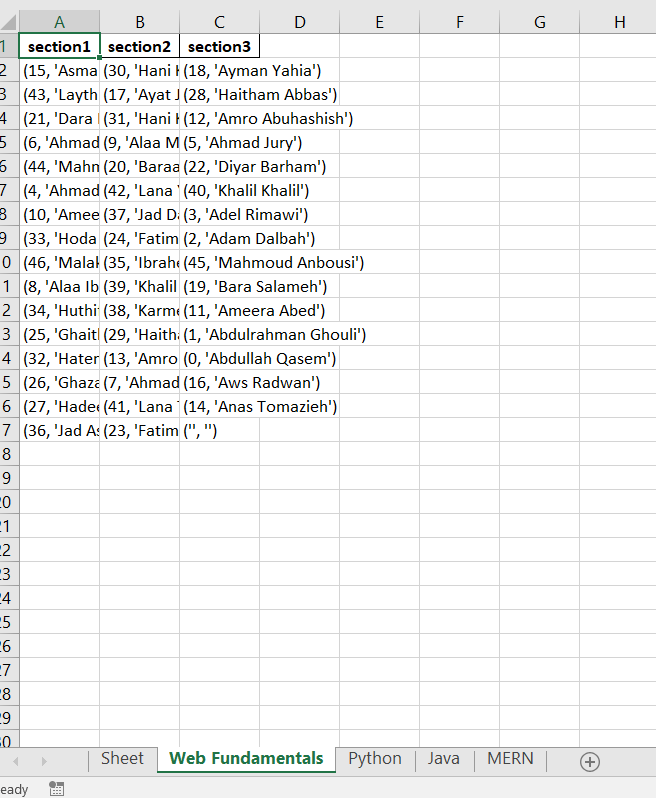
-gexn, --generatedexcelname, which stands for generated excel name.

$ python index.py - gexn students\_shuffled

## Change the shuffling type:

To change the type of shuffling you can switch to it by using the flag “**--shuffletype** “:

$ python index.py -f sample2 --shuffletype 2



As you can see the shuffling now is based on each stack and each column represents a section inside that stack.

For now the too supports only 2 shuffling options:

1. **Stack, Section** per sheet
2. **Stack** per sheet

# Table of commands

I’ll list the table of commands/operations that the tool can perform with the description of each command and when to use, also note that this list can be accessed from the command line using the first command

$ python index.py –h

|  |  |
| --- | --- |
| -f , --file | Enter an existing file name, by **default** set to  "sample1",  Reads string. |
| -p, --path | Enter a path file system  Ex."c:\users\m2\osaid\sample.xlsx"  Reads file system path. |
| -sct, --sections | Enter the numbers of sections to shuffle,  by **default** set to "3",  Reads integer. |
| -st , --stacks | Enter a list of stacks to shuffle  Ex. "Web Fundamentals" "Python" "Java" ... , by **default** set to ["Web Fundamentals", "Python", "Java", "MERN"],  Reads **string.** |
| -gexn, --generatedexcelname | Enter the name of the generated excel file  Ex. "test1", by **default** set to "output",  Reads **string.** |
| -wclsl, --wcolsl | Enter generated column names, by **default** [] works only with shuffling type "1",  Reads **multiple arguments of any type.** |
| -sft, --shuffletype | Enter a number that represents a supported shuffling type as for now there is two shuffling types "1" and "2", by **default** set to "1",  Reads integer. |
| -wst, --withstyle | Enter bool for so basic styling, currently this feature is not ready to release :3.  Reads **boolean** (true, false, t , f, 0, 1) |
| -rcln, --rcolname | Enter column name to read from, by **default** "Name"  Reads **string.** |
| -rclsl, --rcolslist | Enter a list of columns to read from 2 columns max for now :((((, by **default** empty [],  Reads **multiple arguments of any type.** |