

# DummPy

Random Data Generator

User Guide

## Contents

Chapter 0: Introduction and How to Execute .....	- 2 -
Chapter 1: Main Window .....	- 3 -
Chapter 1.1 Exit / Reset confirmations.....	- 4 -
Chapter 2: The Menu .....	- 5 -
Chapter 2.1: Save.....	- 6 -
Chapter 2.2: Load.....	- 6 -
Chapter 2.3: Theme.....	- 7 -
Chapter 2.4: About.....	- 7 -
Chapter 3: Configuration.....	- 8 -
Chapter 3.1: Name and e-mail.....	- 8 -
Chapter 3.2: Number.....	- 9 -
Chapter 3.3: Location.....	- 10 -
Chapter 4: Generate the Excel file.....	- 11 -
Credits.....	- 11 -

## Chapter 0: Introduction and How to Execute

---

DummPy is a random data generator written in Python. With DummPy you can produce random instances or combinations of the sample data, that are provided in an external Excel file, save them in a Python dictionary, and create a new Excel file through a dataframe created with the dictionary random data.

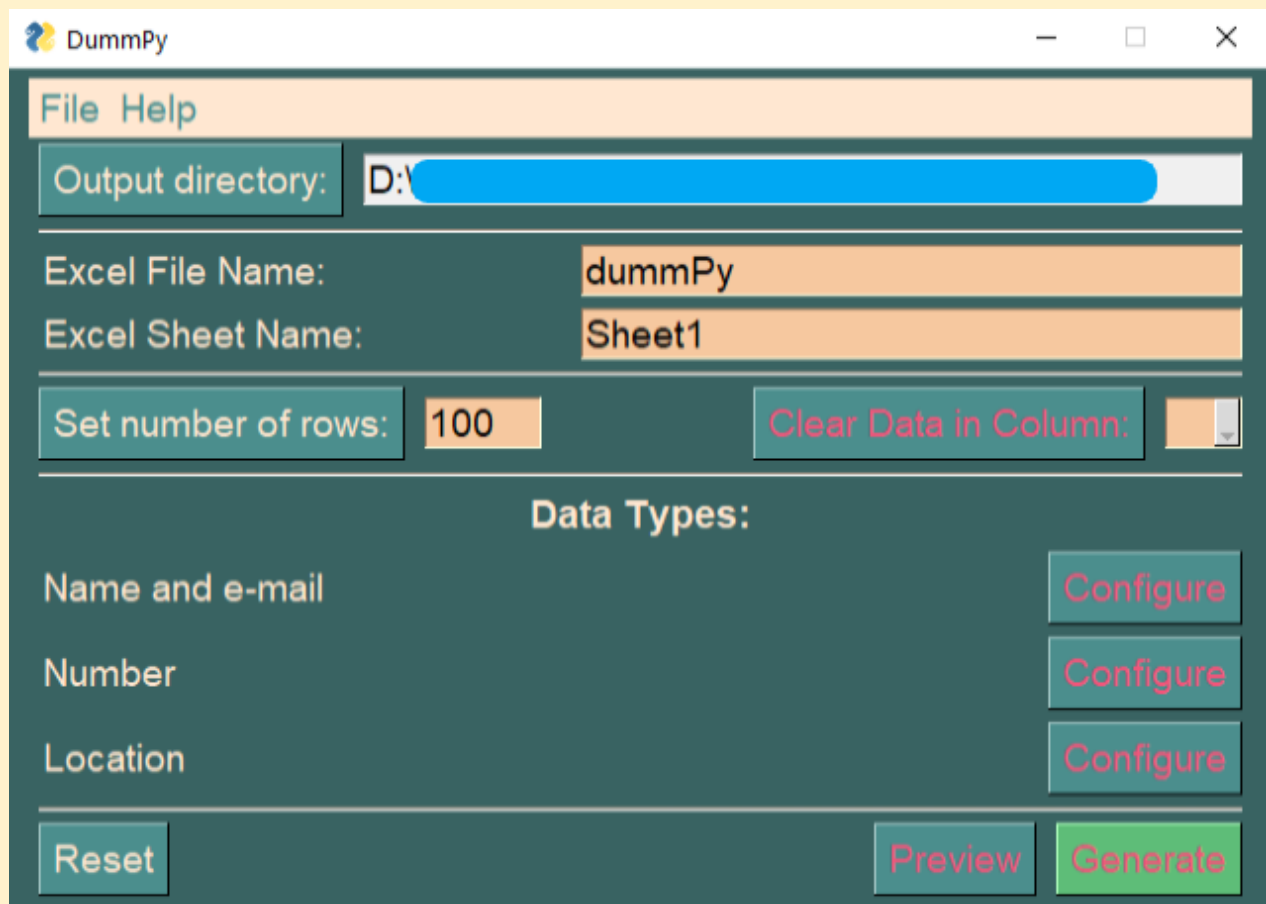
In order to run the application on a Windows PC, you will need to have a recent version of [Python](#) installed on your computer (version 3.10 was used to write the code, so it should work if anything goes wrong). With Python installed, open Powershell and type the following commands, followed by pressing the Enter key after each one:

- 1) **cd "directory path"** : this sets the current directory. The directory path is where the main.py file is located (for example: cd "C:\dummPy").
- 2) **pip install -r requirements.txt** : this will install any dependencies that are required to run the application. Before this step, you can also create a new Python virtual environment and activate it.
- 3) **python main.py** : this will run the main.py file and start the application. The first thing you will see is the [Main Window](#).

If you don't have a lot of experience with coding, instead of the above, you can navigate to the [releases](#) page in the GitHub repository and download the latest release version, which contains a standalone executable file (no need to install python or perform the above console commands).

Thank you for using DummPy!

## Chapter 1: Main Window



The screenshot shows the 'DummPy' application window. At the top is a menu bar with 'File' and 'Help'. Below it is a section for 'Output directory:' with a text input field containing 'D:\'. This is followed by 'Excel File Name:' with a text input field containing 'dummPy', and 'Excel Sheet Name:' with a text input field containing 'Sheet1'. Below these is a 'Set number of rows:' section with a text input field containing '100' and a 'Clear Data in Column:' button next to a dropdown menu. The 'Data Types:' section lists 'Name and e-mail', 'Number', and 'Location', each with a corresponding 'Configure' button. At the bottom are three buttons: 'Reset', 'Preview', and 'Generate'.

*The Main Window*

At the top of the window, we can see the [Menu Bar](#), which will be discussed in the following chapter.

Directly below it, is the button used to select the output directory of the created Excel file. By default, this is set as the directory where the main.py file is located.

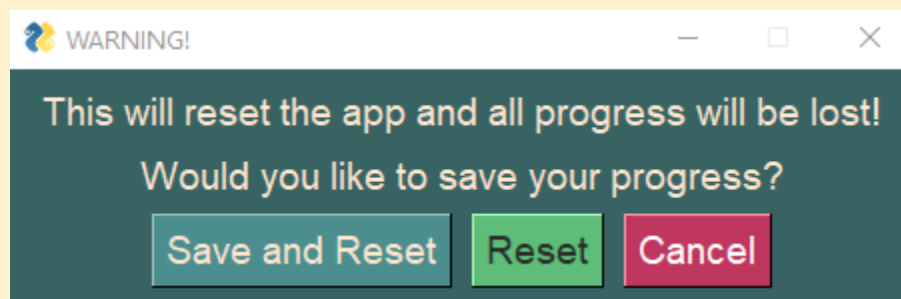
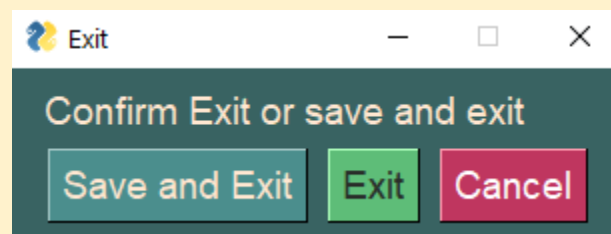
In the section below, we can set the Excel File Name and Sheet Name where our data will be created.

The following section is used to set the number of rows of data you want to create, with an acceptable value of up to 99999. We can also remove a column from the dictionary by selecting the column from the dropdown menu and clicking the [Clear Data in Column:](#) button. The dropdown menu and the button are disabled until the rows are set or a dictionary is [loaded](#).

The Data Types section allows for the [configuration](#) of the various data types you want to add to your file. The **Configure** buttons are initially disabled, and will only be enabled after the rows are set in the above section, or if an existing dictionary is [loaded](#).

Finally, there is the option to reset the application, preview the dataframe, and [generate](#) the Excel file. The **Preview** and **Generate** Buttons are also disabled until the rows are set or a dictionary is loaded.

## Chapter 1.1 Exit / Reset confirmations



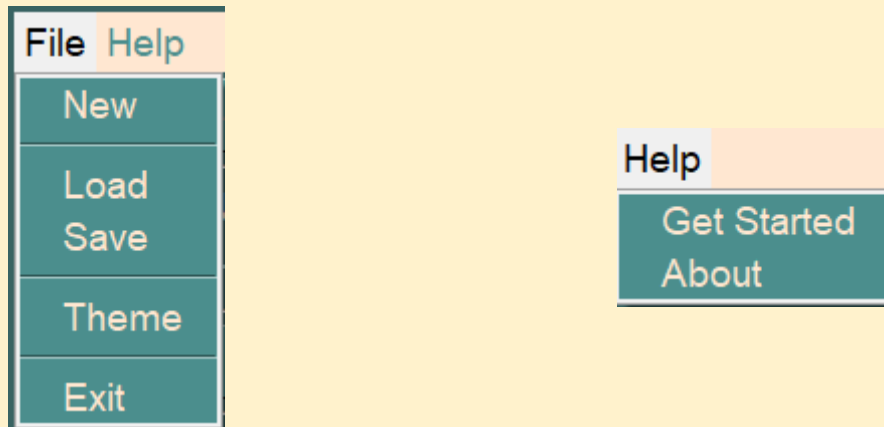
*The Exit (above) and Reset (below) confirmation Windows*

If you attempt to close the application or reset it while there are data in the dictionary, you will be prompted to Save beforehand, or continue the action. By pressing the **Save and Exit** button you will be able to save your progress in an external .txt file. You can find more information on the Save function in [Chapter 2.1](#).

If you attempt to close or reset the application while there is no data in the dictionary, then the action will be instant.

## Chapter 2: The Menu

---



*The Menu options*

By clicking the **File** menu, you will get the following options:

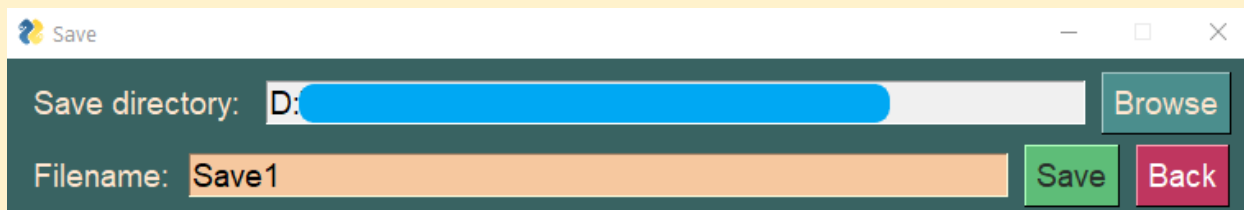
- 1) **New** : Close the existing window and create a new one. This has the same functionality as the **Reset** button in the Main Window.
- 2) **Load** : Load a dictionary of values from an external .txt file.
- 3) **Save** : Save dictionary in an external .txt file
- 4) **Theme** : Change the theme of the app and reset it.
- 5) **Exit** : Exit the application.

By clicking the **Help** menu, you will get the following options:

- 1) **Get Started** : This option will open a browser tab or window to this file located on the GitHub repository of the project.
- 2) **About** : Show general information about the application.

## Chapter 2.1: Save

---

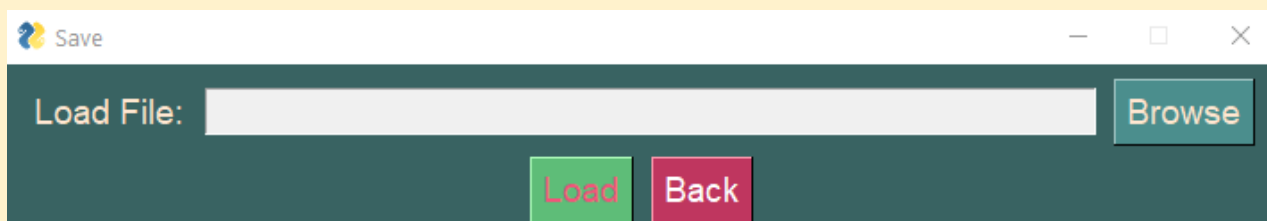


*The Save Menu*

With the **Save** menu you can save your progress in a .txt file, in order to [load](#) it at a later time. Simply click **Browse** to select the save file directory (default is the directory with the main.py file), followed by the **Save** button.

## Chapter 2.2: Load

---



*The Load Menu*

With the **Load** submenu you can load previously [saved](#) dictionaries from a .txt file. Click on the **Browse** button to navigate and select the file, followed by the **Load** button.

After loading a dictionary file, you will notice that the number of rows will be set according to the loaded dictionary, and all the disabled buttons in the Main Window will be enabled.

## Chapter 2.3: Theme

---

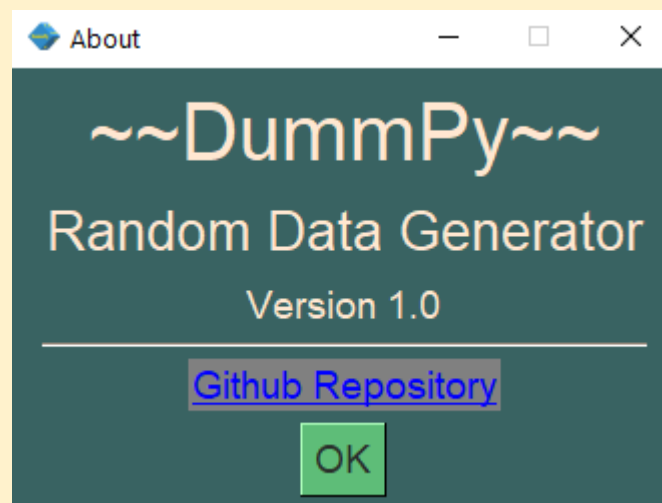


*The Theme Menu*

With the Theme submenu you can change the theme of the application, by selecting one of the available themes from the dropdown menu, followed by clicking **OK**.

## Chapter 2.4: About

---



*The About Window*

By selecting the About submenu, you will see the About Window, with information on the version of the application, as well as the link to the [Github Repository](#).



## Chapter 3: Configuration

Once you have selected the number of lines of random data you want to create, the **Configure** buttons will become active, and you will be able to specify all the details about the created data. In this chapter we will see all the configuration options for every data type.

### Chapter 3.1: Name and e-mail

First Name Samples	Last Name Samples	E-mail Domain Samples
James	Smith	gmail.com
Robert	Johnson	yahoo.com
John	Williams	outlook.com
Michael	Brown	
David	Jones	

☒ Add Full Name on Column: with title: Name

☐ Add First Name on Column: with title: First Name

☐ Add Last Name on Column: with title: Last Name

☐ Add email address on column: with title: E-mail Address

Preview Sample Reload Add Back

*The Name and E-mail Configuration Window*

At the top of the **Name and E-mail Configuration** window you will see three lists created with data from the external *SAMPLE\_DATA.xlsx* file.

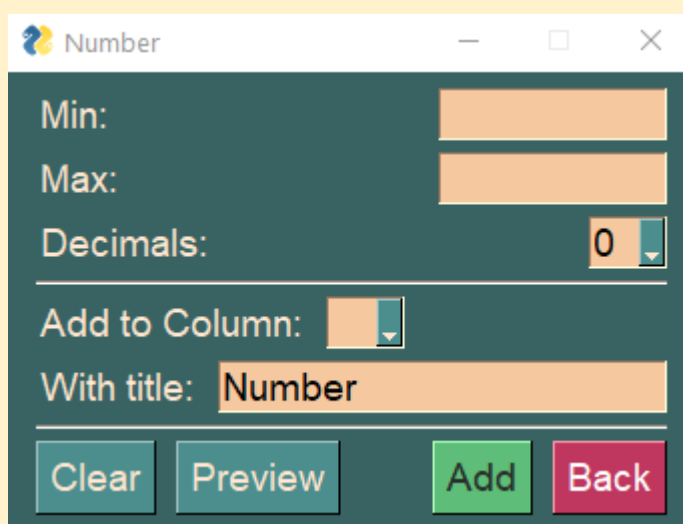
The next two sections contain the option on how you want the data to be created. You can choose to add full name data by checking the first checkbox, only first name data by checking the second checkbox, only last name data by checking the third checkbox, or both first and last name in separate columns by checking both the second and third checkboxes.

For each option you will have to determine the Excel column from the dropdown list, as well as provide a column title name.

The next section provides the option to add an e-mail address to a specified column determined by the names generated.

Finally, in the last section you have the option to preview the dataframe, reload the sample data from the external file, if you have made any adjustments, and add the specified data to the dictionary.

## Chapter 3.2: Number



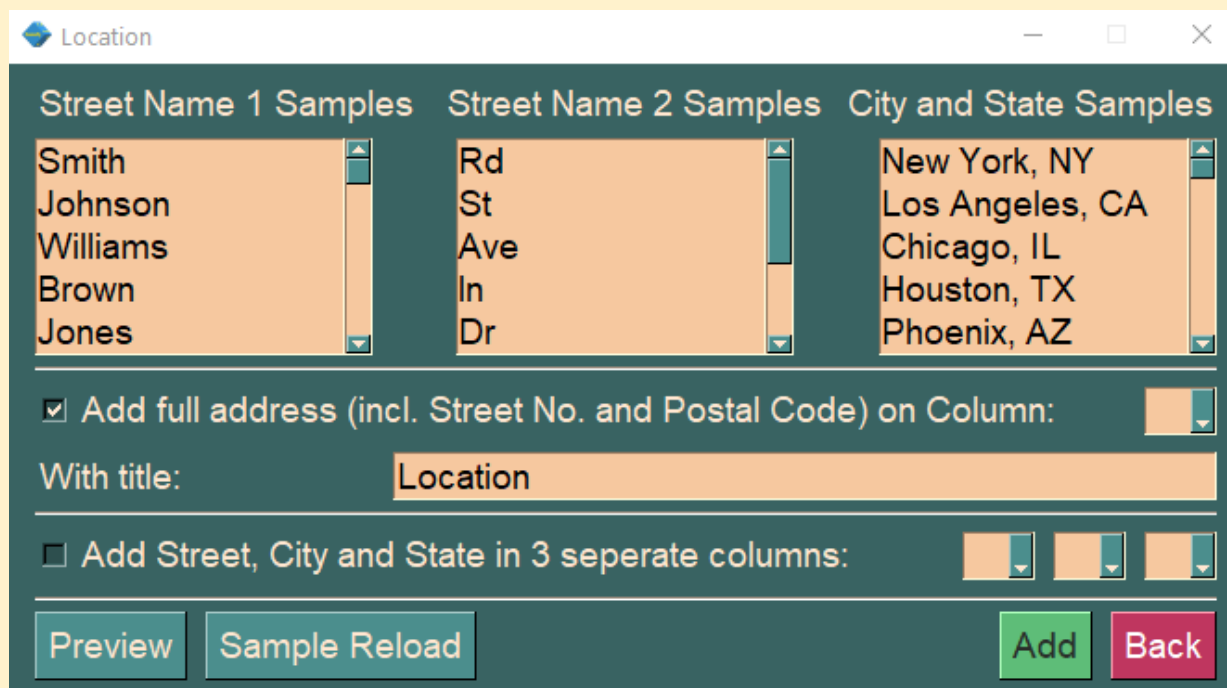
*The Number Configuration Window*

At the top of the **Number Configuration** window, you will be able to specify a range in which the random numbers will be generated, as well as how many decimals they should have, up to 8.

In the next section you will be able to specify the Excel sheet column where the data will be added by selecting it from the dropdown list, as well as the column title.

Finally, in the last section you will be able to Clear the data in this window, Preview the dataframe, and add the data to the dictionary.

## Chapter 3.3: Location



The screenshot shows a window titled "Location" with three columns of sample data: "Street Name 1 Samples", "Street Name 2 Samples", and "City and State Samples". The first column lists names (Smith, Johnson, Williams, Brown, Jones), the second lists street suffixes (Rd, St, Ave, In, Dr), and the third lists city and state pairs (New York, NY, Los Angeles, CA, Chicago, IL, Houston, TX, Phoenix, AZ). Below these lists are three configuration options: a checked checkbox for "Add full address (incl. Street No. and Postal Code) on Column:" with a dropdown menu, a text field for "With title:" containing "Location", and an unchecked checkbox for "Add Street, City and State in 3 separate columns:" with three dropdown menus. At the bottom are four buttons: "Preview", "Sample Reload", "Add", and "Back".

Street Name 1 Samples	Street Name 2 Samples	City and State Samples
Smith	Rd	New York, NY
Johnson	St	Los Angeles, CA
Williams	Ave	Chicago, IL
Brown	In	Houston, TX
Jones	Dr	Phoenix, AZ

☒ Add full address (incl. Street No. and Postal Code) on Column:

With title:

☐ Add Street, City and State in 3 separate columns:

Preview Sample Reload Add Back

*The Location Configuration Window*

At the top of the **Location Configuration** window, you will see three lists of sample data that will be used to generate the random location data.

In the next section, check the checkbox if you want the full address added in a single column. You will be able to specify the Excel Column from the dropdown list, as well as the column title. This option will also add a random street number and postal code to the address.

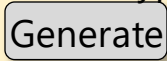
If, however, you want the data in separate columns, you will have to check the checkbox in the next section, along with specifying the Excel columns for each of Street name, City and State. Street number and Postal Code are not included in this option, but you can add them with the [Number](#) data type.

Finally, in the last section you can preview the dataframe, reload the sample data from the external file, if you have made any adjustments, and add the specified data to the dictionary.

## Chapter 4: Generate the Excel file

---

***It is advisable to save the dictionary before generating the Excel file. While there are many fail-safes in place, the application could still crash and you will lose your data!***

After you have configured all the data types that you want, simply return to the [Main Window](#) and click on the  button. Depending on the size of the data, you might have to wait a few seconds until the file is generated. After the confirmation, ensure that you can open the generated file in excel and feel free to close the application or start over.

## Credits

---

Sample data sources:

- <https://www.ssa.gov/oact/babynames/decades/century.html>
- <https://www.thoughtco.com/most-common-us-surnames-1422656>
- <https://www.moving.com/tips/largest-cities-in-us/>
- [https://www.faa.gov/air\\_traffic/publications/atpubs/cnt\\_html/appendix\\_a.html](https://www.faa.gov/air_traffic/publications/atpubs/cnt_html/appendix_a.html)

Inspiration for this project:

- <https://github.com/Sven-Bo/advanced-gui-with-usersettings-and-menubar>