
spreadscript Documentation

Release latest

Sep 29, 2018

Contents:

1	Introduction	3
2	Installation	5
2.1	From source	5
3	Usage	7
3.1	Command line interface	7
4	Library	11
5	Contributors	13

This project provides a way to use spreadsheets from the command line or from Python programs. In this way, spreadsheets can be used in automated data analysis processes.

Please see [ReadTheDocs](#) for the latest documentation.

CHAPTER 1

Introduction

This project provides a command line interface and an API for spreadsheets.

The inputs and outputs are defined by two tables in a new sheet named “Interface”. SpreadScript will read the input variables from column B and the values from column C. Likewise, the output variables are read from column E and their values from column F. In both cases, the variables are read from row 4 onward until an empty cell is encountered.

This method should work with any format that is supported by [LibreOffice Calc](#). It has been tested using file formats ODS, XLS and XLSX.

Prerequisites:

```
apt install python3-uno
```

The software is distributed via [PyPI](#), it can be installed with `pip`:

```
pip install spreadscript
```

2.1 From source

The source is hosted on [GitHub](#), to install the latest development version, use the following commands.

```
git clone https://github.com/jfjlaros/spreadscript.git
cd spreadscript
pip install .
```


Suppose we have the following table with heights of family members.

Since Janie and Johnny have not reached full height, we might want to export their heights as input variables. Suppose we are interested in the average height and the tallest person in this family. This information goes to the “Interface” sheet.

In this sheet we put the input variables in column B and references to the values in column C. The value of C4 is `=Sheet1.C5` and that of C5 is `=Sheet1.C6`.

Likewise, the output variables are put in column E and references to the values in column F. The value of F4 is `=Sheet1.C7` and that of F5 is `=Sheet1.C9`.

3.1 Command line interface

With the command line interface, the input and output table can be read.

```
$ spreadsript read_input data/test.ods
{"height_janie": 1.41, "height_johnny": 1.52}

$ spreadsript read_output data/test.ods
{"tallest": 1.76, "average": 1.5775}
```

To manipulate the input, use the `process` subcommand:

```
$ spreadsript process data/test.ods '{"height_johnny": 1.56}'
{"tallest": 1.76, "average": 1.5875}
```

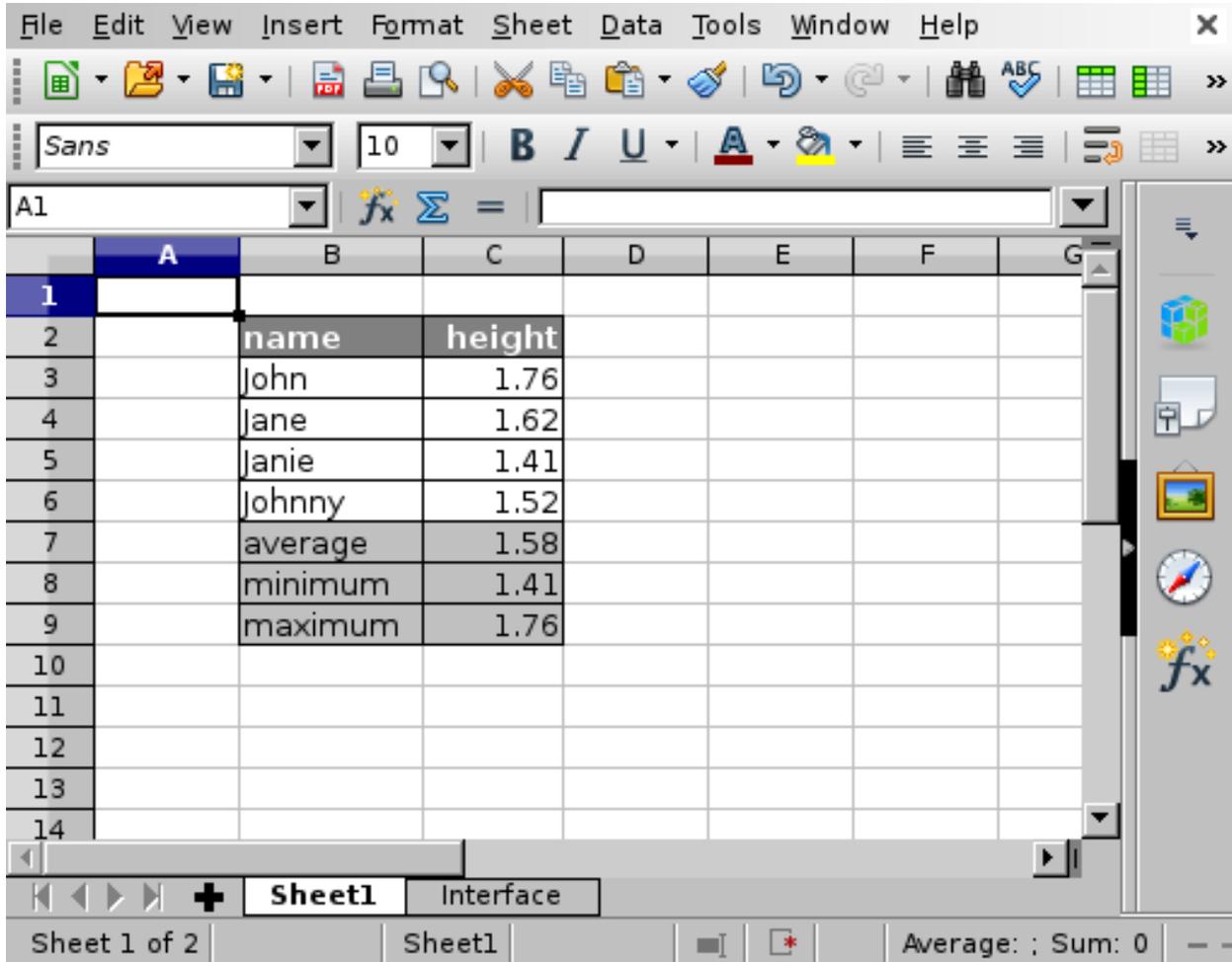


Fig. 1: Example table.

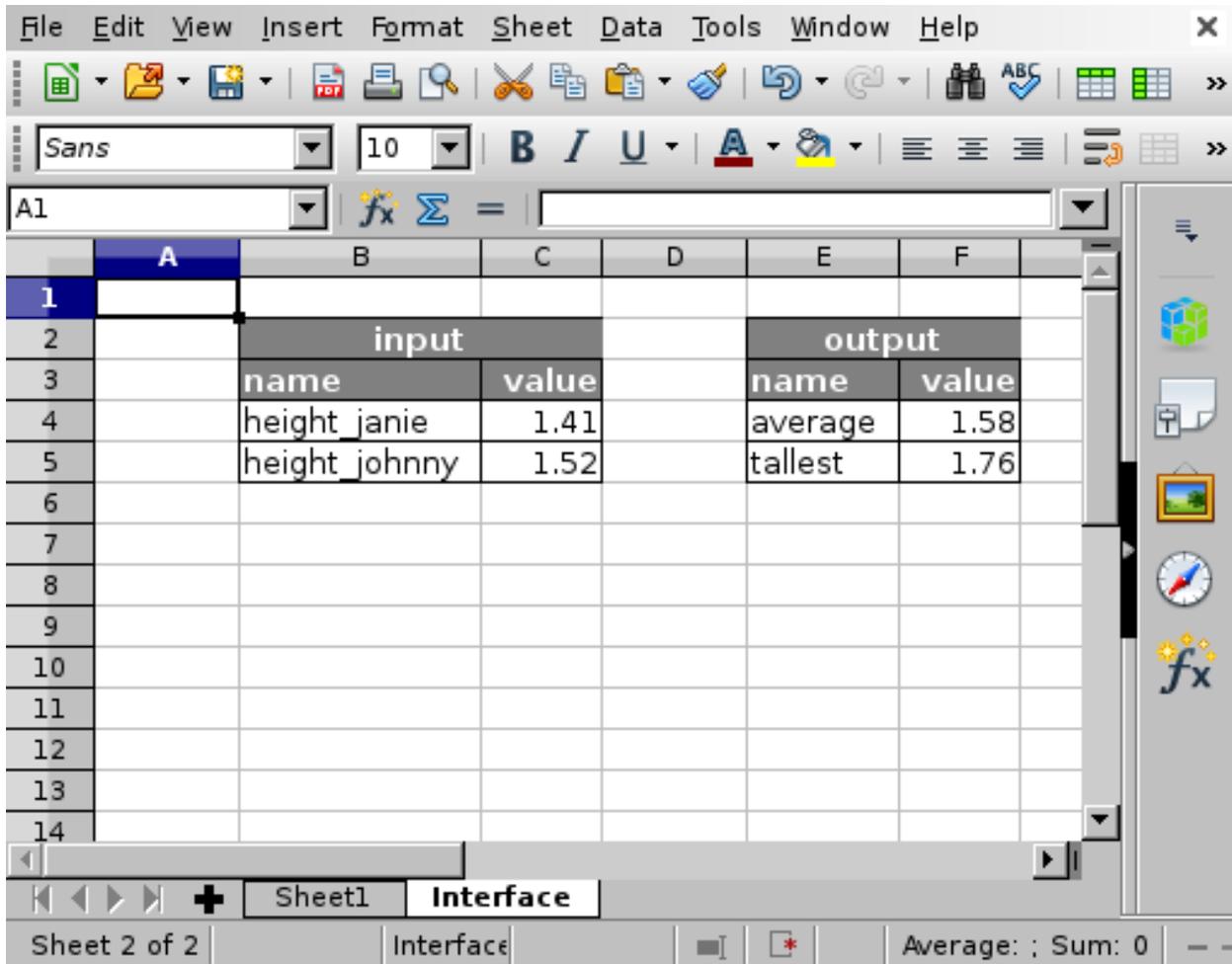


Fig. 2: Example interface.

First import the SpreadScript class and load a spreadsheet.

```
>>> from spreadscript import SpreadScript
>>>
>>> spreadsheet = SpreadScript('data/test.ods')
```

The input and output variables can be read with the `read_input` and `read_output` methods respectively.

```
>>> spreadsheet.read_input ()
{'height_johnny': 1.52, 'height_janie': 1.41}
>>>
>>> spreadsheet.read_output ()
{'average': 1.5775, 'tallest': 1.76}
```

The `write_input` method is used to update any variables.

```
>>> spreadsheet.write_input({'height_johnny': 1.56})
>>> spreadsheet.read_output ()
{'average': 1.5875, 'tallest': 1.76}
```


CHAPTER 5

Contributors

- Jeroen F.J. Laros <jlaros@fixedpoint.nl> (Original author, maintainer)

Find out who contributed:

```
git shortlog -s -e
```