

ScriptOven – User Guide

ScriptOven was designed for people who would like to manipulate data with their own scripts, but who don't have the access permissions to use a command shell, or to install a scripting language (such as Perl or Python).

Users can instead write their script in Javascript, and then upload them onto ScriptOven, to be ran on their data. There are also helper functions within ScriptOven that make it easier to work with CSV data (see the “testMainScript.js” file, in the project's root directory, for an example of this).

Contents:

- 1. Writing Your Scripts**
- 2. Working With CSV Data**
- 3. Compatibility**

Writing Your Scripts:

(To see an example script, look at "testMainScript.js" in the project's root directory.)

When writing your scripts, there is 1 function that is required, and 2 optional functions (Please note, while these 3 functions will be overridden or forgotten when you load in a new script file, any other functions or global variables in the first file may remain.).

main(inputText)

The required function will be called "main", and should take only 1 parameter (which will be a string). The parameter will be the input data that is entered/uploaded into ScriptOven. The "main" function will be ran when the "Run Function" button is clicked. This function must return a string value (which will then be displayed in the output box, where it can be downloaded).

assertions()

If the "assertions" function is defined, it will be called anytime you click "Run Function", before the "main" function is ran. This function allows you to define tests to make sure that any changes to your script have not caused it to do something it shouldn't.

This function should return an array, and this array should contain 1 or more arrays. Each of these arrays should contain 2 string values. The first value is the string to be inputted into the "main" function, and the second value is the string you expect to be returned.

If any assertions fail, an error message containing more information will be displayed in the output textbox.

options()

Create an "options" function to define how ScriptOven should behave in certain scenarios. This function should return a single object, that has each option as a property. If you do not define an option's property in this object, then its default value will be used.

The current available options are:

- `alertWhenAssertionsFailed` – Should a browser alert be displayed if 1 or more assertions have failed? This is a boolean value.
- `consoleLogWhenAssertionsFailed` – Should the error message for failed assertions also be logged to the console? This is a boolean value.

Working With CSV Data:

ScriptOven has some built in functions to help you work with CSV files.

The first thing you will want to do is to convert your input string into an array of strings, separated by the line breaks. The “stringToArray” function will take the input text as a parameter, and then return the data as an array of separated lines.

Next, you will want to convert these lines into an array of objects. To do this, you can use the “csvToArray” function. The first parameter to this function will be your array of CSV lines. If the first line of your CSV file contained the data's headers, then this is the only parameter you'll need to provide. If the headers weren't included, however, then you will need to provide them as the second argument (This parameter should be an array of strings, where each string is a header. The order of the headers will need to match the order of the fields in your data.). Each object in the returned array will contain a parameter for each header that was provided.

Once you are done working with the data within the objects, you may then want to convert them back into an array of CSV lines. To do this, you will need to use the “objectsToCSVLinesArray” function. This function will take 2 parameters. The first is an array of the headers, and the second is the array of objects to be converted. If you do not already have an array of the headers, you will be able to generate one with the “csvLineToStringArray” function (just pass the headers line as the parameter for this). The “objectsToCSVLinesArray” function will return an array of CSV formatted lines (the first line being the headers).

Finally, you can convert this array of strings into one string – ready to be returned from “main” – with the “lineArrayToMultilineString” function, which will take the array of CSV lines as a parameter.

Compatibility:

ScriptOven is compatible with most modern browsers. That being said, it will not be compatible with Internet Explorer. Also, due to Safari no longer being available for Windows, it has not yet been tested on Safari.