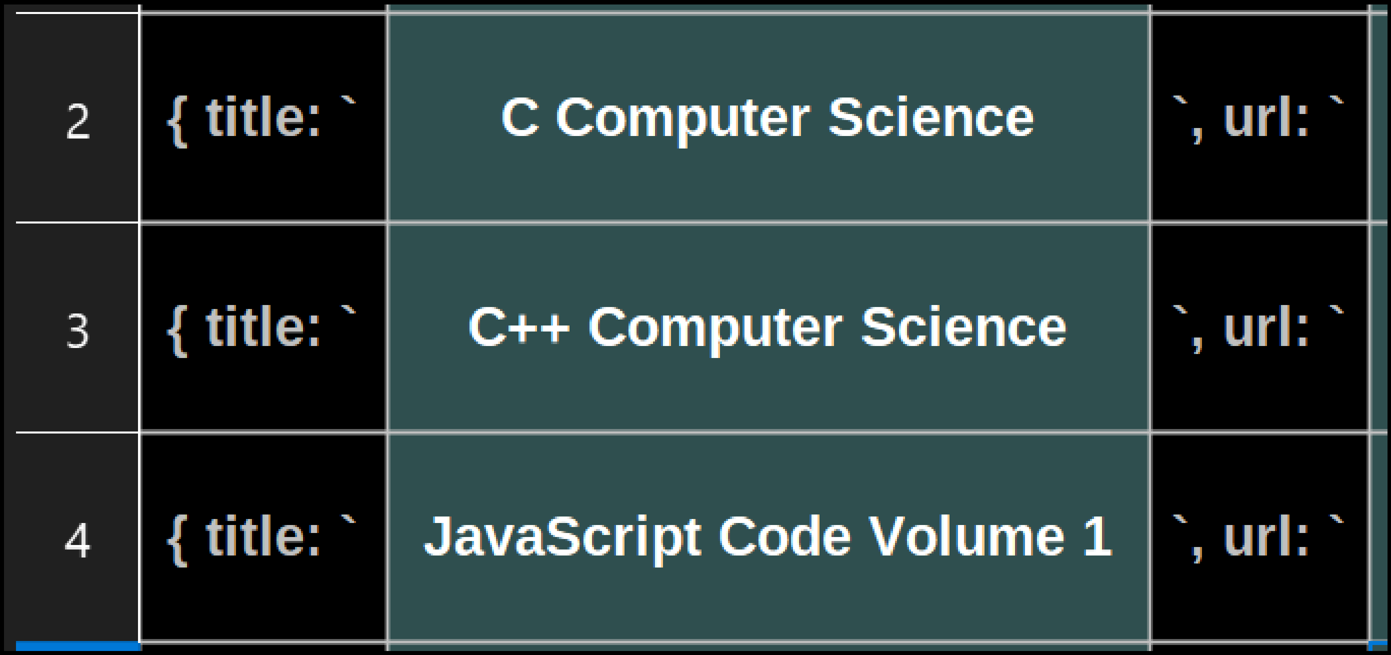
Topalian Spreadsheet to JavaScript

by

Christopher Andrew Topalian

Dedicated
to
God the Father

We format our data with the syntax included in the spreadsheet.

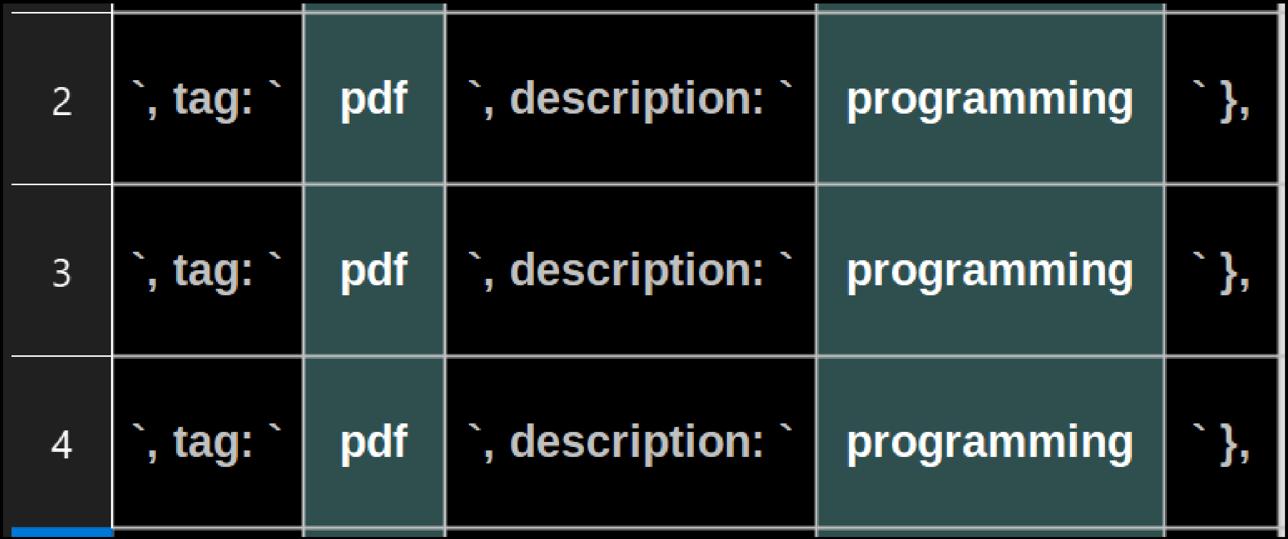


2	{ title: `	C Computer Science	`, url: `
3	{ title: `	C++ Computer Science	`, url: `
4	{ title: `	JavaScript Code Volume 1	`, url: `

We will Concatenate the cells of the row, using a formula, at the end of each row.

= CONCAT(A2:I2)

Here is the ending part of the row, with `},
to signify the end of each object.

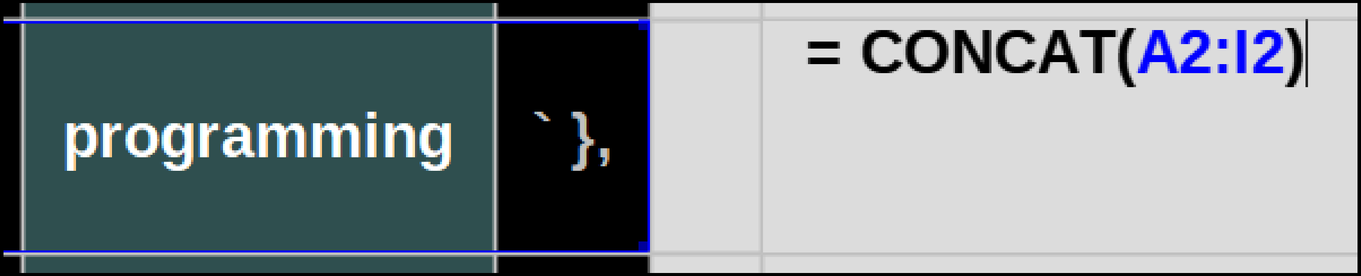


2	`, tag: `	pdf	`, description: `	programming	` },
3	`, tag: `	pdf	`, description: `	programming	` },
4	`, tag: `	pdf	`, description: `	programming	` },

As we see above, the black column is the key and the lighter column is the value.

In this way, each row is now an object in our spreadsheet, which is an array of objects.

At the end of each object we use the formula: = CONCAT(A2:I2)

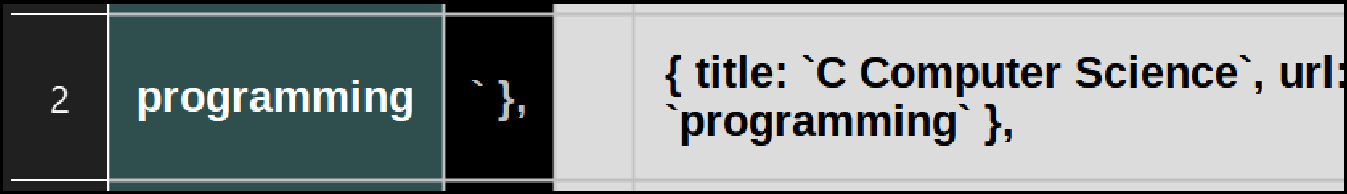


programming

` },

= CONCAT(A2:I2)

The result of this formula is shown below:



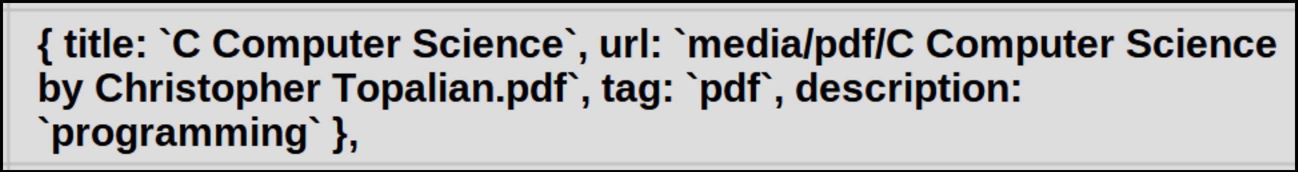
2

programming

` },

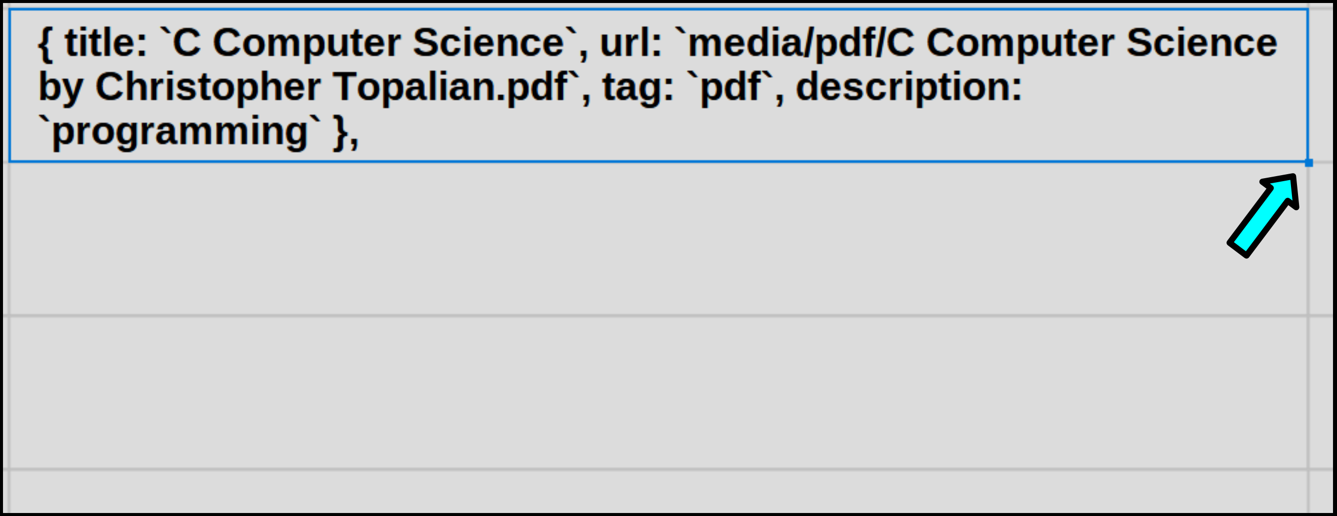
{ title: `C Computer Science`, url:
`programming` },

As we can see, we have a complete object, contained using { and }



{ title: `C Computer Science`, url: `media/pdf/C Computer Science by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },

Now we copy the formula to the other cells of each row



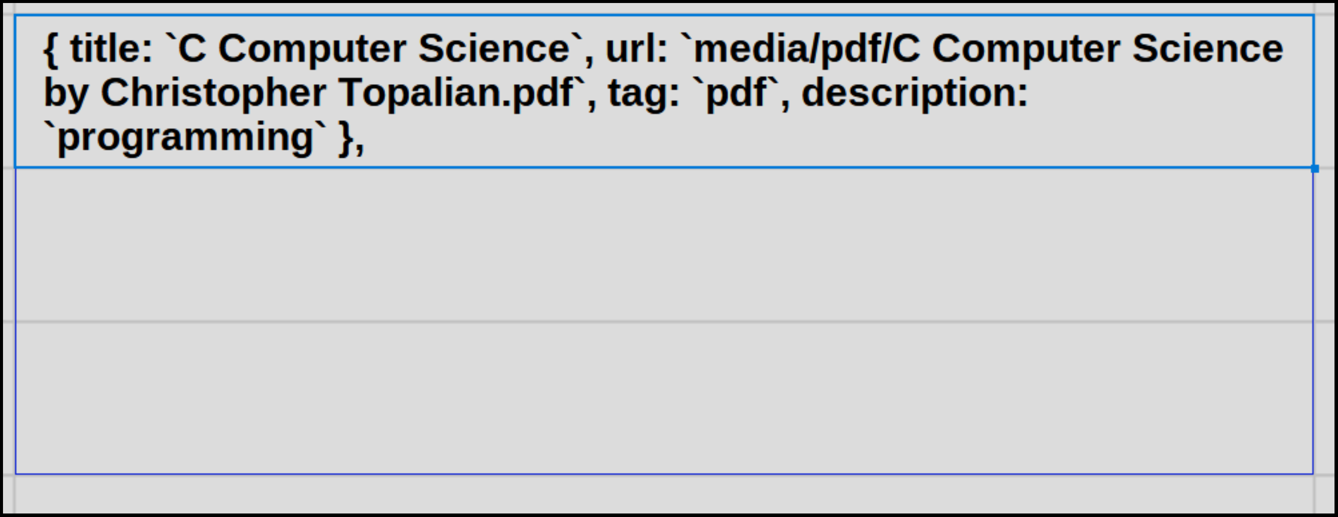
```
{ title: `C Computer Science`, url: `media/pdf/C Computer Science by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
```

Place Mouse Arrow at the Bottom right handle point of the cell, and then:
Hold Left Click + Drag Mouse Down

We drag the mouse down 3 cells, so that the formula will be used in those 3 cells.

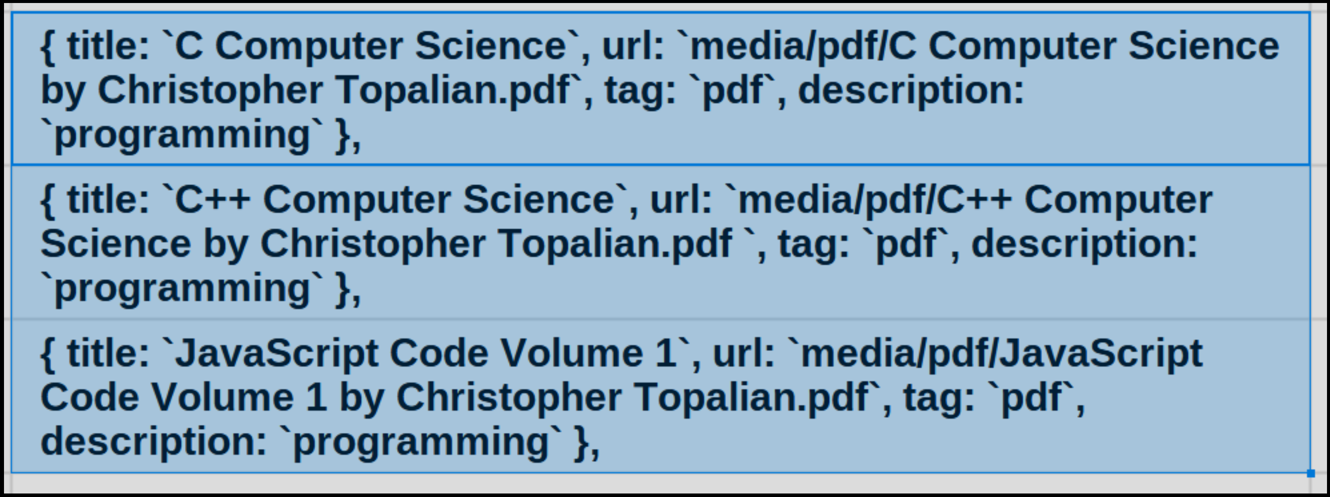
It will automatically correctly put them in sequence, so that each function will use the correct row information.

Mouse Dragging Handle Down 3 cells



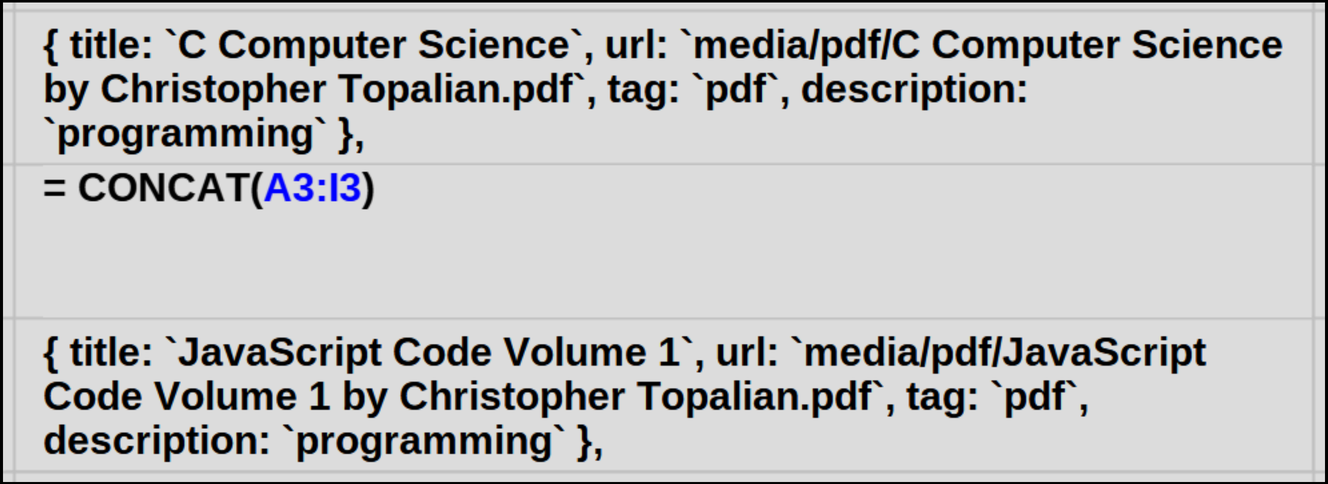
```
{ title: `C Computer Science`, url: `media/pdf/C Computer Science by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
```

The cells are now filled in with the applied formula to each row.



```
{ title: `C Computer Science`, url: `media/pdf/C Computer Science by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
{ title: `C++ Computer Science`, url: `media/pdf/C++ Computer Science by Christopher Topalian.pdf `, tag: `pdf`, description: `programming` },
{ title: `JavaScript Code Volume 1`, url: `media/pdf/JavaScript Code Volume 1 by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
```

When we double click in the 2nd Row, we see the formula that is correctly



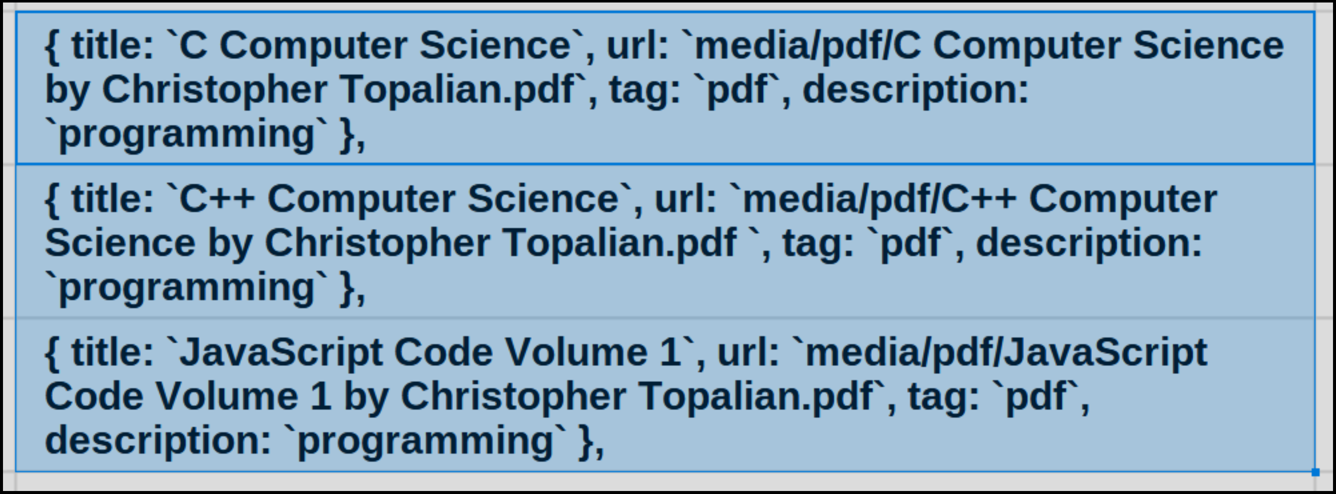
```
{ title: `C Computer Science`, url: `media/pdf/C Computer Science by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
= CONCAT(A3:I3)
{ title: `JavaScript Code Volume 1`, url: `media/pdf/JavaScript Code Volume 1 by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
```

We can see, that it shows: = CONCAT(A3:I3)

Thus, the data of each row is correctly being concatenated into one object at the end of each row.

Press Escape Button to escape from the formula editing mode.

We Copy all of the objects together at once



```
{ title: `C Computer Science`, url: `media/pdf/C Computer Science by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
{ title: `C++ Computer Science`, url: `media/pdf/C++ Computer Science by Christopher Topalian.pdf `, tag: `pdf`, description: `programming` },
{ title: `JavaScript Code Volume 1`, url: `media/pdf/JavaScript Code Volume 1 by Christopher Topalian.pdf`, tag: `pdf`, description: `programming` },
```

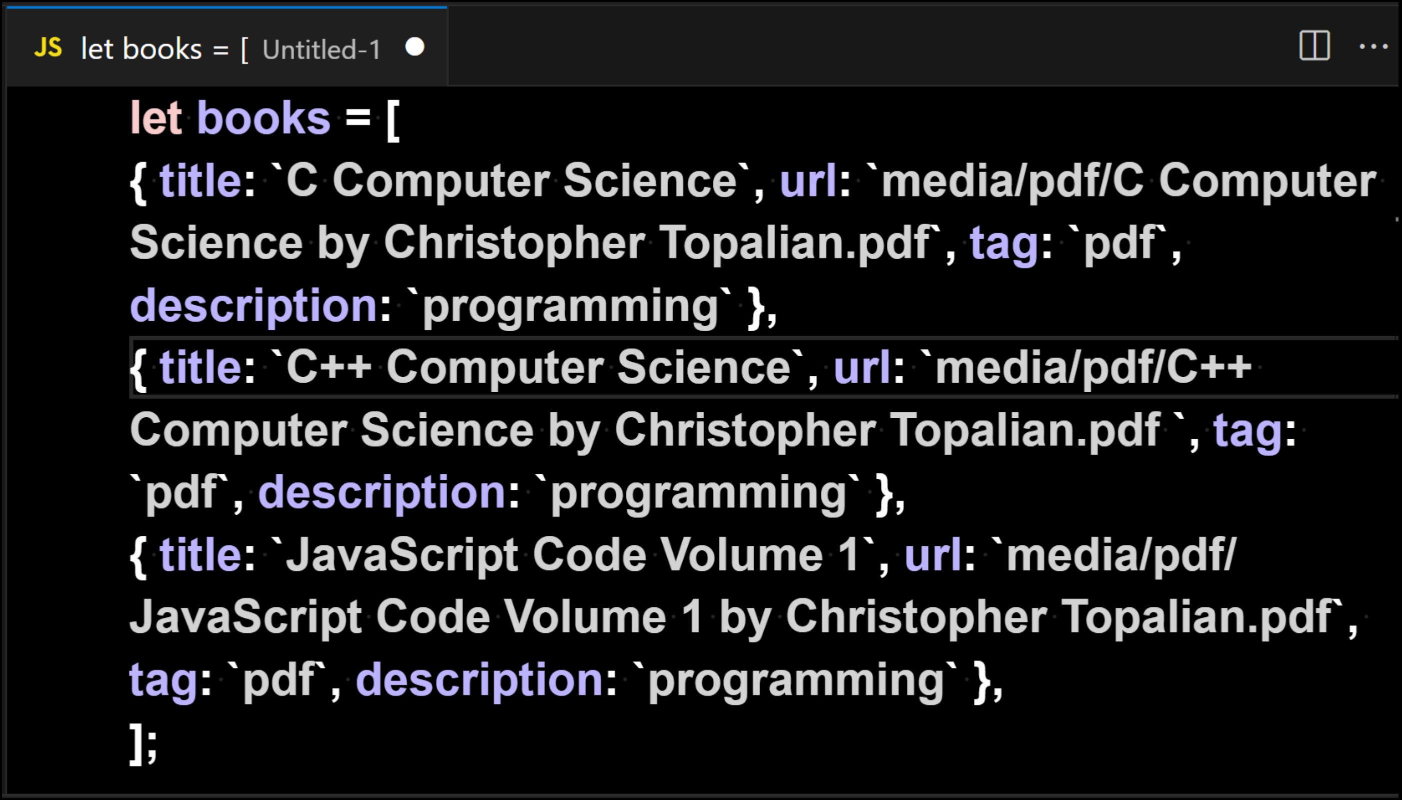
We highlight all cells of our object data and we then press Control + C to copy.

This copied data includes the many objects.

We will paste these objects into an array in VSCode.

We make an array called books by writing:
let books = [

and then we paste our data using Control + V



```
let books = [
{ title: `C Computer Science`, url: `media/pdf/C Computer
Science by Christopher Topalian.pdf`, tag: `pdf`,
description: `programming` },
{ title: `C++ Computer Science`, url: `media/pdf/C++
Computer Science by Christopher Topalian.pdf `, tag:
`pdf`, description: `programming` },
{ title: `JavaScript Code Volume 1`, url: `media/pdf/
JavaScript Code Volume 1 by Christopher Topalian.pdf`,
tag: `pdf`, description: `programming` },
];
```

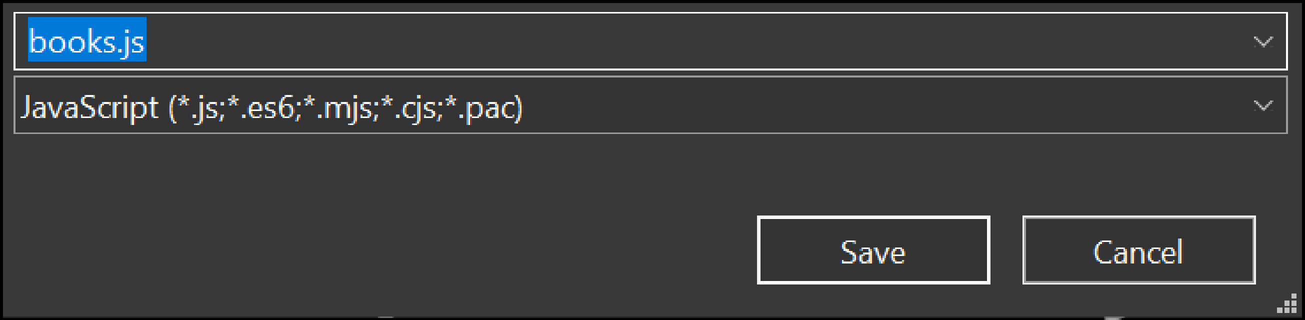
We write]; at the end of the array to close the array. Now we have an array of objects in JS.

```
// Here is the code shown clearer:

let books = [
{ title: `C Computer Science`, url:
`media/pdf/C Computer Science by
Christopher Topalian.pdf`, tag: `pdf`,
description: `programming` },
{ title: `C++ Computer Science`, url:
`media/pdf/C++ Computer Science by
Christopher Topalian.pdf `, tag: `pdf`,
description: `programming` },
{ title: `JavaScript Code Volume 1`, url:
`media/pdf/JavaScript Code Volume 1 by
Christopher Topalian.pdf`, tag: `pdf`,
description: `programming` },
];
```

We Save Our File

Hold Control + S to Save



books.js

JavaScript (*.js;*.es6;*.mjs;*.cjs;*.pac)

We Name the File: books.js

We Left Click: Save Button

We now have a saved JavaScript file that is an Array of Objects of our spreadsheet data!

This Array of Objects JavaScript file can now be used in any of our JS projects.

Summary:

Converting from spreadsheet data to JavaScript data is very easy by including the syntax in the spreadsheet itself and then using a formula at the end of each row to combine each of the cells of the row together into one cell, located at the end of each row.

We copy the formatted data and paste it into VSCode into an array that we create, such as

let books = [

We paste our data into the array and then contain the array using];

We then save the file as books.js

Spreadsheets are useful to help us to create massive worlds that are well organized.

Happy Scripting :-)

Here is the Speadsheet Syntax Explained

{ title: `	C Computer Science	`, url: `	media/pdf/C Computer Science by Christopher Topalian.pdf

In the first cell we write

{ title: `

In the second cell we write

C Computer Science

In the third cell we write

`, url: `

In the fourth cell we write

media/pdf/C Computer Science by Christopher Topalian.pdf

If this was the end of the object, we would then end the object by using

```
` },
```

In this way, at the end of each row, we always make sure to contain each object.

A Spreadsheet is actually an Array with each row being an Object in that Array.
A Spreadsheet is an Array of Objects.

Spreadsheets are an orderly way to organize our data and offer us many tools to filter our data and then convert it to a JS format that is useable offline and online :-)

Happy Scripting :-)

Christopher Topalian

How to Combine .js files into one main.js file using Command Prompt

Version for when we have js scripts in subfolders in our js project folder, that we want to combine.

College of Scripting Music & Science

CollegeOfScripting.weebly.com

// HowToCombineJSFiles.js
TUTORIAL:
How to Combine all .js files in all folders that are in our js folder.

Getting things ready:
We should add two new lines at the end every script. This way they will combine nicely with a line break in between each script.

Step One: Open our js folder

Step Two: Type in the address bar of the js folder, cmd, press Enter

This opens our js folder in the command prompt

Step Three: Type the command shown below in the command prompt and then press Enter

```
for /r "%CD%" %i in (*.js) do type "%i" >> main.js
```

Now we have a newly created .js file named main.js that has all of our js files included into one file.

This makes it easy to upload our application and easy to find out how many lines of code our project is.

To use our main.js file, we include it in our html file code:

```
<script src = 'js/main.js'></script>
```

Happy Scripting :-)

How to Combine .js files into one main.js file using a **batch file**

Version for when we have js scripts in subfolders in our js project folder, that we want to combine.

// HowToCombineJSFilesUsingBatFile.js

We can combine all of the .js files
that are located in our js folder
into one main.js file, using either:

  The Command Prompt Method
  or
  The .bat File Method

The .bat file method is very easy.
We double click the bat file, which is located in our js folder, and it will make a main.js file, which includes all .js files in the js folder, including all .js files in all subdirectories of our js folder.

This is a very easy way to combine our .js files, because we can double click the .bat file anytime, and it will again generate the main.js file, which includes all .js files in the js folder, including all .js files in all subdirectories of our js folder. This makes uploading our application online much easier.

Happy Scripting :-)

```
:: Topalian_Combine_JS_Files.bat

:: This .bat File Combines All .js files in all folders of our project folder, into one main.js file.

:: To activate this .bat file, we double click the .bat file, while it is located in our js folder.

@echo off
:: set the output file name
set "output=main.js"

:: clear existing output file
type nul > "%output%"

:: loop through all JavaScript files in subdirectories
for /r %%i in (*.js) do (
    :: append the content of each file to the
output file
    type "%%i" >> "%output%"
)

echo "JavaScript files combined into
%output% successfully."
```

Christopher Topalian

How to Combine .js files into one main.js file using Node.js

This version will successfully combine a single folder of js files.
It also works to combine js files in all subdirectories.

```
// Topalian_Combine_JS_Files.js

let fs = require('fs');
let path = require('path');

function combineJSFiles(directory,
scriptFilename)
{
    let outputFilePath = path.join(directory,
'main.js');

    let fileContents = [];

    function traverseFolder(folder)
    {
        let files = fs.readdirSync(folder);

        for (let i = 0; i < files.length; i++)
        {
            let file = files[i];

            let filePath = path.join(folder, file);

            let stats = fs.statSync(filePath);

            if (stats.isDirectory())
            {
                traverseFolder(filePath);
            }
            else if (path.extname(filePath) === '.js')
            {
                let content =
fs.readFileSync(filePath, 'utf8');
                // check if file is not script file itself
                if (filePath !== scriptFilename)
                {
                    fileContents.push(content);
                }
            }
        }
    }

    traverseFolder(directory);

    fs.writeFileSync(outputFilePath, 
fileContents.join('\n'), 'utf8');

    console.log(`Combined $
{fileContents.length} .js files into $
{outputFilePath}`);
}

// get current directory of script
let currentDirectory = process.cwd();
```

```
// get filename of script
let scriptFilename = __filename;

combineJSFiles(currentDirectory,
scriptFilename);
```

How to Combine
.js files
into one
main.js file
using
Python

This version will successfully combine a single folder of js files.
It also works to combine js files in all subdirectories.

```
# Topalian_Combine_JS_Files.py

import os

def combineJSFiles(directory,
scriptFileName):
    outputFilePath = os.path.join(directory,
'main.js')
    fileContents = []

    def traverseFolder(folder):
        for root, dirs, files in os.walk(folder):
            for file in files:
                filePath = os.path.join(root, file)
                if filePath != scriptFileName and
filePath.endswith('.js'):
                    with open(filePath, 'r',
encoding='utf-8') as f:
```

```
                        fileContents.append(f.read())

    traverseFolder(directory)

    with open(outputFilePath, 'w',
encoding='utf-8') as f:
        f.write('\n'.join(fileContents))
    print(f"Combined {len(fileContents)} .js
files into {outputFilePath}")

# get current directory of script
currentDirectory =
os.path.dirname(os.path.abspath(__file__))

# get filename of script
scriptFileName = os.path.abspath(__file__)
```

combineJSFiles(currentDirectory, scriptFileName)

What other file types can we combine?

We have combined .js files in this book, but we might choose to instead combine:

.py or .html or .txt

This is very useful for book making.

In each of the scripts shown in this book, we can manually change the parts where it says .js, with .py, if we wanted to, for instance, copy all .py files into one main.py file.

We can do the same thing for .html files, where we change the file type it will be combining to .html and it will combine all .html files into one main.html file.

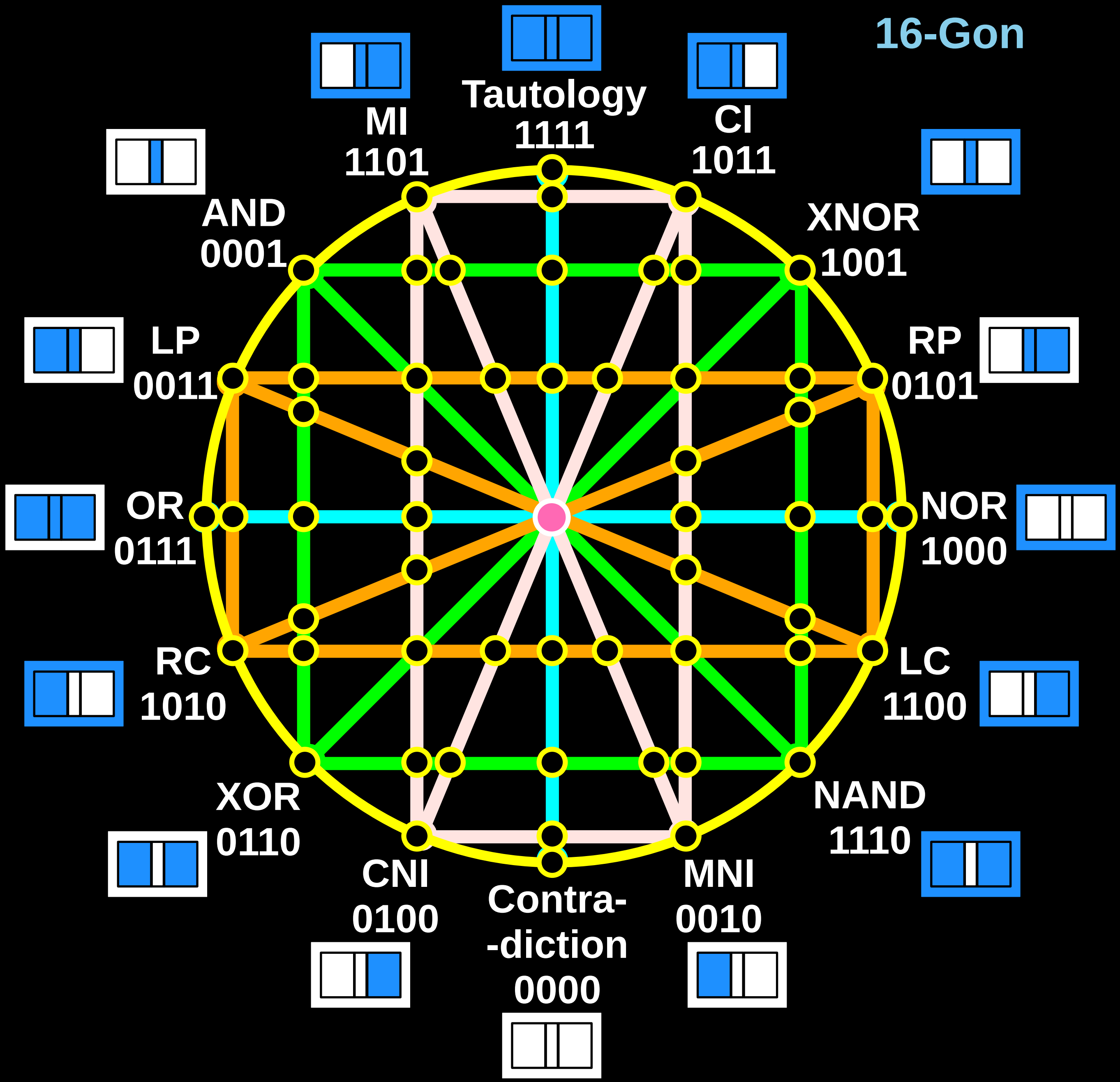
We add two line breaks at the end of all files, so that there is space between files, when they are combined.

Remember too, that not all file types will combine, but the ones above will.

The original files are not changed. The content from the original files is only copied from.

Christopher Topalian

True Artificial Intelligence System



College of Scripting Music & Science

CollegeOfScripting.weebly.com

For More Tutorials:

GitHub.com/ChristopherTopalian

GitHub.com/ChristopherAndrewTopalian

Sites.google.com/view/CollegeOfScripting

CollegeOfScripting.weebly.com

CollegeOfScripting.wordpress.com

Youtube.com/ScriptingCollege

Twitter.com/CollegeOfScript

Rumble.com/user/CollegeOfScripting

Dedicated to God the Father

This book is created by the
College of Scripting Music & Science.
Always remember, that each time you write a script with a pencil and paper, it becomes imprinted so deeply in memory that the material and methods are learned extremely well.

When you Type the scripts, the same is true. The more you type and write out the scripts by keyboard or pencil and paper, the more you will learn programming!

Write and Type every example that you find.
Keep all of your scripts organized.
Every script that you create increases your programming abilities.
SEEING CODE, is one thing,
but WRITING CODE is another.
Write it, Type it, Speak it, See it, Dream it.

CollegeOfScripting.weebly.com