

# **Topalian Combine JavaScript Files**

**by**

**Christopher Andrew Topalian**

All Rights Reserved  
Copyright 2000-2024

# Dedicated to God the Father

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

**Version for when we have  
only ONE folder of .js files  
that we want to combine.**

## // HowToCombineJSFilesOneFolder.js

First, we add two new lines at the end of every script. This way they will later combine nicely with a line break in between each script.

We open our js folder.

In our js project folder, we type  
cmd

in the address bar of the folder and then  
press enter

This opens our js folder in the Command  
prompt

We type in the words  
copy \*.js main.js  
and then press enter

This creates a new file that is named main.js  
This new file contains all .js files in ONE file.

But, there is a junk character at the end of the  
main.js script that we have to delete. In  
VSCode the character might be called SUB

```
titleContainer.append(titleOfApp);  
}
```

SUB

We remove this junk SUB character and the code will now run.

```
titleContainer.append(titleOfApp);  
}
```

As we can see, the junk character is removed.

<!-- Now, in our html code, we can type -->  
<script src = 'main.js'></script>

This makes it much easier to upload our js code to our website.

# **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.**

## // 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."

# **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)
```

## How to Paste Code from a PDF that has Junk Characters.

When we paste from this pdf into VSCode, it might look like this:

```
function combineJSFiles(directory,  
scriptFilename)  
{  
  let outputPath = path.join  
(directory, 'main.js');  
  
  let fileContents = [];
```

We can't leave those junk characters in the code, so we remove them with find/replace.

We Find 1 of the spaces.

We Replace All with the 1 space that we typed.

This gets rid of the junk characters in the code.

The next page shows how to do this step by step.

We highlight 1 space with our mouse arrow:

```
function combineJSFiles(directory,
scriptFilename)
```

```
{
```

```
let outputPath = path.join
(directory, 'main.js');
```

```
let fileContents = [];
```

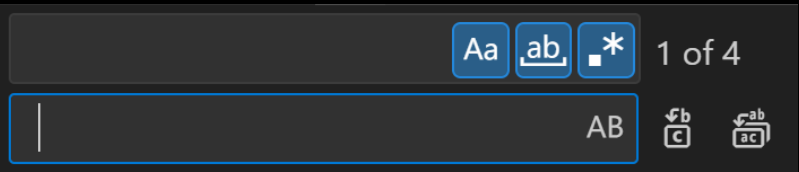
We press Control + H to open the Find/Replace feature and Replace All with our own Space

```
function comb
scriptFilename)
```

```
{
```

```
let outputPath = path.join
(directory, 'main.js');
```

```
let fileContents = [];
```



Here we see that the Find/Replace All has replaced the junk characters with our working spaces instead:

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

Now that the code  
has no junk characters,  
it can run.

## **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.**

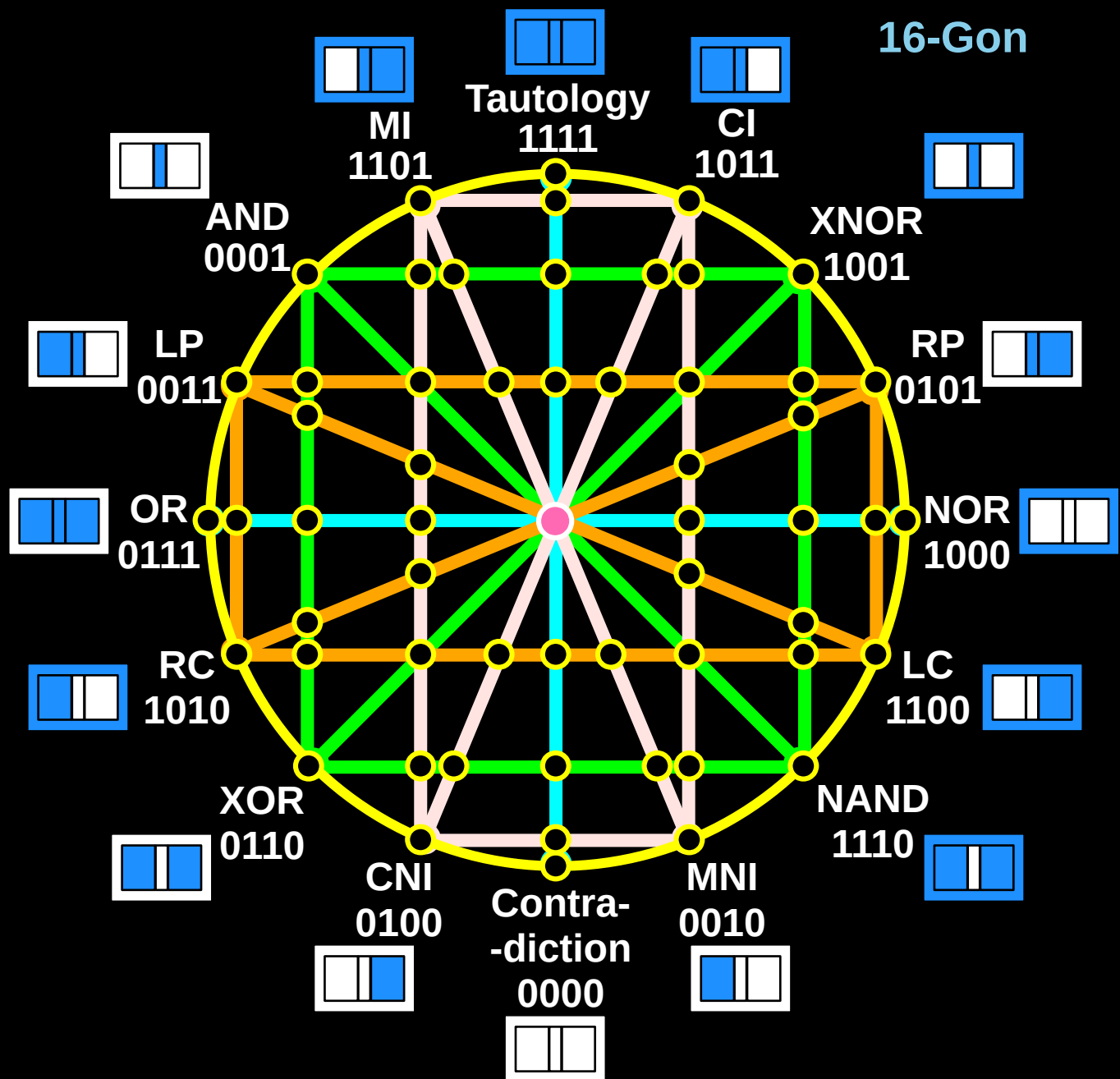
**We can do the same thing for .html files, where we change the file type it will be combining to .html and it will combine all .html files into one main.html file.**

**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.**

# True Artificial Intelligence System



# For More Tutorials:

[\*\*GitHub.com/ChristopherTopalian\*\*](https://github.com/ChristopherTopalian)

[\*\*GitHub.com/ChristopherAndrewTopalian\*\*](https://github.com/ChristopherAndrewTopalian)

[\*\*Sites.google.com/view/CollegeOfScripting\*\*](https://sites.google.com/view/CollegeOfScripting)

[\*\*CollegeOfScripting.weebly.com\*\*](http://CollegeOfScripting.weebly.com)

[\*\*CollegeOfScripting.wordpress.com\*\*](http://CollegeOfScripting.wordpress.com)

[\*\*Youtube.com/ScriptingCollege\*\*](https://youtube.com/ScriptingCollege)

[\*\*Twitter.com/CollegeOfScript\*\*](https://twitter.com/CollegeOfScript)

[\*\*Rumble.com/user/CollegeOfScripting\*\*](https://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**