

Topalian JavaScript Text Investigator

**by
Christopher Andrew Topalian**

All Rights Reserved
Copyright 2000-2024

Dedicated to God the Father

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

Date: Fri Jul 12 2024 4:20:26 AM, Magnitude: 4.8, Location: 203 km SW of Port McNeill, Canada

Earthquake (latest) Earthquake (All) Get Random Joke

Word Count Word Frequency Pattern Recognition Average Word Length Sentiment Rating Extract Keywords

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

Date: Fri Jul 12 2024 4:20:26 AM, Magnitude: 4.8, Location: 203 km SW of Port McNeill, Canada

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

Date: Fri Jul 12 2024 4:20:26 AM, Magnitude: 4.8, Location: 203 km SW of Port McNeill, Canada

Date: Fri Jul 12 2024 1:47:02 AM, Magnitude: 4.6, Location: central East Pacific Rise

Date: Fri Jul 12 2024 12:53:52 AM, Magnitude: 5.1, Location: Komandorskiye Ostrova, Russia region

Date: Thu Jul 11 2024 11:55:54 PM, Magnitude: 5, Location: 113 km NNE of Hihifo, Tonga

Earthquake (latest) Earthquake (All) Get Random Joke

Word Count Word Frequency Get all Earthquake data from the internet Page Word Length Sentiment Rating Extract Keywords

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

[object Promise]

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

What goes after USA? USB.

Earthquake (latest) Earthquake (All) **Get Random Joke**

Word Count Word Frequency Pattern Recogn Get a random joke from the internet Sentiment Rating Extract Keywords

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

undefined

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/Topalian_Je ★ 100%

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

Word Count Word Frequency Pattern Recognition Average Word Length Sentiment Rating Extract Keywords

Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

Count the number of words

7

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/Topalian_Je

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

Word Count Word Frequency Pattern Recognition Average Word Length Sentiment Rating Extract Keywords

Capitalize Text Lowercase Text Count the frequency of each word Capitalize First Letter Bold Text Normal Text Italic Text Underline Text

we: 1
can: 1
type: 1
or: 1
paste: 1
text: 1
here: 1

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

Word Count Word Frequency **Pattern Recognition** Average Word Length Sentiment Rating Extract Keywords

Capitalize Text Lowercase Text Capitalize 1st Letter **Search for Pattern of word THE** Normal Text Italic Text Underline Text Fo

0

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

Word Count Word Frequency Pattern Recognition **Average Word Length** Sentiment Rating Extract Keywords

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

Average length of words

3.43

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

Word Count Word Frequency Pattern Recognition Average Word Length **Sentiment Rating** Extract Keywords

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text *Italic Text* **The general intention of the text** Fo

Neutral

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating **Extract Keywords** Extract Keywords Uppercase Titlecase Keywords Categorize Words

Capitalize Text Lowercase Text Get Keywords without getting common link words Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

we, type, paste, text

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Uppercase extracted keywords Italic Text Underline Text Fo

We, Type, Paste, Text

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase **Titlecase Keywords** Categorize Words

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

We can Type or Paste Text here

Topalian JavaScript Text Investigator × +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

Length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Put words into categories

we = n/a
can = n/a
type = n/a
or = connector
paste = n/a
text = n/a
here = n/a

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Capitalize Text Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

Capitalizes all words

WE CAN TYPE OR PASTE TEXT HERE

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Capitalize Text **Lowercase Text** Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Fo

Lowercases all words

we can type or paste text here

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Capitalize Text Lowercase Text **Capitalize 1st Letter** Bold Text Normal Text Italic Text Underline Text Fo

Capitalizes the 1st Letter of each word

We Can Type Or Paste Text Here

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Capitalize Text Lowercase Text Capitalize 1st Letter **Bold Text** Normal Text Italic Text Underline Text Fo

Bold all words

We can type or paste text here

Topalian JavaScript Text Investigator

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Format as Bulleted List

Normalize all words

We can type or paste text here

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Lowercase Text Capitalize 1st Letter Bold Text Normal Text **Italic Text** Underline Text Format as Bulleted List

Italicize all words

We can type or paste text here

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★ 100%

Topalian JavaScript Text Investigator

We can type or paste text here

Earthquake (latest) Earthquake (All) Get Random Joke

length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

Lowercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text Format as Bulleted List

Underline all words

We can type or paste text here

Topalian JavaScript Text Investigator x +

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Text_Investigator/To ★

100%

We can type or paste text here

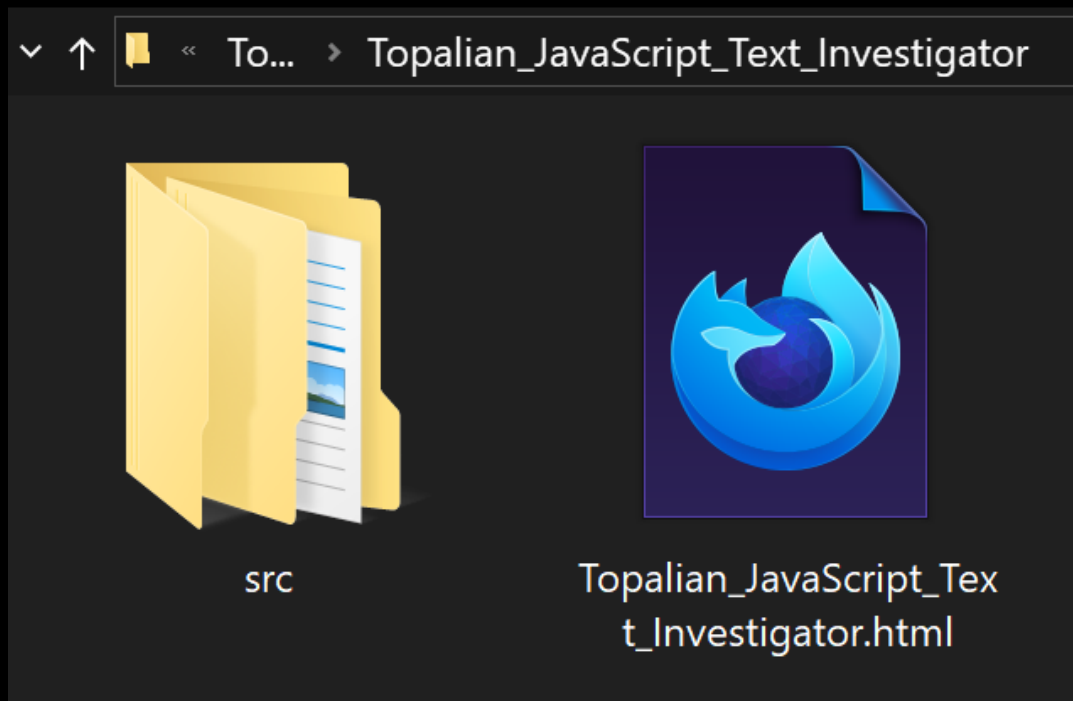
Earthquake (latest) Earthquake (All) Get Random Joke

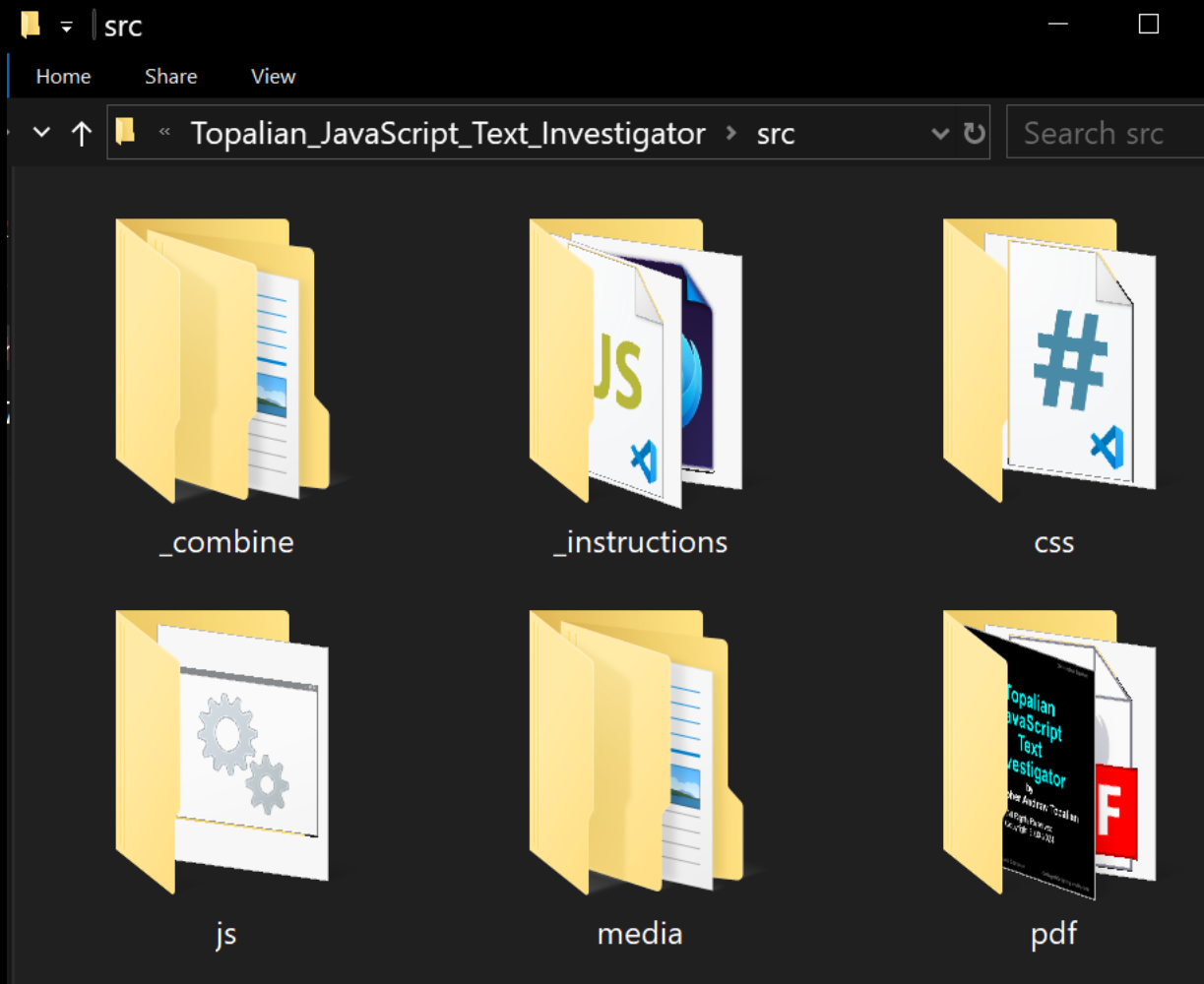
Length Sentiment Rating Extract Keywords Extract Keywords Uppercase Titlecase Keywords Categorize Words

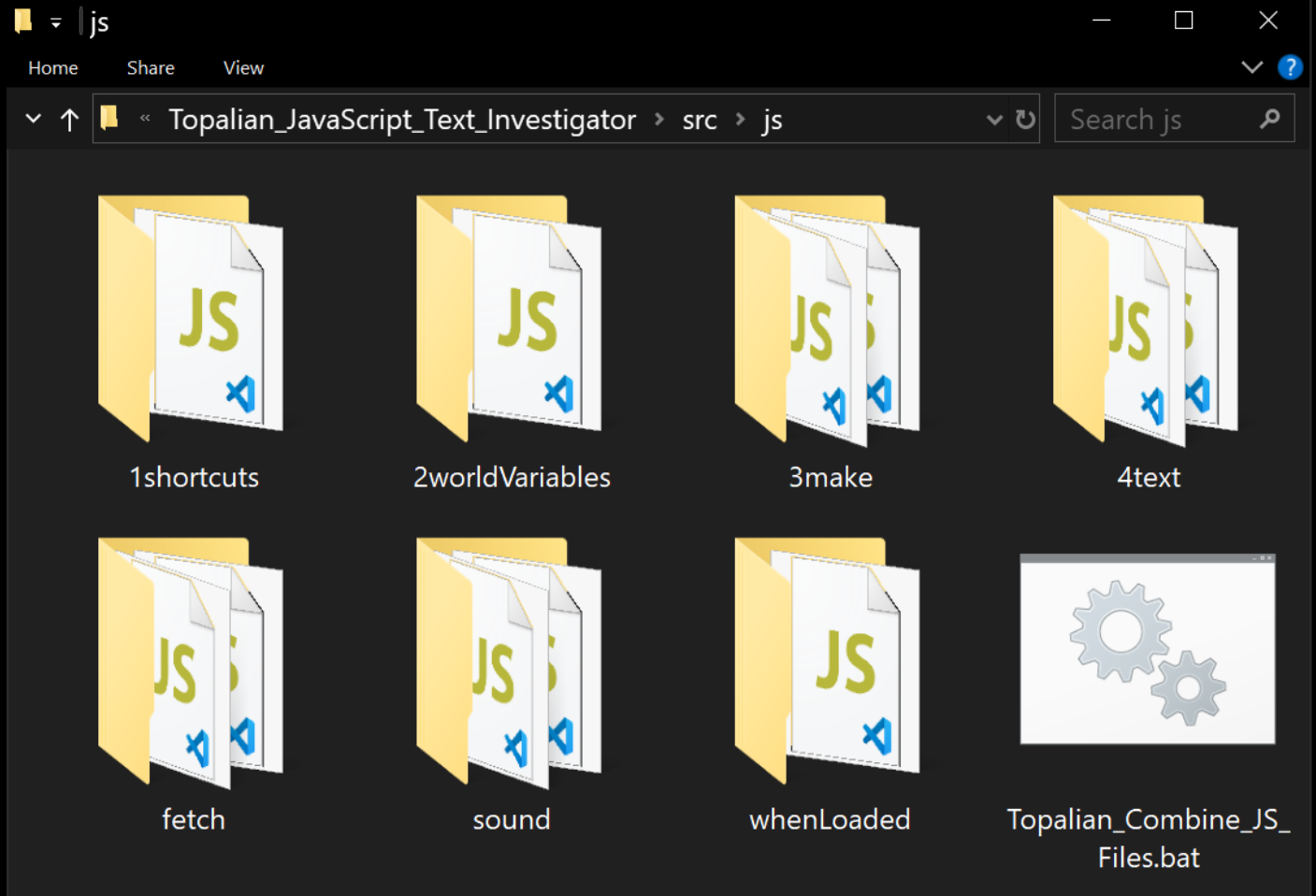
Uppercase Text Capitalize 1st Letter Bold Text Normal Text Italic Text Underline Text **Format as Bulleted List**

Format words as Bullet List

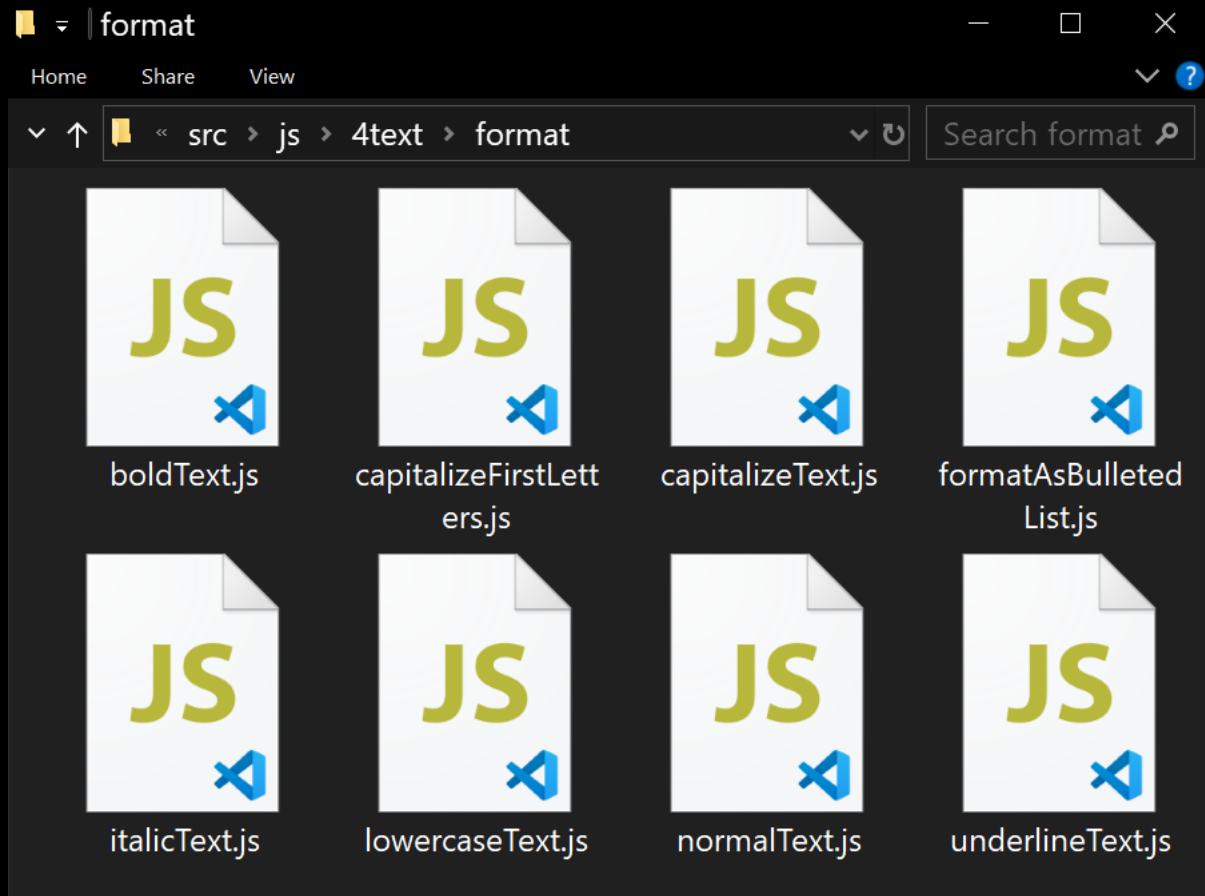
- * We
- * Can
- * Type
- * Or
- * Paste
- * Text
- * Here











```
<!-- Dedicated to God the Father -->
<!-- All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024 -->
<!-- https://github.com/ChristopherTopalian --
>
<!--
https://github.com/ChristopherAndrewTopalia
n -->
<!-- https://github.com/ChristopherTopalian/ --
>
<!--
Topalian_JavaScript_Text_Investigator.html --
>
<!-- Version 001 - (2024-07-12) -->

<html>
<head>
```

```
<title> Topalian JavaScript Text Investigator  
</title>
```

```
<link rel = 'stylesheet' href =  
'src/css/style001.css'>
```

```
<!-- shortcuts -->
```

```
<script src =  
'src/js/1shortcuts/shortcuts.js'></script>
```

```
<!-- world variables -->
```

```
<script src =  
'src/js/2worldVariables/worldVariables.js'></s  
cript>
```

```
<!-- make -->
```

```
<script src =  
'src/js/3make/makeInterface.js'></script>
```

```
<script src =  
'src/js/3make/makeTitleOfApp.js'></script>
```

```
<!-- text -->
```

```
<script src =  
'src/js/4text/detectPattern.js'></script>
```

```
<script src =  
'src/js/4text/getWordCount.js'></script>
```

```
<script src =  
'src/js/4text/calculateAvgWordLength.js'></sc  
ript>
```

```
<script src =  
'src/js/4text/getWordFrequency.js'></script>
```

```
<script src =  
'src/js/4text/detectSentiment.js'></script>
```

```
<script src =  
'src/js/4text/extractKeywords.js'></script>
```

```
<script src =  
'src/js/4text/categorizeWords.js'></script>
```

```
<script src =  
'src/js/4text/extractKeywordsUppercase.js'></  
script>
```

```
<script src =  
'src/js/4text/titleCaseKeywords.js'></script>
```

```
<!-- text format -->
```

```
<script src =  
'src/js/4text/format/capitalizeText.js'></script>
```

```
<script src =  
'src/js/4text/format/lowercaseText.js'></script  
>
```

```
<script src =  
'src/js/4text/format/capitalizeFirstLetters.js'><  
/script>
```

```
<script src =  
'src/js/4text/format/boldText.js'></script>
```

```
<script src =  
'src/js/4text/format/normalText.js'></script>
```



```
<script src =  
'src/js/4text/format/italicText.js'></script>
```

```
<script src =  
'src/js/4text/format/underlineText.js'></script>
```

```
<script src =  
'src/js/4text/format/formatAsBulletedList.js'><  
/script>
```

```
<!-- fetch -->  
<script src =  
'src/js/fetch/fetchRandomJoke.js'></script>
```

```
<script src =  
'src/js/fetch/fetchMostRecentEarthquakeData.  
js'></script>
```

```
<script src =  
'src/js/fetch/fetchAllEarthquakeData.js'></scri  
pt>
```

```
<!-- sound -->  
<script src =  
'src/js/sound/sounds.js'></script>
```

```
<script src =  
'src/js/sound/loadSounds.js'></script>
```

```
<script src =  
'src/js/sound/audioPlay.js'></script>
```

```
<!-- when loaded -->  
<script src =  
'src/js/whenLoaded/whenLoaded.js'></script>
```

```
</head>
```

```
<body onload = 'whenLoaded();'>
```

```
</body>
```

```
</html>
```

```
/* Dedicated to God the Father */  
/* All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024 */  
/* https://github.com/ChristopherTopalian */  
/*  
https://github.com/ChristopherAndrewTopalia  
n */  
/* style001.css */
```

```
body  
{  
    background-color: rgb(30, 30, 30);  
    font-family: Arial;  
    font-size: 20px;  
    color: rgb(255, 255, 255);  
}
```

textarea

```
{  
  padding-left: 10px;  
  padding-right: 10px;  
  padding-top: 10px;  
  padding-bottom: 10px;  
  border: solid 1px rgb(0, 0, 0);  
  border-radius: 8px;  
  background-color: rgb(0, 0, 0);  
  font-family: Arial;  
  font-size: 20px;  
  color: rgb(255, 255, 255);  
}
```

```
a  
{  
  color: rgb(100, 100, 100);  
  text-decoration: none;  
}
```

```
a:hover  
{  
  text-decoration: underline;  
}
```

```
#mainContainer  
{  
  width: 95%;  
  margin: auto;  
}
```

```
#textInput  
{  
  width: 100%;  
  height: 150px;  
}
```

#resultContainer

```
{  
  margin-top: 5px;  
  padding-left: 10px;  
  padding-right: 10px;  
  padding-top: 10px;  
  padding-bottom: 10px;  
  border: solid 1px rgb(0, 0, 0);  
  border-radius: 8px;  
  background-color: rgb(0, 0, 0);  
  font-family: Arial;  
  font-size: 20px;  
  color: rgb(255, 255, 255);  
}
```

.buttonStyle001

```
{  
  max-height: 40px;
```

```
margin: 2px;  
padding-left: 10px;  
padding-right: 10px;  
padding-top: 2px;  
padding-bottom: 2px;  
border: solid 1px rgb(100, 100, 100);  
border-radius: 8px;  
background-color: rgb(0, 0, 0);  
font-size: 15px;  
color: rgb(255, 255, 255);  
cursor: pointer;  
box-sizing: border-box;  
}
```

```
.buttonStyle001:hover  
{  
    border-color: rgb(0, 255, 255);  
}
```



```
.buttonStyle001:active
{
    position: relative;
    top: 1px;
    border-color: rgb(255, 0, 255);
    color: rgb(255, 0, 255);
}

* {
    scrollbar-width: thin;
    scrollbar-color: rgb(100, 100, 100) rgb(0, 0, 0);
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// shortcuts.js
```

```
function ge(whichId)
{
    let result =
document.getElementById(whichId);
    return result;
}
```

```
function ce(whichType)
{
```

```
    let result =  
document.createElement(whichType);  
    return result;  
}
```

```
function ba(whichElement)  
{  
    let result =  
document.body.append(whichElement);  
    return result;  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// worldVariables.js
```

```
let online = false;
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// makeInterface.js
```

```
function makeInterface()
{
  // mainContainer
  let mainContainer = ce('div');
  mainContainer.id = 'mainContainer';
  mainContainer.style.position = 'absolute';
  mainContainer.style.left = '20px';
  mainContainer.style.top = '25px';
  ba(mainContainer);
}
```

```
//-//
```

```
// textInput
```

```
let textInput = ce('textarea');  
textInput.id = 'textInput';  
textInput.placeholder = 'Enter words  
here...';  
mainContainer.append(textInput);
```

```
//-//
```

```
// btnDivInternetData
```

```
let btnDivInternetData = ce('div');  
btnDivInternetData.style.display = 'flex';  
btnDivInternetData.style.flexDirection =  
'row';  
btnDivInternetData.style.overflowX = 'auto';
```

```
btnDivInternetData.style.whiteSpace =  
'nowrap';
```

```
// horizontal scrolling mouse wheel,  
without needing to hold shift button
```

```
btnDivInternetData.addEventListener('wheel',  
function(event)  
{  
    if (event.deltaY !== 0)  
    {  
        btnDivInternetData.scrollLeft +=  
event.deltaY;  
        event.preventDefault();  
    }  
});
```

```
mainContainer.append(btnDivInternetData);
```

```
//-//
```

```
let internetDataBtnArray =  
[  
  {  
    name: 'Earthquake (latest)',  
    functionName:  
    `fetchMostRecentEarthquakeData()`,  
    description: 'Get most recent  
Earthquake data from the internet'  
  },  
  
  {  
    name: 'Earthquake (All)',
```



```
functionName:  
`fetchAllEarthquakeData()`,  
  description: 'Get all Earthquake data  
from the internet'  
},  
  
{  
  name: 'Get Random Joke',  
  functionName: `fetchRandomJoke()`,  
  description: 'Get a random joke from  
the internet'  
}  
];  
  
for (let i = 0; i <  
internetDataBtnArray.length; i++)  
{  
  let theButton = ce('button');
```

```
theButton.className = 'buttonStyle001';
theButton.innerHTML =
internetDataBtnArray[i].name;
theButton.title =
internetDataBtnArray[i].description;
theButton.onmouseover = function()
{
    audioPlay('sfx_warp_001', 1.0);
};

theButton.onclick = function(buttonInfo)
{
    audioPlay('sfx_blip_001', 1.0);

    ge('resultContainer').innerHTML =
eval(internetDataBtnArray[i].functionName);
}
```

```
    btnDivInternetData.append(theButton);  
}
```

```
//-//
```

```
// btnDivAnalyzeText
```

```
let btnDivAnalyzeText = ce('div');  
btnDivAnalyzeText.style.display = 'flex';  
btnDivAnalyzeText.style.flexDirection =  
'row';  
    btnDivAnalyzeText.style.overflowX = 'auto';  
    btnDivAnalyzeText.style.whiteSpace =  
'nowrap';
```

```
// horizontal scrolling mouse wheel,  
without needing to hold shift button
```

```
btnDivAnalyzeText.addEventListener('wheel',  
function(event)  
{  
    if (event.deltaY !== 0)  
    {  
        btnDivAnalyzeText.scrollLeft +=  
event.deltaY;  
        event.preventDefault();  
    }  
});
```

```
mainContainer.append(btnDivAnalyzeText);
```

```
//-//
```

```
let analyzeButtons =
```

```
[  
  {  
    name: 'Word Count',  
    functionName:  
    `getWordCount(textInput.value)`,  
    description: 'Count the number of  
words'  
  },  
  
  {  
    name: 'Word Frequency',  
    functionName:  
    `getWordFrequency(textInput.value)`,  
    description: 'Count the frequency of  
each word'  
  },  
  
  {
```

```
    name: 'Pattern Recognition',  
    functionName:  
    `detectPattern(textInput.value)`,  
    description: 'Search for Pattern of  
word THE'  
  },  
  
  {  
    name: 'Average Word Length',  
    functionName:  
    `calculateAvgWordLength(textInput.value)`,  
    description: 'Average length of words'  
  },  
  
  {  
    name: 'Sentiment Rating',  
    functionName:  
    `detectSentiment(textInput.value)`,
```

```
        description: 'The general intention of  
the text'  
    },  
  
    {  
        name: 'Extract Keywords',  
        functionName:  
        `extractKeywords(textInput.value)`,  
        description: 'Get Keywords without  
getting common link words'  
    },  
  
    {  
        name: 'Extract Keywords Uppercase',  
        functionName:  
        `extractKeywordsUppercase(textInput.value)`  
    },  
    ,
```

```
        description: 'Uppercase extracted  
keywords'  
    },  
  
    {  
        name: 'Titlecase Keywords',  
        functionName:  
        `titleCaseKeywords(textInput.value)`,  
        description: 'Titlecases the words'  
    },  
  
    {  
        name: 'Categorize Words',  
        functionName:  
        `categorizeWords(textInput.value)`,  
        description: 'Put words into  
categories'  
    },
```



```
];  
  
for (let i = 0; i < analyzeButtons.length; i++)  
{  
    let theButton = ce('button');  
    theButton.className = 'buttonStyle001';  
    theButton.innerHTML =  
analyzeButtons[i].name;  
    theButton.title =  
analyzeButtons[i].description;  
    theButton.onmouseover = function()  
    {  
        audioPlay('sfx_warp_001', 1.0);  
    };  
  
    theButton.onclick = function(buttonInfo)  
    {
```

```
    audioPlay('sfx_blip_001', 1.0);

    ge('resultContainer').innerHTML =
eval(analyzeButtons[i].functionName);
    }

    btnDivAnalyzeText.append(theButton);
}

// - //

// btnDivFormatting
let btnDivFormatting = ce('div');
btnDivFormatting.style.display = 'flex';
btnDivFormatting.style.flexDirection =
'row';
btnDivFormatting.style.overflowX = 'auto';
```

```
btnDivFormatting.style.whiteSpace =  
'nowrap';
```

```
// horizontal scrolling mouse wheel,  
without needing to hold shift button
```

```
btnDivFormatting.addEventListener('wheel',  
function(event)  
{  
    if (event.deltaY !== 0)  
    {  
        btnDivFormatting.scrollLeft +=  
event.deltaY;  
        event.preventDefault();  
    }  
});  
mainContainer.append(btnDivFormatting);
```

//-//

```
let formattingButtons =  
[  
  {  
    name: 'Capitalize Text',  
    functionName:  
    `capitalizeText(textInput.value)`,  
    description: 'Capitalizes all words'  
  },  
  
  {  
    name: 'Lowercase Text',  
    functionName:  
    `lowercaseText(textInput.value)`,  
    description: 'Lowercases all words'  
  },  
]
```

```
{  
  name: 'Capitalize 1st Letter',  
  functionName:  
  `capitalizeFirstLetters(textInput.value)`,  
  description: 'Capitalizes the 1st Letter  
of each word'  
},  
  
{  
  name: 'Bold Text',  
  functionName:  
  `boldText(textInput.value)`,  
  description: 'Bold all words'  
},  
  
{  
  name: 'Normal Text',
```

```
functionName:  
`normalText(textInput.value)`,  
description: 'Normalize all words'  
},  
  
{  
  name: 'Italic Text',  
  functionName:  
`italicText(textInput.value)`,  
description: 'Italicize all words'  
},  
  
{  
  name: 'Underline Text',  
  functionName:  
`underlineText(textInput.value)`,  
description: 'Underline all words'  
},
```

```
{  
  name: 'Format as Bulleted List',  
  functionName:  
  `formatAsBulletedList(textInput.value)`,  
  description: 'Format words as Bullet  
List'  
}  
];  
  
for (let i = 0; i < formattingButtons.length;  
i++)  
{  
  let theButton = ce('button');  
  theButton.className = 'buttonStyle001';  
  theButton.innerHTML =  
formattingButtons[i].name;
```

```
theButton.title =  
formattingButtons[i].description;  
theButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
  
theButton.onclick = function(buttonInfo)  
{  
    audioPlay('sfx_blip_001', 1.0);  
  
    ge('resultContainer').innerHTML =  
eval(formattingButtons[i].functionName);  
}  
  
btnDivFormatting.append(theButton);  
}
```


//-//

```
let resultContainer = ce('div');  
resultContainer.id = 'resultContainer';  
resultContainer.style.minHeight = '100px';  
mainContainer.append(resultContainer);  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// makeTitleOfApp.js
```

```
function makeTitleOfApp()
{
  // titleContainer
  let titleContainer = ce('div');
  titleContainer.style.position = 'absolute';
  titleContainer.style.right = '20px';
  titleContainer.style.top =
titleContainer.style.top = '2px';
  titleContainer.style.zIndex = 1;
```

```
ba(titleContainer);
```

```
//-//
```

```
// titleOfApp
```

```
let titleOfApp = ce('div');
```

```
titleOfApp.id = 'titleOfApp';
```

```
titleOfApp.innerHTML =
```

```
`<a href =
```

```
'https://github.com/christophertopalian/topali  
an_javascript_text_investigator' target =
```

```
'_blank'> Topalian JavaScript Text  
Investigator </a>`;
```

```
titleOfApp.style.fontSize = '17px';
```

```
titleOfApp.style.fontWeight = 'bold';
```

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

```
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// calculateAvgWordLength.js
```

```
function calculateAvgWordLength(whichText)
{
    let words = whichText.split(/\s+/);
    let totalLength = 0;
    let wordCount = 0;

    for (let i = 0; i < words.length; i++)
    {
        let word = words[i];
```

```
    if (word.length > 0)
    {
        totalLength += word.length;
        wordCount++;
    }
}

if (wordCount > 0)
{
    return (totalLength /
wordCount).toFixed(2);
}
else
{
    return 0;
}
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
//categorizeWords.js
```

```
function categorizeWords(whichText)
{
    // define word categories
    let categories =
    {
        specifier: ["the", "a", "an", "those",
"that", "them", "they", "this"],
        connector: ["and", "but", "or",
"however", "therefore"],
```

```
    thing: ["chair", "table", "car"],  
    time: ["early", "late", "morning",  
"evening", "yesterday", "today", "tomorrow",  
"now", "then", "when"]  
};
```

```
// convert whichText to lowercase and split  
it into words
```

```
let words = whichText.toLowerCase().split('');
```

```
// array to hold categorized words
```

```
let categorizedWords = [];
```

```
// loop through words in whichText
```

```
for (let i = 0; i < words.length; i++)
```

```
{
```

```
    let word = words[i];
```

```
let categoryFound = false;

// check each category
let categoryKeys =
Object.keys(categories);

for (let k = 0; k < categoryKeys.length; k++)
{
    let category = categoryKeys[k];

    let categoryWords =
categories[category];

    for (let j = 0; j < categoryWords.length;
j++)
    {
        if (word === categoryWords[j])
```



```
        {
            categorizedWords.push(word + "
= " + category);

            categoryFound = true;
            break;
        }
    }

    if (categoryFound)
    {
        break;
    }
}

// if no category found, label as n/a
if (!categoryFound)
{
```

```
        categorizedWords.push(word + " =  
n/a");  
    }  
}  
  
return categorizedWords.join("<br> ");  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// detectPattern.js
```

```
function detectPattern(whichText)
{
    let pattern = /the/gi;

    let matches = whichText.match(pattern);

    let matchCount;

    if (matches)
```

```
{  
    matchCount = matches.length;  
}  
else  
{  
    matchCount = 0;  
}  
  
return matchCount;  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// detectSentiment.js
```

```
function detectSentiment(whichText)
{
    // positive words
    let positiveWords = ["happy", "joy", "love",
"excellent", "good", "great", "fantastic",
"positive", "nice", "wonderful"];

    // negative words
```

```
let negativeWords = ["sad", "anger",  
"hate", "terrible", "bad", "horrible",  
"negative", "mean", "evil"];
```

```
// initialize sentiment scores
```

```
let positiveScore = 0;
```

```
let negativeScore = 0;
```

```
// convert whichText to lowercase and split  
it into words
```

```
let words =  
whichText.toLowerCase().split(/\W+/);
```

```
// loop through words in whichText
```

```
for (let i = 0; i < words.length; i++)
```

```
{
```

```
    let word = words[i];
```

```
// is word in list of pos words
for (let j = 0; j < positiveWords.length; j+
+)
{
    if (word === positiveWords[j])
    {
        positiveScore++;
        // exit inner loop if match found
        break;
    }
}
```

```
// is word in list of neg words
for (let k = 0; k < negativeWords.length;
k++)
{
    if (word === negativeWords[k])
    {
```

```
        negativeScore++;  
        // exit inner loop if match found  
        break;  
    }  
}  
}
```

```
let sentiment;
```

```
// determine sentiment  
if (positiveScore > negativeScore)  
{  
    sentiment = "Positive";  
}  
else if (negativeScore > positiveScore)  
{  
    sentiment = "Negative";  
}
```



```
else  
{  
    sentiment = "Neutral";  
}  
  
return sentiment;  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// extractKeywords.js
```

```
function extractKeywords(whichText)
{
    // array of common stop words to ignore
    let stopWords = ["a", "an", "and", "the",
    "in", "on", "at", "of", "to", "is", "it", "that",
    "this", "with", "for", "as", "by", "but", "or",
    "nor", "up", "down", "out", "over", "under",
    "again", "further", "then", "once", "here",
    "there", "when", "where", "why", "how", "all",
```

```
"any", "both", "each", "few", "more", "most",  
"other", "some", "such", "no", "nor", "too",  
"very", "can", "will", "just", "should", "now"];
```

```
// convert whichText to lowercase and split  
it into words
```

```
let words = whichText.toLowerCase().split('');
```

```
// object to store word frequencies
```

```
let wordFrequencies = {};
```

```
// loop through words in whichText
```

```
for (let i = 0; i < words.length; i++)
```

```
{
```

```
    let word = words[i];
```

```
    let isStopWord = false;
```

```
// is word not a stop word
for (let j = 0; j < stopWords.length; j++)
{
    if (word === stopWords[j])
    {
        isStopWord = true;
        break;
    }
}

if (!isStopWord && word.length > 1)
{
    // if word is not a stop word, count its
    frequency
    if (wordFrequencies[word])
    {
        wordFrequencies[word]++;
    }
}
```

```
}  
else  
{  
    wordFrequencies[word] = 1;  
}  
}  
}
```

// convert word frequencies object into an
array of [word, frequency] pairs

```
let frequencyArray = [];  
  
for (let word in wordFrequencies)  
{  
    frequencyArray.push([word,  
wordFrequencies[word]]);  
}
```

```
// sort array by frequency in descending  
order
```

```
frequencyArray.sort(function(a, b)  
{  
    return b[1] - a[1];  
});
```

```
// get top 10 keywords
```

```
let topKeywords = frequencyArray.slice(0,  
10).map(function(pair)  
{  
    return pair[0];  
});
```

```
// return top keywords as comma separated  
string
```

```
return topKeywords.join(', ');  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// extractKeywordsUppercase.js
```

```
function
extractKeywordsUppercase(whichText)
{
    // array of common stop words to ignore
    let stopWords = ["a", "an", "and", "the",
    "in", "on", "at", "of", "to", "is", "it", "that",
    "this", "with", "for", "as", "by", "but", "or",
    "nor", "up", "down", "out", "over", "under",
    "again", "further", "then", "once", "here",
```

```
"there", "when", "where", "why", "how", "all",  
"any", "both", "each", "few", "more", "most",  
"other", "some", "such", "no", "nor", "too",  
"very", "can", "will", "just", "should", "now"];
```

```
// convert whichText to lowercase and split  
it into words
```

```
let words = whichText.toLowerCase().split('');
```

```
// object to store word frequencies
```

```
let wordFrequencies = {};
```

```
// loop through words in whichText
```

```
for (let i = 0; i < words.length; i++)
```

```
{
```

```
    let word = words[i];
```



```
// is word not a stop word
```

```
let isStopWord = false;
```

```
for (let j = 0; j < stopWords.length; j++)
```

```
{
```

```
    if (word === stopWords[j])
```

```
    {
```

```
        isStopWord = true;
```

```
        break;
```

```
    }
```

```
}
```

```
if (!isStopWord && word.length > 1)
```

```
{
```

```
    // if word is not a stop word, count its  
frequency
```

```
    if (wordFrequencies[word])
```

```
    {
```

```
        wordFrequencies[word]++;  
    }  
    else  
    {  
        wordFrequencies[word] = 1;  
    }  
}  
}
```

// convert word frequencies object into an
array of [word, frequency] pairs

```
let frequencyArray = [];  
  
for (let word in wordFrequencies)  
{  
    frequencyArray.push([word,  
wordFrequencies[word]]);  
}
```

```
// sort array by frequency in descending
order
for (let i = 0; i < frequencyArray.length - 1;
i++)
{
    for (let j = i + 1; j <
frequencyArray.length; j++)
    {
        if (frequencyArray[i][1] <
frequencyArray[j][1])
        {
            let temp = frequencyArray[i];
            frequencyArray[i] =
frequencyArray[j];
            frequencyArray[j] = temp;
        }
    }
}
```

```
}
```

```
// get top 10 keywords
```

```
let topKeywords = [];
```

```
for (let i = 0; i < 10 && i <  
frequencyArray.length; i++)
```

```
{
```

```
    topKeywords.push(frequencyArray[i][0]);
```

```
}
```

```
// capitalize first letter of each keyword
```

```
for (let i = 0; i < topKeywords.length; i++)
```

```
{
```

```
    topKeywords[i] =
```

```
topKeywords[i].charAt(0).toUpperCase() +
```

```
topKeywords[i].slice(1);
```

```
}
```

```
// return top keywords as comma separated  
string  
return topKeywords.join(', ');  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// getWordCount.js
```

```
function getWordCount(whichText)
{
    let words = whichText.split(/\s+/);

    let wordCount = 0;

    for (let i = 0; i < words.length; i++)
    {
        if (words[i].length > 0)
```

```
{  
    wordCount++;  
}  
  
return wordCount;  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// getWordFrequency.js
```

```
function getWordFrequency(whichText)
{
    let words =
whichText.toLowerCase().split(/\s+/);

    let frequency = {};

    for (let i = 0; i < words.length; i++)
    {
```



```
let word = words[i];  
  
if (word.length > 0)  
{  
    if (frequency[word])  
    {  
        frequency[word]++;  
    }  
    else  
    {  
        frequency[word] = 1;  
    }  
}  
}
```

```
let frequencyList = "";
```

```
for (let word in frequency)
```

```
{  
    if (frequency.hasOwnProperty(word))  
    {  
        frequencyList += word + ': ' +  
frequency[word] + '<br>';  
    }  
}  
  
return frequencyList;  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// titleCaseKeywords.js
```

```
function titleCaseKeywords(whichText)
{
    // array of common stop words to ignore
    let stopWords = ["a", "an", "and", "the",
    "in", "on", "at", "of", "to", "is", "it", "that",
    "this", "with", "for", "as", "by", "but", "or",
    "nor"];
```

```
// convert whichText to lowercase and split  
it into words  
let words = whichText.toLowerCase().split('');
```

```
// object to store word frequencies  
let wordFrequencies = {};
```

```
// loop through words in whichText  
for (let i = 0; i < words.length; i++)  
{  
    let word = words[i];
```

```
    let isStopWord = false;
```

```
    // is word not a stop word  
    for (let j = 0; j < stopWords.length; j++)  
    {
```

```
if (word === stopWords[j])  
{  
    isStopWord = true;  
    break;  
}  
}
```

```
if (!isStopWord && word.length > 1)  
{  
    // if word is not a stop word, count its  
frequency  
    if (wordFrequencies[word])  
    {  
        wordFrequencies[word]++;  
    }  
    else  
    {  
        wordFrequencies[word] = 1;  
    }  
}
```

```
}  
}  
}
```

// convert word frequencies object into an array of [word, frequency] pairs

```
let frequencyArray = [];
```

```
let keys = Object.keys(wordFrequencies);
```

```
for (let i = 0; i < keys.length; i++)
```

```
{
```

```
    let word = keys[i];
```

```
    frequencyArray.push([word,  
wordFrequencies[word]]);
```

```
}
```

```
// sort array by frequency in descending
order
for (let i = 0; i < frequencyArray.length - 1;
i++)
{
    for (let j = i + 1; j <
frequencyArray.length; j++)
    {
        if (frequencyArray[i][1] <
frequencyArray[j][1])
        {
            let temp = frequencyArray[i];
            frequencyArray[i] =
frequencyArray[j];
            frequencyArray[j] = temp;
        }
    }
}
```

```
// get top 10 keywords
let topKeywords = [];

for (let i = 0; i < 10 && i <
frequencyArray.length; i++)
{
    topKeywords.push(frequencyArray[i][0]);
}

// loop through words in whichText and
capitalize only the keywords
let resultWords = [];

for (let i = 0; i < words.length; i++)
{
    let word = words[i];
    let isKeyword = false;
```



```
if (i === 0)  
{
```

```
    // always capitalize first word
```

```
    resultWords.push(word.charAt(0).toUpperCase()  
    + word.slice(1));  
    continue;  
}
```

```
for (let j = 0; j < topKeywords.length; j++)  
{  
    if (word === topKeywords[j])  
    {  
        isKeyword = true;  
        break;  
    }  
}
```

```
    if (isKeyword)
    {
resultWords.push(word.charAt(0).toUpperCase() + word.slice(1));
    }
    else
    {
        resultWords.push(word);
    }
}

return resultWords.join(' ');
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// boldText.js
```

```
function boldText(whichText)
{
    let bold = "<b>" + whichText + "</b>";

    return bold;
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// capitalizeFirstLetters.js
```

```
function capitalizeFirstLetters(whichText)
{
    // split whichText into words
    let words = whichText.split(" ");

    // array to hold capitalized words
    let capitalizedWords = [];

    // loop through each word
```

```
for (let i = 0; i < words.length; i++)  
{  
    let word = words[i];  
  
    // if word is not empty, capitalize first  
    letter and add to array  
    if (word.length > 0)  
    {  
        let capitalizedWord =  
word[0].toUpperCase() + word.slice(1);  
  
capitalizedWords.push(capitalizedWord);  
    }  
    else  
    {  
        capitalizedWords.push(word);  
    }  
}
```

```
}
```

```
// join capitalized words into a single string  
return capitalizedWords.join(" ");
```

```
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// capitalizeText.js
```

```
function capitalizeText(whichText)
{
    let upperCase = whichText.toUpperCase();

    return upperCase;
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// formatAsBulletedList.js
```

```
function formatAsBulletedList(whichText)
{
    let words = whichText.split(' ');

    // array to hold formatted list
    let formattedList = [];

    // loop through words in whichText
    for (let i = 0; i < words.length; i++)
```



```
{  
    // trim leading/trailing whitespace  
    let word = words[i].trim();  
  
    // add each non empty word as a new  
bullet point  
    if (word.length > 0)  
    {  
        formattedList.push("* " +  
word.charAt(0).toUpperCase() + word.slice(1)  
+ '<br>');  
    }  
}  
  
// return formatted list as a string with line  
breaks  
return formattedList.join("\n");  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// italicText.js
```

```
function italicText(whichText)
{
    let bold = "<i>" + whichText + "</i>";

    return bold;
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
// Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
// https://github.com/ChristopherAndrewTopalia
// n
// lowercaseText.js
```

```
function lowercaseText(whichText)
{
    let lowerCase = whichText.toLowerCase();

    return lowerCase;
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// normalText.js
```

```
function normalText(whichText)
{
    // replace <b> and </b> tags with empty
strings
    whichText = whichText.replace(/<\/?b>/gi,
    "");

    // replace <strong> and </strong> tags with
empty strings
```

```
whichText = whichText.replace(/<V?  
strong>/gi, "");  
  
return whichText;  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// underlineText.js
```

```
function underlineText(whichText)
{
    let underline = "<u>" + whichText + "</u>";

    return underline;
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// fetchAllEarthquakeData.js
```

```
function fetchAllEarthquakeData()
{
    return
    fetch('https://earthquake.usgs.gov/fdsnws/ev
ent/1/query?
format=geojson&orderby=time&minmagnitud
e=4.5')

    .then(function(response)
```

```
{  
  if (!response.ok)  
  {  
    throw new Error('Network response  
was not ok');  
  }  
  return response.json();  
})
```

```
.then(function(data)  
{  
  if (data.features.length > 0)  
  {  
    let earthquakeList = [];
```

**// iterate through all earthquakes and
collect information**


```
for (let i = 0; i < data.features.length; i+
+)
{
    let earthquake = data.features[i];

    // format date
    let date = new
Date(earthquake.properties.time);

    let formattedDate = `
${date.toDateString()} $
${date.toLocaleTimeString()}`;

    // make earthquake info string
    let earthquakeInfo = `Date: $
${formattedDate}, Magnitude: $
${earthquake.properties.mag}, Location: $
${earthquake.properties.place}`;
```

```
earthquakeList.push(earthquakeInfo);  
}
```

```
    // join earthquake info into a single  
string with newlines
```

```
    let allEarthquakesInfo =  
earthquakeList.join("\n");
```

```
    ge("textInput").value =  
allEarthquakesInfo;
```

```
    // return concatenated earthquake info  
    return allEarthquakesInfo;  
}  
else  
{
```

```
        throw new Error('No earthquake data  
found');  
    }  
})  
  
    .catch(function(error)  
    {  
        console.error('Error fetching earthquake  
data:', error);  
  
        // return error message or throw error  
        throw error;  
    });  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// fetchMostRecentEarthquakeData.js
```

```
function fetchMostRecentEarthquakeData()
{

fetch('https://earthquake.usgs.gov/fdsnws/ev
ent/1/query?
format=geojson&orderby=time&minmagnitud
e=4.5')

.then(function(response)
```

```
{  
  if (!response.ok)  
  {  
    throw new Error('Network response  
was not ok');  
  }  
  return response.json();  
})  
  
.then(function(data)  
{  
  if (data.features.length > 0)  
  {  
    // get most recent earthquake  
    let earthquake = data.features[0];  
  
    // format date
```

```
    let date = new  
    Date(earthquake.properties.time);
```

```
    let formattedDate = `$  
    {date.toDateString()} $  
    {date.toLocaleTimeString()}`;
```

```
    // make earthquake info string  
    let earthquakeInfo = `Date: $  
    {formattedDate}, Magnitude: $  
    {earthquake.properties.mag}, Location: $  
    {earthquake.properties.place}`;
```

```
    // set value of textInput element with  
    earthquake info  
    ge("textInput").value = earthquakeInfo;  
  
    return earthquakeInfo;
```

```
    }  
    else  
    {  
        throw new Error('No earthquake data  
found');  
    }  
})  
  
    .catch(function(error)  
    {  
        console.error('Error fetching earthquake  
data:', error);  
    });  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// fetchRandomJoke.js
```

```
function fetchRandomJoke()
{
    fetch('https://official-joke-
api.appspot.com/random_joke')

    .then(function(response)
    {
        return response.json();
    })
}
```



```
.then(function(data)
{
    let joke = data.setup + " " +
data.punchline;

    ge("textInput").value = joke;
})

.catch(function(error)
{
    console.error('Error fetching joke:',
error);
});
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// audioPlay.js
```

```
function audioPlay(whichId, whichVolume)
{
    let audio = ge(whichId);

    if (audio)
    {
        audio.volume = whichVolume;
        audio.play();
    }
}
```



```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
// Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
// https://github.com/ChristopherAndrewTopalia
// n
// loadSounds.js
```

```
function loadSounds(whichArray)
{
    for (let x = 0; x < whichArray.length; x++)
    {
        let theSound = ce('audio');

        if (online == false)
        {
```

```
        theSound.src =  
whichArray[x].soundFileOffline;  
    }  
    else  
    {  
        theSound.src =  
whichArray[x].soundFileOnline;  
    }
```

```
        theSound.id = whichArray[x].name;  
        theSound.loop = false;  
        theSound.volume = 1.0;  
        ba(theSound);  
    }  
}
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// sounds.js
```

```
let sounds =
[
  {
    name: 'sfx_blip_001',
    soundFileOffline:
'src/media/sounds/sfx_blip_001.mp4',
    soundFileOnline:
'https://collegeofscripting.weebly.com/upload
s/6/4/4/8/64482293/sfx_blip_001.mp4'
```

```
},  
  
{  
  name: 'sfx_warp_001',  
  soundFileOffline:  
'src/media/sounds/sfx_warp_001.mp3',  
  soundFileOnline:  
'https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/sfx_warp_001.mp3'  
}  
];
```

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalia
n
// whenLoaded.js
```

```
function whenLoaded()
{
    loadSounds(sounds);

    makeInterface();

    makeTitleOfApp();
}
```


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)
```

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.

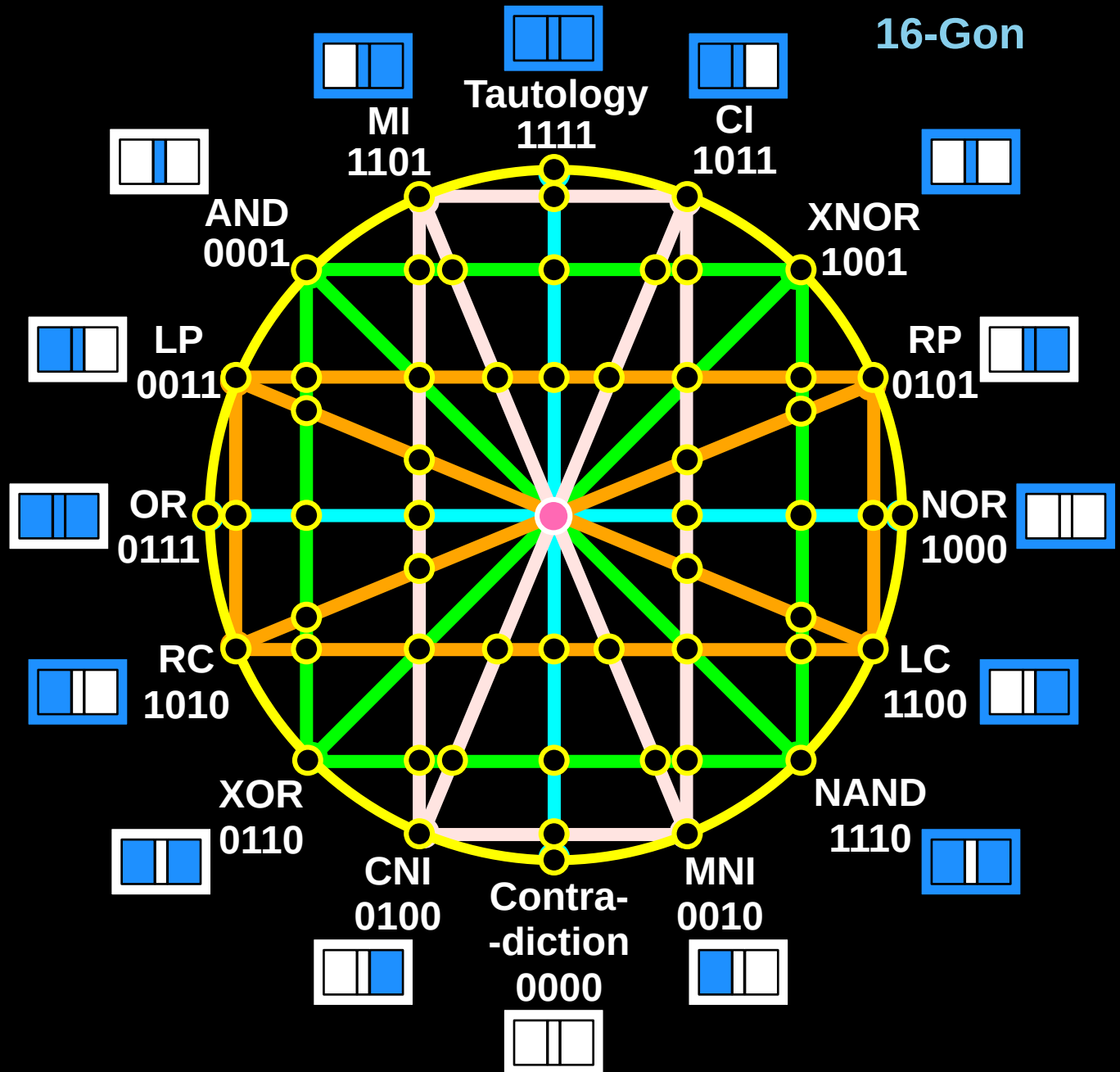
We can do the same thing for .html files, where we change the file type it will be combining to .html and it will combine all .html files into one main.html file.

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

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