Topalian JavaScript List Generator

by
Christopher Andrew Topalian

All Rights Reserved
Copyright 2000-2024

Dedicated
to
God the Father

Topalian JavaScript List Generator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_List_Generator/Topalian_Java

100%

Topalian JavaScript List Generator

Add Text Box

Array

Comma Separated List in a Row

List in a Column

Comma Separated List in a Column

2:06 PM
11/3/2024
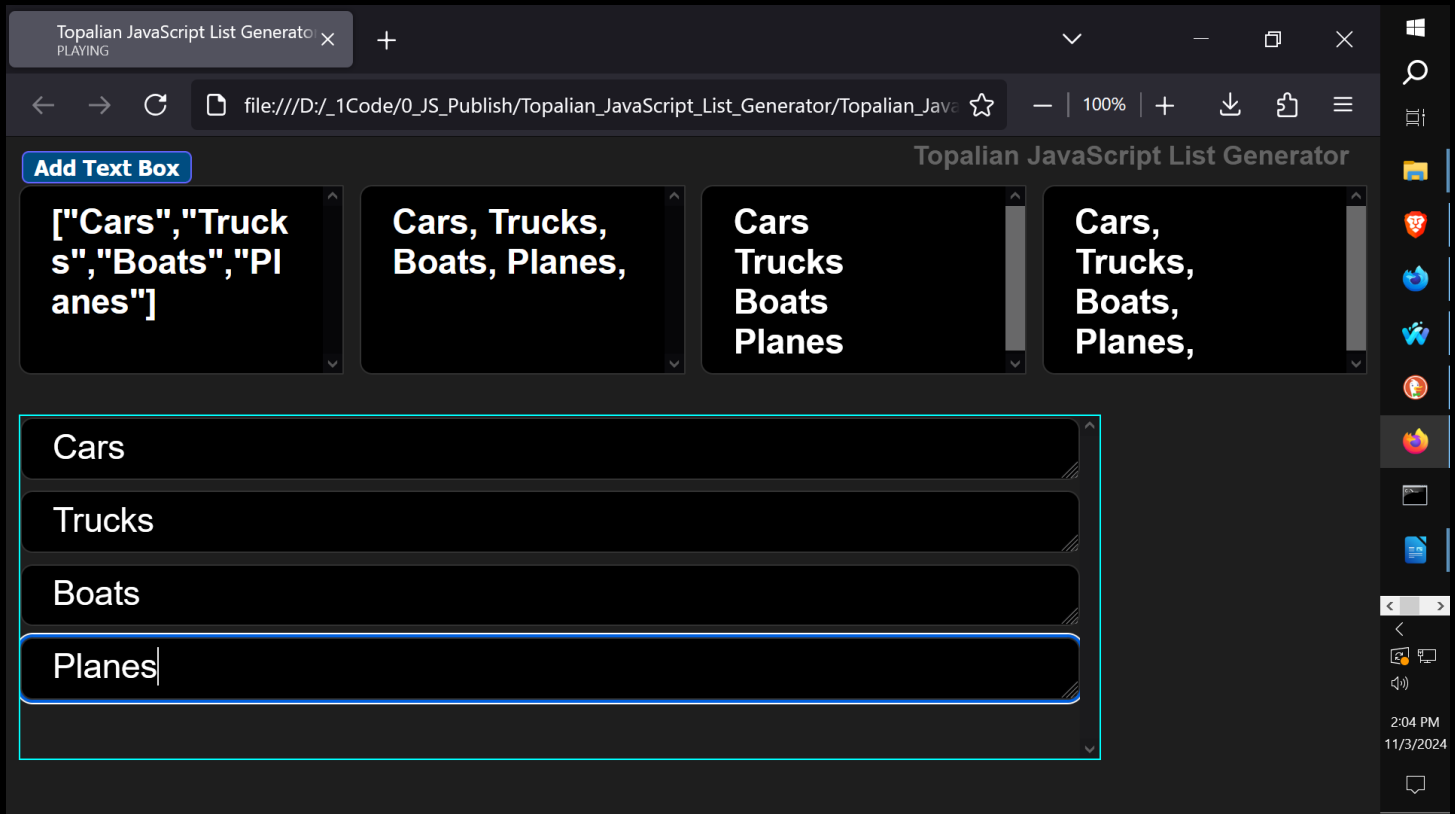
Topalian JavaScript List Generator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_List_Generator/Topalian_Java

100%

Add Text Box

Topalian JavaScript List Generator

["Cars","Trucks","Boats","Planes"]

Cars, Trucks, Boats, Planes,

Cars
Trucks
Boats
Planes

Cars,
Trucks,
Boats,
Planes,

Cars

Trucks

Boats

Planes

2:04 PM
11/3/2024

```html
<!-- Dedicated to God the Father -->
<!-- All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024 -->
<!-- https://github.com/ChristopherTopalian -->
<!-- https://github.com/ChristopherAndrewTopalian -->
<!-- https://github.com/ChristopherTopalian -->
<!--
Topalian_JavaScript_List_Generator_(Online_Version).html -->
<!-- Version 001 - (2024-11-03) -->

<html>
<head>
<title> Topalian JavaScript List Generator </title>

<style>

/* style001.css */

body
{
  background-color: rgb(30, 30, 30);
  font-weight: bold;
  color: rgb(255, 255, 255);
}
```

```css
button
{
  background-color: rgb(0, 77, 129);
  margin-left: 2px;
  margin-right: 2px;
  margin-top: 1px;
  margin-bottom: 1px;
  border-style: solid;
  border-width: 1px;
  border-color: rgb(100, 100, 255);
  border-radius: 4px;
  padding-left: 7px;
  padding-right: 7px;
  padding-top: 1px;
  padding-bottom: 1px;
  opacity: 1.0;
  font-size: 14px;
  font-weight: bold;
  color: rgb(255, 255, 255);
  text-align: center;
}

button:hover
{
  background-color: rgb(7, 31, 47);
  border-color: rgb(0, 255, 255);
```

```css
    color: rgb(0, 255, 255);
    cursor: pointer;
}


button:active
{
    color: rgb(255, 0, 255);
}


div:hover
{
    border-color: rgb(0, 255, 255);
}


.textEntry
{
    background-color: rgb(0, 0, 0);
    border: solid 1px rgb(50, 50, 50);
    border-radius: 8px;
    padding-left: 20px;
    padding-right: 20px;
    padding-top: 5px;
    padding-bottom: 5px;
    font-family: arial;
    font-size: 22px;
    color: rgb(255, 255, 255);
```

```css
}

.reportContainer
{
  display: flex;
  flex-direction: row;
  gap: 10px;
}

.stringifiedArrayDiv
{
  width: 300px;
  height: 100px;
  border: solid 1px rgb(50, 50, 50);
  border-radius: 8px;
  background-color: rgb(0, 0, 0);
  padding-left: 20px;
  padding-right: 20px;
  padding-top: 10px;
  padding-bottom: 10px;
  color: rgb(255, 255, 255);
  overflow-y: scroll;
  word-wrap: break-word;
}

.listColumnDiv
```

```css
{
  width: 300px;
  height: 100px;
  border: solid 1px rgb(50, 50, 50);
  border-radius: 8px;
  background-color: rgb(0, 0, 0);
  padding-left: 20px;
  padding-right: 20px;
  padding-top: 10px;
  padding-bottom: 10px;
  color: rgb(255, 255, 255);
  overflow-y: scroll;
}

.csvListDiv
{
  width: 300px;
  height: 100px;
  border: solid 1px rgb(50, 50, 50);
  border-radius: 8px;
  background-color: rgb(0, 0, 0);
  padding-left: 20px;
  padding-right: 20px;
  padding-top: 10px;
  padding-bottom: 10px;
  color: rgb(255, 255, 255);
```

```css
    overflow-y: scroll;
    word-wrap: break-word;
}

.listRowDiv
{
    width: 300px;
    height: 100px;
    border: solid 1px rgb(50, 50, 50);
    border-radius: 8px;
    background-color: rgb(0, 0, 0);
    padding-left: 20px;
    padding-right: 20px;
    padding-top: 10px;
    padding-bottom: 10px;
    color: rgb(255, 255, 255);
    overflow-y: scroll;
    word-wrap: break-word;
}

.entriesContainer
{
    border: solid 1px rgb(50, 50, 50);
    display: flex;
    flex-direction: column;
    overflow-y: scroll;
```

```css
    gap: 5px;
}


/*----*/


/* Dedicated to God the Father */
/* All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024 */
/* https://github.com/ChristopherTopalian */
/* https://github.com/ChristopherAndrewTopalian */
/* https://sites.google.com/view/CollegeOfScripting */


</style>


<script>


// 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;
}


function ct(whichText)
{
  let result = document.createTextNode(whichText);
  return result;
}


function cl(whichMessage)
{
  let result = console.log(whichMessage);
  return result;
}


function gr(whichId)
{
  let result =  ge(whichId).getBoundingClientRect();
  return result;
```

```
}
```

//----//

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting

// worldVariables.js

let theTextEntryId = 0;

let textArray = [];
```

//----//

```
// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting
```

```
// makeEntriesContainer.js

function makeEntriesContainer()
{
  let entriesContainer = ce('div');
  entriesContainer.id = 'entriesContainer';
  entriesContainer.className = 'entriesContainer';
  entriesContainer.style.width = 80 + '%';
  entriesContainer.style.height = 220 + 'px';
  ba(entriesContainer);
}


//----//


// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting//
makeInterface.js

function makeInterface()
{
  let mainDiv = ce("div");
  mainDiv.id = 'mainDivApp';
```

```javascript
ba(mainDiv);

//-//

let buttonAdd = ce("button");
buttonAdd.textContent = "Add Text Box";
buttonAdd.onmouseover = function()
{
  hoverSound();
};
buttonAdd.onclick = function()
{
  clickSound();
  makeTextEntryInput();
};
mainDiv.append(buttonAdd);

//-//

let reportContainer = ce("div");
reportContainer.id = "reportContainer";
reportContainer.className = 'reportContainer';
mainDiv.append(reportContainer);

//-//
```

```javascript
let stringifiedArrayDiv = ce("div");
stringifiedArrayDiv.id = 'stringifiedArrayDiv';
stringifiedArrayDiv.title = 'stringifiedArrayDiv';
stringifiedArrayDiv.innerHTML = 'Array';
stringifiedArrayDiv.className = "stringifiedArrayDiv";
stringifiedArrayDiv.onmouseover = function()
{
  hoverSound();
};
reportContainer.append(stringifiedArrayDiv);


//-//


let listRowDiv = ce("div");
listRowDiv.id = 'listRowDiv';
listRowDiv.title = 'listRowDiv';
listRowDiv.innerHTML = 'Comma Separated List in a
Row';
listRowDiv.className = 'listRowDiv';
listRowDiv.onmouseover = function()
{
  hoverSound();
};
reportContainer.append(listRowDiv);


//-//
```

```javascript
let listColumnDiv = ce("div");
listColumnDiv.id = 'listColumnDiv';
listColumnDiv.title = 'listColumnDiv';
listColumnDiv.innerHTML = 'List in a Column';
listColumnDiv.className = 'listColumnDiv';
listColumnDiv.onmouseover = function()
{
  hoverSound();
};
reportContainer.append(listColumnDiv);

//-//

let csvListDiv = ce("div");
csvListDiv.id = 'csvListDiv';
csvListDiv.title = 'csvListDiv';
csvListDiv.innerHTML = 'Comma Separated List in a
Column';
csvListDiv.className = 'csvListDiv';
csvListDiv.onmouseover = function()
{
  hoverSound();
};
reportContainer.append(csvListDiv);
```

```
  //-//


  mainDiv.append(ce('br'));
}


//----//


// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting


// makeTextEntryInput.js


function makeTextEntryInput()
{
  theTextEntryId += 1;


  let textEntry = ce("textarea");
  textEntry.id = theTextEntryId;
  textEntry.title = textEntry.id;
  textEntry.className = "textEntry";
  textEntry.style.height = 40 + 'px';
  textEntry.onclick = function()
```

```javascript
  {
    clickSound();
  };
  textEntry.onkeyup = function()
  {
    getAllTextareas();
  };
  ge('entriesContainer').append(textEntry);
}

//----//

// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting

// makeTitleOfApp.js

function makeTitleOfApp()
{
  let mainDiv = ce('div');
  mainDiv.style.position = 'absolute';
  mainDiv.style.right = '20px';
```

```javascript
  mainDiv.style.top = '2px';
  mainDiv.style.zIndex = 1;
  ba(mainDiv);


  //-//


  let titleOfApp = ce('a');
  titleOfApp.id = 'titleOfApp';
  titleOfApp.textContent = 'Topalian JavaScript List
Generator';
  titleOfApp.href =
'https://github.com/christophertopalian/';
  titleOfApp.target = '_blank';
  titleOfApp.style.fontSize = '17px';
  titleOfApp.style.fontWeight = 'bold';
  titleOfApp.style.textDecoration = 'none';
  titleOfApp.style.color = 'rgb(100, 100, 100)';
  mainDiv.append(titleOfApp);
}


//----//


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

```
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting

// getAllTextareas.js

function getAllTextareas()
{
  textArray = [];

  let theTextareas =
document.getElementsByTagName("textarea");

  for (let x = 0; x < theTextareas.length; x++)
  {
    textArray.push(theTextareas[x].value);
  }

  cl(textArray);

  // show result as JSON stringified array
  ge("stringifiedArrayDiv").innerHTML =
JSON.stringify(textArray);

  let theString = "";

  for (let z = 0; z < theTextareas.length; z++)
```

Christopher Topalian

```
{
  theString += theTextareas[z].value;

  theString += "<br>";
}

// send theString to listColumnDiv
ge("listColumnDiv").innerHTML = theString;


//----//


let theString2 = "";


for (let z = 0; z < textArray.length; z++)
{
  theString2 += textArray[z] + ', ';
  theString2 += "<br>";
}


// send theString2 to csvListDiv
ge('csvListDiv').innerHTML = theString2;


//-//


let theString3 = "";
```

```
  for (let j = 0; j < textArray.length; j++)
  {
    theString3 += textArray[j] + ', ';
  }


  // send theString3 to listRowDiv
  ge('listRowDiv').innerHTML = theString3;


  //-//


  return textArray;
}


//----//


// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting


// sound.js

let audioCtx = new (window.AudioContext ||
window.webkitAudioContext)();
```

```
function playTone(frequency, duration, waveType,
volume, detune, attackTime, releaseTime, panValue)
{
  let oscillator = audioCtx.createOscillator();
  // 'sine', 'square', 'sawtooth', 'triangle'
  oscillator.type = waveType;
  oscillator.frequency.setValueAtTime(frequency,
audioCtx.currentTime);
  oscillator.detune.setValueAtTime(detune,
audioCtx.currentTime);

  let gainNode = audioCtx.createGain();
  let pannerNode = audioCtx.createStereoPanner();

  oscillator.connect(gainNode);
  gainNode.connect(pannerNode);
  pannerNode.connect(audioCtx.destination);

  gainNode.gain.setValueAtTime(0,
audioCtx.currentTime);

  // attack
  gainNode.gain.linearRampToValueAtTime(volume,
audioCtx.currentTime + attackTime);
  gainNode.gain.linearRampToValueAtTime(0,
```

```
  // release
  audioCtx.currentTime + duration / 1000 - releaseTime);

  pannerNode.pan.setValueAtTime(panValue,
audioCtx.currentTime);

  oscillator.start();
  oscillator.stop(audioCtx.currentTime + duration /
1000);
}

function hoverSound()
{
  playTone(
    300,     // frequency
    500,     // duration
    'sine',  // waveType
    0.2,     // volume
    50,      // detune
    0.2,     // attackTime
    0.2,     // releaseTime
    -0.5     // panValue -0.5 is left
  );
}
```

```
function clickSound()
{
  playTone(
    400,      // frequency
    500,      // duration
    'sine',   // waveType
    0.2,      // volume
    50,       // detune
    0.2,      // attackTime
    0.2,      // releaseTime
    -0.5      // panValue -0.5 is left
  );
}

//--//


// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting


// whenLoaded.js


function whenLoaded()
```
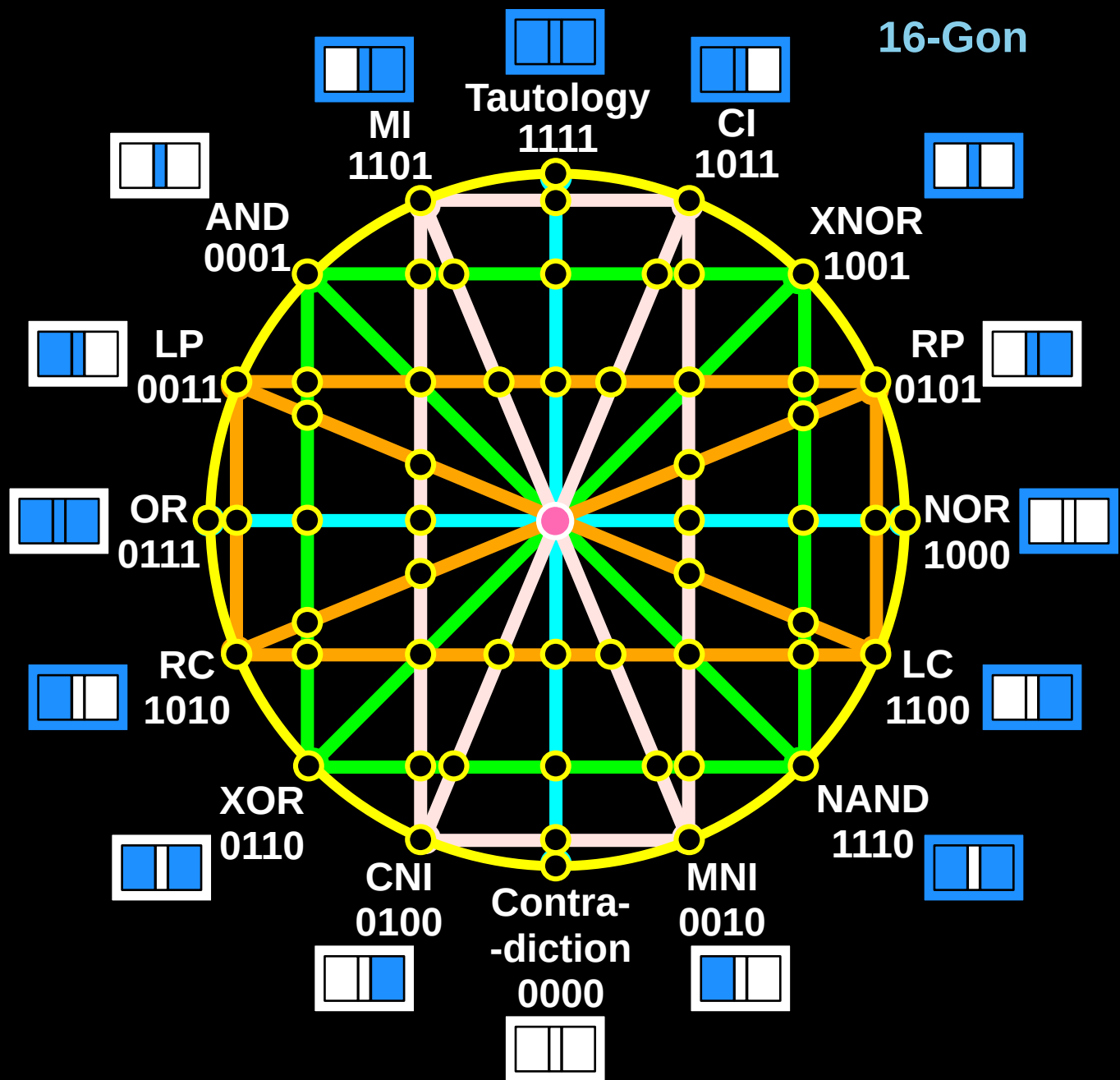
```
{
    makeInterface();

    makeEntriesContainer();

    makeTitleOfApp();
}

//----//

// Dedicated to God the Father
// All Rights Reserved Christopher Andrew Topalian
Copyright 2000-2024
// https://github.com/ChristopherTopalian
// https://github.com/ChristopherAndrewTopalian
// https://sites.google.com/view/CollegeOfScripting

</script>

</head>

<body onload = "whenLoaded();">

</body>

</html>
```

True Artificial Intelligence System

16-Gon

Tautology
1111

MI
1101

CI
1011

AND
0001

XNOR
1001

LP
0011

RP
0101

OR
0111

NOR
1000

RC
1010

LC
1100

XOR
0110

NAND
1110

CNI
0100

Contra-
-diction
0000

MNI
0010

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