

Topalian Game Engine Data Navigator 025

**by
Christopher Andrew Topalian**

All Rights Reserved
Copyright 2000-2024

Dedicated to God the Father

Topalian Game Engine Data Navigator X +

file:///D:/_1Code/0_JS_Published/Topalian_Game_Engine_Data_Navigator/Topalian_Ga ★ 100%

Dedicated to God the Father

Topalian Game Engine Data Navigator



```
structuresArray = [
{
  "name": "House 1",
  "id": "1",
  "posX": 1100,
  "posY": 400,
  "sizeX": 110,
  "sizeY": 45,
  "colorFill": "black",
  "colorText": "white",

```

```
playerArray = [
{
  "name": "Christopher",
  "id": "thePlayer",
  "x": 140,
  "y": 60,
  "width": 90,
  "height": 250,
  "speedMultiplier": 7,
  "colorFill": "rgb(73, 100, 150)",
  "colorText": "rgb(255, 255, 255)",

```

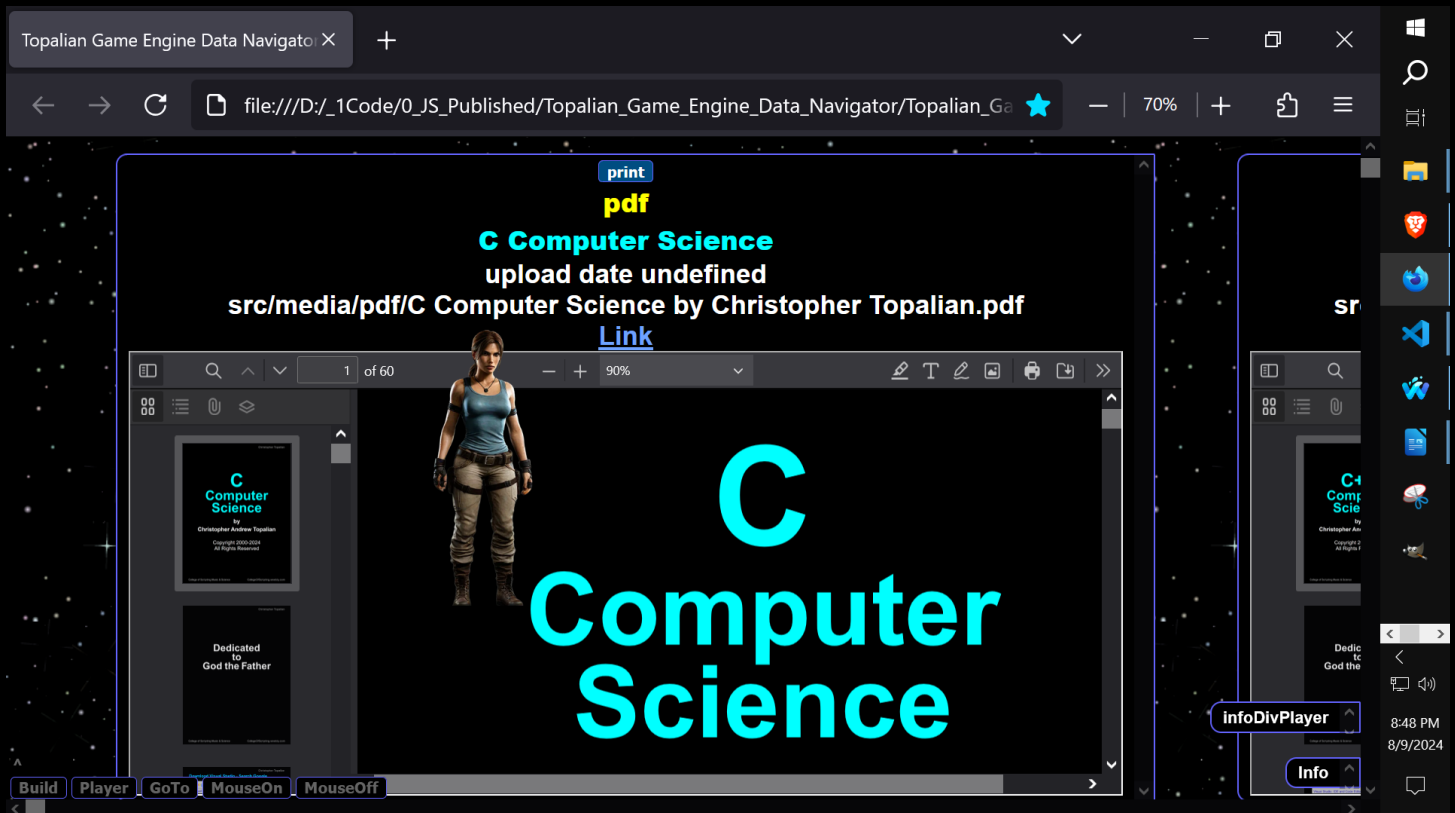
4 House

infoDivPlayer

Info

Build Player GoTo MouseOn MouseOff

8:47 PM 8/9/2024



Topalian Game Engine Data Navigator X +

file:///D:/_1Code/0_JS_Published/Topalian_Game_Engine_Data_Navigator/Topalian_Ga ★ 90%

JavaScript


array

at()

single quote ' and double quote "
Returns the indexed element
from the array

array.at(index)

let ourArray = ourArray.at(index);
returns the element of the given
position (index) in the array.



JavaScript

array

concat()

Joins two or more arrays without
changing the existing arrays, but
instead returns a new array.

array1.concat(array2, array3);

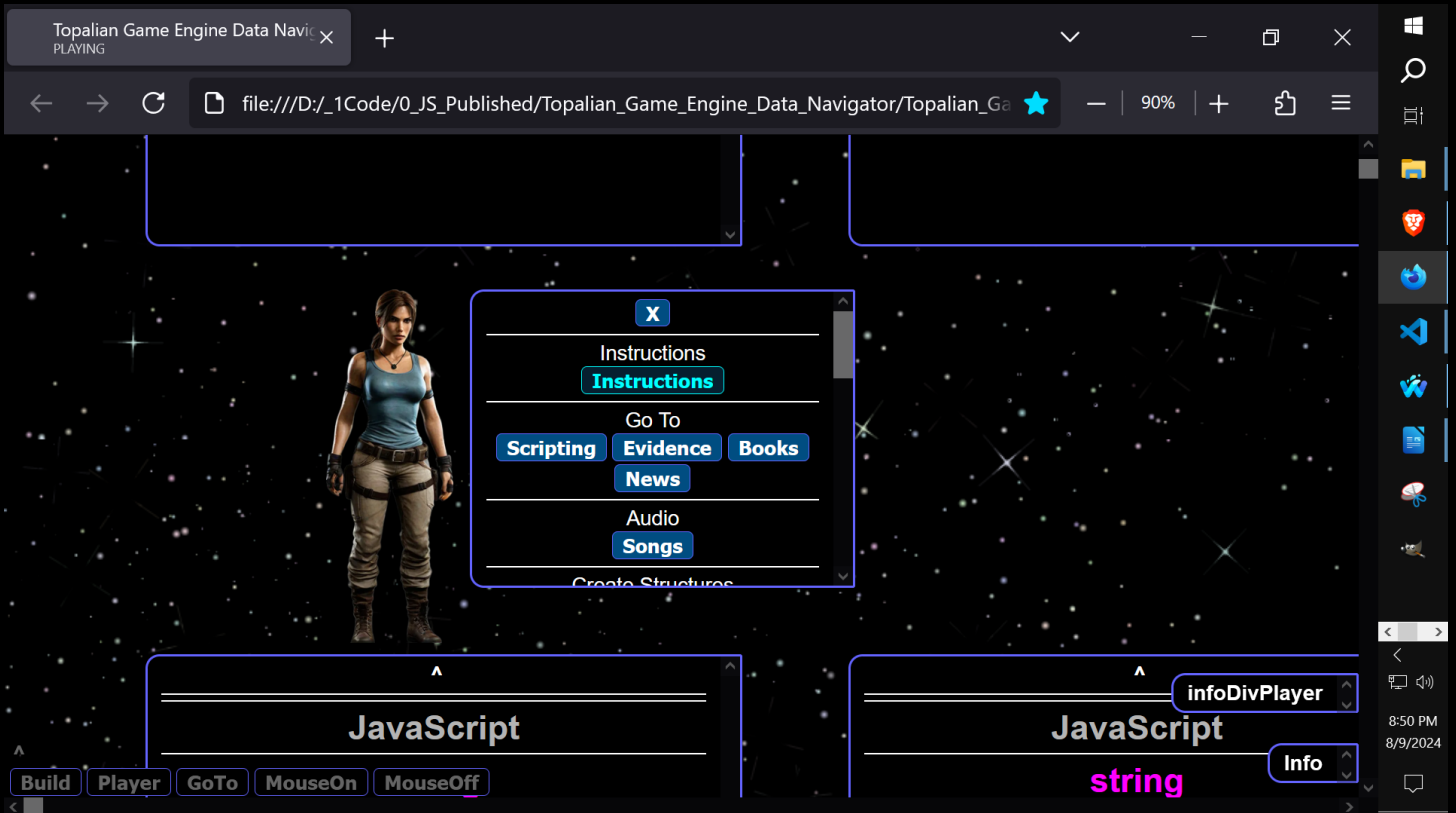
var combinedArray =
theNotes.concat(theNews,
evidence);

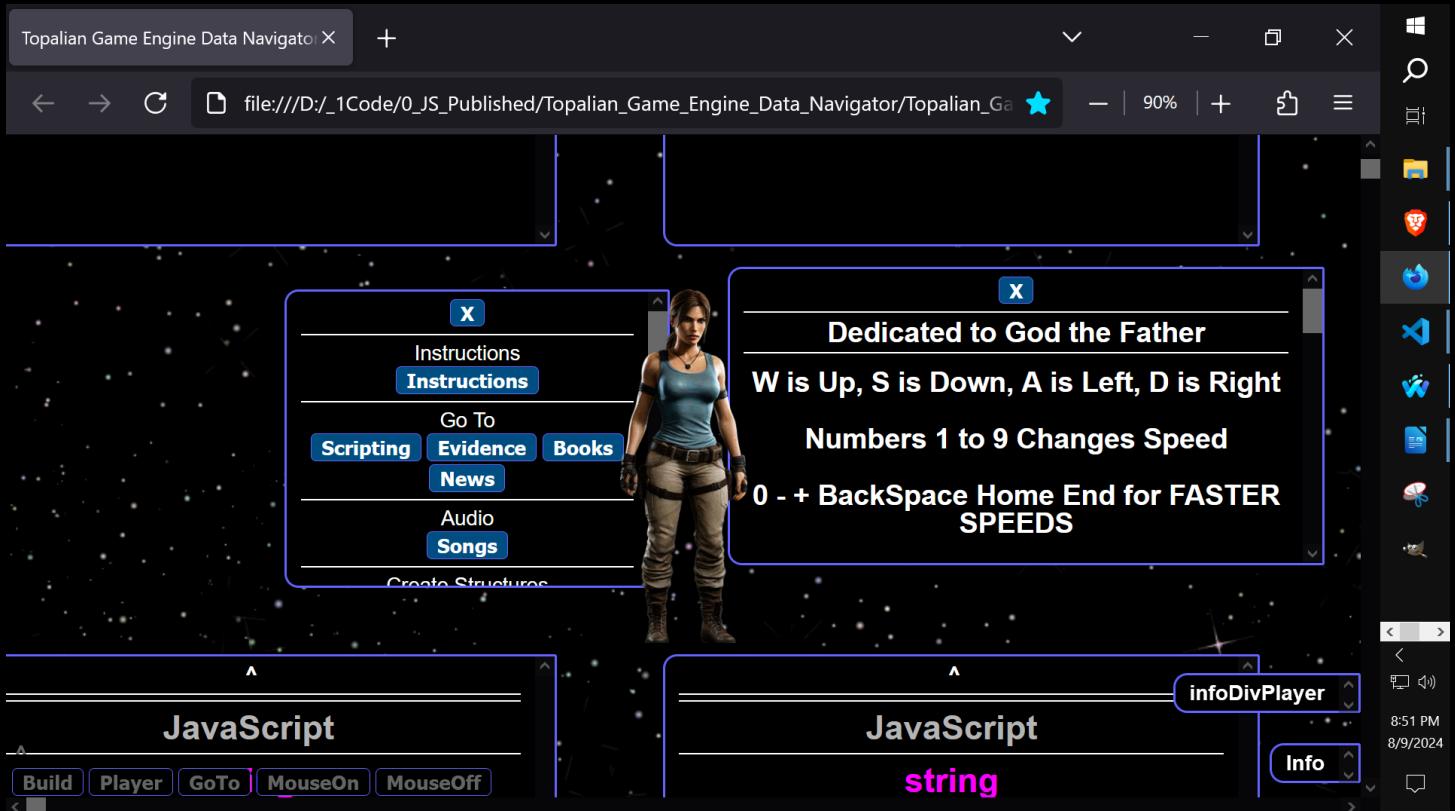
returns a new array

Build Player GoTo MouseOn MouseOff

infoDivPlayer Info

8:49 PM
8/9/2024





```
<!-- 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/Topalia
nGameEngineDataNavigator -->
<!--
Topalian_Game_Engine_Data_Navigator.html --
>
<!-- Version 025 - (2024-08-10) -->
```

```
<html>
<head>
<title> Topalian Game Engine Data Navigator
</title>

<style>
```

```
html, body
{
    background-color: rgb(0, 0, 0);
}
```



```
</style>
```

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

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

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

```
<!-- playerArray -->
```

```
<script src =  
"src/js/2data/playerData.js"></script>
```

```
<!-- structuresArray -->
```

```
<script src =  
"src/js/2data/structuresArray.js"></script>
```

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

```
<script src =  
"src/js/3worldVariables/worldVariables.js"></sc  
ript>
```

```
<!-- data -->
```

```
<script src = "src/data/booksPdf.js"></script>
```

```
<script src =  
"src/data/booksWebsite.js"></script>
```

```
<script src =  
"src/data/boxingVideos.js"></script>
```

```
<script src =  
"src/data/collegeBooks.js"></script>
```

```
<script src =  
"src/data/collegeLinks.js"></script>
```

```
<script src =  
"src/data/collegeVideos.js"></script>
```

```
<script src = "src/data/evidence.js"></script>
```

```
<script src =  
"src/data/javascriptLanguage.js"></script>
```

```
<script src =  
"src/data/lindenLanguage.js"></script>
```

```
<script src =  
"src/data/mathematics.js"></script>
```

```
<script src =  
"src/data/petitoGabrielleInvestigation.js"></scr  
ipt>
```

```
<script src =  
"src/data/ourVideoFiles.js"></script>
```

```
<script src =  
"src/data/scienceImages.js"></script>
```

```
<script src =  
"src/data/rickyHattonInterview.js"></script>
```

```
<script src =  
"src/data/socialMediaEvidence.js"></script>
```

```
<script src = "src/data/techVideos.js"></script>
```

```
<script src = "src/data/theNews.js"></script>
```

```
<script src = "src/data/theNotes.js"></script>
```

```
<script src =  
"src/data/videoEvidence.js"></script>
```

```
<script src =  
"src/data/videoInvestigator.js"></script>
```

```
<script src =  
"src/data/computerSciencePdf.js"></script>
```

```
<!-- get -->  
<script src = "src/js/get/get.js"></script>
```

```
<!-- player -->  
<script src =  
"src/js/player/playerUpdate.js"></script>
```

```
<!-- engine -->  
<script src =  
"src/js/engine/cameraFollowsPlayer.js"></script>
```

```
<script src =  
"src/js/engine/keyboardControl.js"></script>
```

```
<script src =  
"src/js/engine/keepPlayerInWorld.js"></script>
```

```
<script src =  
"src/js/engine/keyboardMotion.js"></script>
```

```
<script src =  
"src/js/engine/movePlayer.js"></script>
```

```
<script src =  
"src/js/engine/gameLoop.js"></script>
```

```
<!-- content containers -->
```

```
<script src =  
"src/js/contentContainers/1contentWorldVariab  
les.js"></script>
```

```
<script src =  
"src/js/contentContainers/containerBooks.js">  
</script>
```

```
<script src =  
"src/js/contentContainers/containerBoxingVide  
os.js"></script>
```

```
<script src =  
"src/js/contentContainers/containerLanguage.j  
s"></script>
```

```
<script src =  
"src/js/contentContainers/containerNews.js"><  
/script>
```

```
<script src =  
"src/js/contentContainers/containerNewsVideo  
s.js"></script>
```

```
<script src =  
"src/js/contentContainers/containerScience.js"  
></script>
```

```
<script src =  
"src/js/contentContainers/wasLineCrossed.js"  
></script>
```

```
<!-- utility -->  
<script src = "src/js/utility/copylt.js"></script>
```

```
<script src =  
"src/js/utility/removeElement.js"></script>
```

```
<script src = "src/js/utility/listens.js"></script>
```

```
<script src =  
"src/js/utility/loadScript.js"></script>
```

```
<script src =  
"src/js/utility/printing.js"></script>
```

```
<script src =  
"src/js/utility/preventContextMenu.js"></script  
>
```

```
<!-- menu -->  
<script src =  
"src/js/menu/menuPlayer.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerColor.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerColorText.js"></script  
>
```

```
<script src =  
"src/js/menu/menuPlayerDistance.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerOpacity.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerPositionSet.js"></scri  
pt>
```

```
<script src =  
"src/js/menu/menuPlayerPositionGet.js"></scri  
pt>
```

```
<script src =  
"src/js/menu/menuPlayerName.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerSizeGet.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerSizeSet.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerSpeed.js"></script>
```

```
<script src =  
"src/js/menu/menuPlayerTexture.js"></script>
```

```
<script src =  
"src/js/menu/menuSearch.js"></script>
```



```
<script src =  
"src/js/menu/menuStructureColor.js"></script>
```

```
<script src =  
"src/js/menu/menuStructureColorText.js"></scr  
ipt>
```

```
<script src =  
"src/js/menu/menuStructureDistance.js"></scri  
pt>
```

```
<script src =  
"src/js/menu/menuStructureEdit.js"></script>
```

```
<script src =  
"src/js/menu/menuStructureName.js"></script>
```

```
<script src =  
"src/js/menu/menuStructureOpacity.js"></scrip  
t>
```

```
<script src =  
"src/js/menu/menuStructurePositionGet.js"></  
script>
```

```
<script src =  
"src/js/menu/menuStructurePositionSet.js"></s  
cript>
```

```
<script src =  
"src/js/menu/menuStructureTexture.js"></scrip  
t>
```

```
<script src =  
"src/js/menu/menuStructureSizeGet.js"></scrip  
t>
```

```
<script src =  
"src/js/menu/menuStructureSizeSet.js"></scrip  
t>
```

```
<!-- make -->  
<script src =  
"src/js/make/makePlayer.js"></script>
```

```
<script src =  
"src/js/make/makeBackground.js"></script>
```

```
<script src =  
"src/js/make/makeStructureTextArea.js"></scri  
pt>
```

```
<script src =  
"src/js/make/makeEngineTitle.js"></script>
```

```
<script src =  
"src/js/make/makeInfoDiv.js"></script>
```

```
<script src =  
"src/js/make/makeInfoDivPlayer.js"></script>
```

```
<script src =  
"src/js/make/makeDedication.js"></script>
```

```
<script src =  
"src/js/make/makePlayerTextArea.js"></script>
```

```
<script src =  
"src/js/make/makeLine.js"></script>
```

```
<script src =  
"src/js/make/makeInstructions.js"></script>
```

```
<script src =  
"src/js/make/makeLogo.js"></script>
```

```
<script src =  
"src/js/make/makeBottomToolbar.js"></script>
```

```
<script src =  
"src/js/make/makeVideo.js"></script>
```

```
<script src =  
"src/js/mouse/mousePos.js"></script>
```

```
<script src =  
"src/js/mouse/mouseOn.js"></script>
```

```
<script src =  
"src/js/mouse/mouseOff.js"></script>
```

```
<!-- pos size -->
```

```
<script src =  
"src/js/posSize/sizeGet.js"></script>
```

```
<script src =  
"src/js/posSize/positionDistance.js"></script>
```

```
<script src =  
"src/js/posSize/positionGet.js"></script>
```

```
<script src =  
"src/js/posSize/positionGetSizeAndSides.js"><  
/script>
```

```
<script src =  
"src/js/posSize/travelTo.js"></script>
```

```
<!-- rows -->  
<script src =  
"src/js/rows/rowOfLanguage.js"></script>
```

```
<script src =  
"src/js/rows/rowOfLanguageShowAll.js"></scri  
pt>
```

```
<script src =  
"src/js/rows/rowOfNews.js"></script>
```

```
<script src =  
"src/js/rows/rowOfNotes.js"></script>
```

```
<script src =  
"src/js/rows/rowOfNotesWithTextArea.js"></scr  
ipt>
```

```
<script src =  
"src/js/rows/rowOfVideos.js"></script>
```

```
<script src =  
"src/js/rows/rowOfVideosMultipleFilters.js"></s  
cript>
```

```
<script src =  
"src/js/rows/rowOfWebsitesOrVideos.js"></scri  
pt>
```

```
<script src =  
"src/js/rows/rowOfEvidence.js"></script>
```

```
<script src =  
"src/js/rows/rowOfEvidenceWithImage.js"></sc  
ript>
```

```
<script src =  
"src/js/rows/rowOfInvestigation.js"></script>
```

```
<script src =  
"src/js/rows/rowOfScienceImagesAndLink.js">  
</script>
```

```
<script src =  
"src/js/rows/rowOfSocialMediaEvidence.js"></  
script>
```

```
<script src =  
"src/js/rows/rowOfDetails.js"></script>
```

```
<script src =  
"src/js/rows/rowOfCollegeBooks.js"></script>
```

```
<script src =  
"src/js/rows/rowOfCollegeLinks.js"></script>
```

```
<script src =  
"src/js/rows/rowOfBookImages.js"></script>
```

```
<script src =  
"src/js/rows/rowOfVideosOnlineStartEnd.js"></  
script>
```

```
<!-- data functions -->
```

```
<script src =  
"src/js/dataFunctions/playerLoadData.js"></scr  
ipt>
```

```
<script src =  
"src/js/dataFunctions/downloadData.js"></scri  
pt>
```

```
<script src =  
"src/js/dataFunctions/recordDelete.js"></script  
>
```

```
<script src =  
"src/js/dataFunctions/showPlayerData.js"></sc  
ript>
```

```
<script src =  
"src/js/dataFunctions/showStructureData.js"><  
/script>
```

```
<script src =  
"src/js/dataFunctions/sort/sortData.js"></script  
>
```

```
<script src =  
"src/js/dataFunctions/sort/sortByDate.js"></scr  
ipt>
```



```
<script src =  
"src/js/dataFunctions/sort/sortByNumber.js"><  
/script>
```

```
<!-- sound -->  
<script src =  
"src/js/sound/1soundsArray.js"></script>
```

```
<script src =  
"src/js/sound/2songsArray.js"></script>
```

```
<script src = "src/js/sound/sound.js"></script>
```

```
<script src =  
"src/js/sound/songChoicesMenu.js"></script>
```

```
<script src =  
"src/js/sound/volumeSet.js"></script>
```

```
<script src =  
"src/js/sound/stopSound.js"></script>
```

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

```
<script src =  
"src/js/sound/playSong.js"></script>
```

```
<script src =  
"src/js/sound/enterButtonSound.js"></script>
```

```
<!-- structures -->
```

```
<script src =  
"src/js/structures/updateStructures.js"></script>
```

```
<script src =  
"src/js/structures/structuresColorAll.js"></script>
```

```
<script src =  
"src/js/structures/structuresCreate.js"></script>
```

```
<script src =  
"src/js/structures/structuresCreateFromFile.js"></script>
```

```
<script src =  
"src/js/structures/structuresMenu.js"></script>
```

```
<script src =  
"src/js/structures/structuresRemoveAll.js"></s  
cript>
```

```
<script src =  
"src/js/structures/structuresRow.js"></script>
```

```
<script src =  
"src/js/structures/structuresGoTo.js"></script>
```

```
<script src =  
"src/js/whenLoaded/whenLoaded.js"></script>
```

```
</head>
```

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

```
</body>
```

```
</html>
```

```
/* style001.css */
```

```
hr
```

```
{  
    margin-top: 4px;  
    margin-bottom: 4px;  
    border: 0 none;  
    padding-top: 0.5px;  
    padding-bottom: 0.5px;  
    background-color: rgb(255, 255, 255);  
}
```

```
a:link
```

```
{  
    color:rgb(110, 162, 255);  
}
```

```
a:visited
```

```
{  
    color:rgb(133, 134, 209);  
}
```

```
b
```

```
{  
    color: aqua;  
    font-weight: 900;
```

```
}  
  
.divStyle001  
{  
    padding-left: 10px;  
    padding-right: 10px;  
    padding-top: 4px;  
    padding-bottom: 4px;  
    border-style: solid;  
    border-width: 2px;  
    border-color: rgba(100, 100, 255, 1.0);  
    border-radius: 10px;  
    background-color: rgb(0, 0, 0);  
    z-index: 998;  
    font-family: arial;  
    font-weight: bold;  
    font-size: large;  
    color: rgb(255, 255, 255);  
    text-align: center;  
    overflow-y: scroll;  
}
```

```
.divBackground  
{  
    background: black;
```

```
background-image:  
url("https://collegeofscripting.weebly.com/uplo  
ads/6/4/4/8/64482293/stars-nice-  
edited_orig.png");  
/* background-repeat: repeat; */  
}
```

.ourButton

```
{  
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;  
}
```

```
.ourButton:hover  
{  
  background-color: rgb(6, 31, 47);  
  border-color: rgb(0, 255, 255);  
  color: rgb(0, 255, 255);  
}
```

```
.ourButton:active  
{  
  color: rgb(255, 0, 255);  
}
```

```
.hrCompact  
{  
  background-color: rgb(255,255,255);  
  border: 0 none;  
  margin-top: 4px;  
  margin-bottom: 4px;  
  padding-top: 0.5px;  
  padding-bottom: 0.5px;  
}
```

```
.textCompact
```

```
{  
  font-family: arial;  
  font-size: 11pt;  
  font-weight: normal;  
  color: rgb(255,255,255);  
}
```

.theImage

```
{  
  height: 75%;  
  object-fit: contain;  
  vertical-align: middle;  
  z-index: 10;  
}
```

.theSciencelImages

```
{  
  max-width: 100%;  
  height: auto;  
  object-fit: contain;  
  margin-left: auto;  
  margin-right: auto;  
  display: block;  
  z-index: 10;  
}
```



```
.vidStyle
```

```
{  
  transition: transform .2s;  
}
```

```
/* zoom feature for videos to be added */
```

```
.vidStyle:hover
```

```
{  
  /* transform: scale(2.5); */  
  position: absolute;  
  transform: translateY(+79%) scale(2.5);  
}
```

```
.bookImage
```

```
{  
  object-fit: contain;  
  max-width: 100%;  
  vertical-align: middle;  
  z-index: 10;  
}
```

```
/* zoom feature to be added */
```

```
.bookImage:hover
```

```
{  
  /* position: "absolute"; */  
  /* transform: scale(1.2); */
```

```

    z-index:100;
    /* transform: translateY(+79%) scale(2.5); */
    /* transform: translateY(+5%) scale(1.2); */
}

```

.textInput

```

{
    height: 20px;
    width: 100px;
    overflow: hidden;
    resize: none;
}

```

.glowBlue

```

{
    position: absolute;
    left: 20px;
    top: 20px;
    font-size: 60px;
    font-weight: bold;
    text-shadow: 0 0 5px #FFF, 0 0 10px #FFF, 0
0 15px #FFF, 0 0 20px #18dcff, 0 0 30px
#18b6ff, 0 0 40px #003cff, 0 0 55px #00d9ff, 0 0
75px #00e1ff, 2px 2px 2px rgba(115,246,255,0);
    z-index: 1;
}

```

.glowRed

```
{
  position: absolute;
  left: 55px;
  top: 100px;
  font-size: 40px;
  font-weight: bold;
  text-shadow: 0 0 5px #FFF, 0 0 10px #FFF, 0
0 15px #FFF, 0 0 20px #ff1878, 0 0 30px
#0ddbff, 0 0 40px #21f0ff, 0 0 55px #ff008c, 0 0
75px #ff168b, 2px 2px 2px rgba(115, 246, 255,
0);
  z-index: 1;
}
```

.buttonBottomToolbar

```
{
  color: white;
}
```

summary

```
{
  list-style-type: '^';
  margin-left: 4px;
  font-size: 16px;
```

```
font-weight: bold;  
border: none;  
cursor: pointer;  
}
```

#instructions

```
{  
    margin: 4px;  
    padding-left: 10px;  
    padding-right: 10px;  
    padding-top: 4px;  
    padding-bottom: 4px;  
    border: solid 2px rgba(100, 100, 255, 1.0);  
    border-radius: 10px;  
    background-color: rgb(0, 0, 0);  
    z-index: 998;  
    font-family: arial;  
    font-weight: bold;  
    color: rgb(255, 255, 255);  
    text-align: center;  
    line-height: 20px;  
    overflow-y: scroll;  
}
```

```
/* -- */
```

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

```
// 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 textClear(whichId)  
{  
    ge(whichId).innerHTML = "";  
}
```

```
//--//
```

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

```
// playerArray.js
```

```
playerArray = [  
{  
    name: "Christopher",  
    id: "thePlayer",  
    x: 140,
```

```
y: 60,  
width: 90,  
height: 250,  
speedMultiplier: 7,  
colorFill: "rgb(73, 100, 150)",  
colorText: "rgb(255, 255, 255)",  
opacity: 1.0,  
texture: 'url("src/media/textures/jane.png")',  
}
```

```
];
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// structuresArray.js
```

```
structuresArray = [  
{
```

```
  "name": "House 1",
```



```
"id": "1",  
"posX": 1100,  
"posY": 400,  
"sizeX": 110,  
"sizeY": 45,  
"colorFill": "black",  
"colorText": "white",  
"opacity": 1,  
"texture": "Default"  
},
```

```
{  
  "name": "House 2",  
  "id": "2",  
  "posX": 1200,  
  "posY": 500,  
  "sizeX": 110,  
  "sizeY": 45,  
  "colorFill": "black",  
  "colorText": "white",  
  "opacity": 1,  
  "texture": "Default"  
},
```

```
{  
  "name": "Our Third House",
```

```
"id": "3",  
"posX": 1400,  
"posY": 500,  
"sizeX": 110,  
"sizeY": 45,  
"colorFill": "black",  
"colorText": "white",  
"opacity": 1,  
"texture": "Default"  
},  
  
{  
  "name": "House 4",  
  "id": "4",  
  "posX": 850,  
  "posY": 300,  
  "sizeX": 114,  
  "sizeY": 49,  
  "colorFill": "rgb(150, 50, 100)",  
  "colorText": "rgb(255, 255, 255)",  
  "opacity": "1",  
  "texture": "none"  
},  
  
{  
  "name": "House 5",
```

```
"id": "5",  
"posX": 800,  
"posY": 500,  
"sizeX": 114,  
"sizeY": 49,  
"colorFill": "rgb(0, 0, 0)",  
"colorText": "rgb(100, 200, 255)",  
"opacity": "1",  
"texture": "none"  
},  
  
{  
  "name": "House 6",  
  "id": "6",  
  "posX": 1900,  
  "posY": 500,  
  "sizeX": 114,  
  "sizeY": 49,  
  "colorFill": "rgb(0, 0, 0)",  
  "colorText": "rgb(255, 255, 255)",  
  "opacity": "1",  
  "texture":  
    "url('https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/concrete1.jpg')"  
},
```

```
{  
  "name": "House 7",  
  "id": "7",  
  "posX": 1800,  
  "posY": 400,  
  "sizeX": 114,  
  "sizeY": 49,  
  "colorFill": "rgb(0, 0, 0)",  
  "colorText": "rgb(255, 255, 255)",  
  "opacity": "1",  
  "texture": "none"  
},
```

```
{  
  "name": "House 8a",  
  "id": "8",  
  "posX": 2000,  
  "posY": 500,  
  "sizeX": 114,  
  "sizeY": 49,  
  "colorFill": "rgb(0, 0, 0)",  
  "colorText": "rgb(255, 255, 255)",  
  "opacity": "1",  
  "texture": "none"  
}
```

];

//--//

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew

Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// worldVariables.js

let online = false;

let backgroundSizeX = 250000;

let backgroundSizeY = 250000;

let controlFlag = 1;

let activePlayer = "thePlayer";

let playerNumber = 0;

let structuresArray = [];

```
let ourPlayer =  
{  
  x: playerArray[playerNumber].x,  
  y: playerArray[playerNumber].y,  
  width: playerArray[playerNumber].width,  
  height: playerArray[playerNumber].height,  
  speedMultiplier: 2,  
  id: playerArray[playerNumber].id,  
  playerColor:  
playerArray[playerNumber].colorFill,  
  texture: `url($  
{playerArray[playerNumber].texture})`,  
  texturePos: '0% 0%',  
  velocityY: 0,  
};
```

```
let colorChoices = [  
  { name: 'aqua', color: '0, 255, 255' },  
  { name: 'black', color: '0, 0, 0' },  
  { name: 'blue', color: '0, 0, 255' },  
  { name: 'fuchsia', color: '255, 0, 255' },  
  { name: 'gray', color: '128, 128, 128' },  
  { name: 'green', color: '0, 128, 0' },  
  { name: 'lime', color: '0, 255, 0' },  
  { name: 'maroon', color: '128, 0, 0' },  
  { name: 'navy', color: '0, 0, 128' },
```

```

{ name: 'olive', color: '128, 128, 0' },
{ name: 'purple', color: '128, 0, 128' },
{ name: 'red', color: '255, 0, 0' },
{ name: 'silver', color: '192, 192, 192' },
{ name: 'teal', color: '0, 128, 128' },
{ name: 'white', color: '255, 255, 255' },
{ name: 'yellow', color: '255, 255, 0' },
{ name: 'orange', color: '255, 165, 0' },
{ name: 'pink', color: '255, 192, 203' },
{ name: 'brown', color: '165, 42, 42' },
{ name: 'violet', color: '238, 130, 238' },
{ name: 'cyan', color: '0, 255, 255' },
{ name: 'magenta', color: '255, 0, 255' },
{ name: 'limegreen', color: '50, 205, 50' },
{ name: 'gold', color: '255, 215, 0' },
{ name: 'coral', color: '255, 127, 80' },
{ name: 'indigo', color: '75, 0, 130' },
{ name: 'khaki', color: '240, 230, 140' },
{ name: 'plum', color: '221, 160, 221' },
{ name: 'salmon', color: '250, 128, 114' },
{ name: 'tan', color: '210, 180, 140' },
{ name: 'turquoise', color: '64, 224, 208' },
{ name: 'wheat', color: '245, 222, 179' }
];

```

//--//

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

```
// 1contentWorldVariables.js
```

```
// content starts at specified y Position  
let yPos = 600;
```

```
let space = 800;
```

```
// content is Positioned in Rows  
// based on the Functions Place in the Array  
let rowNumber = 1;
```

```
//--//
```

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



```
//  
https://github.com/ChristopherAndrewTopalian  
  
// containerBooks.js  
  
function containerBooks()  
{  
    let contentArray =  
    [  
        function()  
        {  
            // theX, theY, sizeX, sizeY, whichArray,  
criteria1, criteria2  
            rowOfSciencelImagesAndLink(100,  
yPos, 600, 600, sciencelImages, "electricity",  
"electron flow")  
        },  
  
        function()  
        {  
            // theX, theY, whichArray  
            rowOfCollegeLinks(100, yPos,  
collegeLinks)  
        },  
  
        function()  
    ]  
}
```

```
{  
    // theX, theY, sizeX, sizeY, whichArray  
    rowOfCollegeBooks(100, yPos, 400,  
400, collegeBooks)  
},  
  
function()  
{  
    // theX, theY, sizeX, sizeY, whichArray,  
criteria1, theHeading  
    rowOfBookImages(100, yPos, 600, 600,  
mathematics, "book", "Mathematics",  
"texture")  
}  
];  
  
for (i = 0; i < contentArray.length; i++)  
{  
    // adds one row each time through loop  
contentArray[i]();  
  
    // add 900 to the yPos, between new  
section below  
yPos = yPos + space;  
}
```

```
// spacer sector 1 line  
makeLine(yPos, rowNumber, "Cross this  
Line to Load Content");
```

```
    rowNumber += 1;  
}
```

```
//--//
```

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

```
// containerBoxingVideos.js
```

```
function containerBoxingVideos()  
{  
    let contentArray =  
    [  
        function()  
        {  
            // theX, theY, sizeX, sizeY, whichArray,  
criteria1, criteria2
```

```

        rowOfVideosMultipleFilters(100, yPos,
800, 600, boxingVideos,"boxing",
"furyWilder1")
    },

    function()
    {
        // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
        rowOfVideosMultipleFilters(100, yPos +
200, 800, 600, boxingVideos, "boxing",
"furyWilder2")
    },
];

for (i = 0; i < contentArray.length; i++)
{
    contentArray[i]();
    yPos = yPos + space;
}

// spacer sector line
// makeLine(yPos, rowNumber, "Cross this
Line to Load Content");

rowNumber += 1;

```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// containerLanguage.js
```

```
function containerLanguage()
```

```
{
```

```
    let contentArray =
```

```
    [
```

```
        function()
```

```
        {
```

```
            if (online == false)
```

```
            {
```

```
                rowOfWebsitesOrVideos(100, yPos,  
900, 400, computerSciencePdf, 'pdf')
```

```
            }
```

```
        },
```

```
function()  
{  
    // theX, theY, sizeX, sizeY, whichArray,  
criteria1, criteria2  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "array", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "string", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "math", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "date", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "boolean", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "global", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "number", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "regular expression",  
"function")  
},
```

```
function()
```

```
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "classes", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "error", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "json", "function")  
},
```

```
function()  
{  
    rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "console", "function")  
},
```

```
function()  
{
```



```
        rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "history", "function")  
    },  
  
    function()  
    {  
        rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "storage", "function")  
    },  
  
    function()  
    {  
        rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "geolocation", "function")  
    },  
  
    function()  
    {  
        rowOfLanguage(100, yPos, 400, 500,  
javascriptLanguage, "string", "function")  
    },  
  
    function()  
    {  
        // whichArray, theX, theY, sizeX, sizeY,  
catetory, tag
```

```
        rowOfLanguageShowAll(100, yPos +
300, 400, 500, javascriptLanguage, "function",
"JavaScript")
    },

    // Linden Scripting Language - Show All
function()
{
    // rowOfLanguageShowAll(100, yPos,
400, 500, lindenLanguage, "function",
"Linden")
},
];

for (i = 0; i < contentArray.length; i++)
{
    // adds one row each time through loop
contentArray[i]();

    // add space to yPos, between new section
below
    yPos = yPos + space;
}

// spacer sector line
```

```
makeLine(yPos, rowNumber, "Cross this  
Line to Load Content");
```

```
    rowNumber += 1;  
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// containerNews.js
```

```
function containerNews()
```

```
{
```

```
    let contentArray =
```

```
[
```

```
    function()
```

```
    {
```

```
        // theX, theY, sizeX, sizeY, whichArray,
```

```
criteria1
```

```
        rowOfNews(100, yPos, 800, 250,  
theNews, "news")  
    },
```

```
    function()  
    {  
        // theX, theY, sizeX, sizeY,whichArray,  
criteria1  
        rowOfEvidence(100, yPos, 700, 300,  
evidence, "fact")  
    },
```

```
    function()  
    {  
        // theX, theY, sizeX, sizeY,whichArray,  
criteria1  
        rowOfSocialMediaEvidence(100, yPos,  
700, 500, socialMediaEvidence, "twitter")  
    },
```

```
    function()  
    {  
        // theX, theY, sizeX, sizeY, whichArray,  
criteria1  
        rowOfDetails(100, yPos, 700, 500,  
socialMediaEvidence, "twitter")
```

```
},
```

```
function()  
{
```

```
    // theX, theY, sizeX, sizeY, whichArray,  
criteria1, theTitle
```

```
rowOfEvidenceWithImage(100,yPos,1250,600,  
evidence, "fact", "Evidence")  
},
```

```
function()  
{
```

```
    // theX, theY, sizeX, sizeY, whichArray,  
criteria1
```

```
    rowOfNotes(100, yPos, 400, 400,  
theNotes, "note")  
},
```

```
function()  
{
```

```
    // theX, theY, sizeX, sizeY, whichArray,  
criteria1
```

```
    rowOfNotesWithTextArea(100, yPos,  
600, 500, theNotes, "note")  
},
```

```
// Investigation
function()
{
    rowOfInvestigation(100, yPos, 400, 400,
petitoGabrielleInvestigation, "note")
    },
];

for (i = 0; i < contentArray.length; i++)
{
    // adds one row each time through loop
    contentArray[i]();

    // add 900 to the yPos, between new
section below
    yPos = yPos + space;
}

// spacer sector line
makeLine(yPos, rowNumber, "Cross this
Line to Load Content");

rowNumber += 1;
}
```

//--//

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// containerNewsVideos.js

function containerNewsVideos()

{

let contentArray =

[

function()

{

// theX, theY, sizeX, sizeY, whichArray,

criteria1

**rowOfWebsitesOrVideos(100, yPos, 700,
500, videoEvidence, "youtubeVideo")**

},

function()

{

```

        // theX, theY, sizeX, sizeY, whichArray,
criteria1
        rowOfVideos(100, yPos, 800, 600,
videoEvidence, "exercise")
    },

    function()
    {
        // theX, theY, sizeX, sizeY, whichArray,
criteria1
        rowOfVideos(100, yPos, 800, 600,
videoEvidence, "mask")
    },

    function()
    {
        // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
        makeVideosOnlineStartEnd(100, yPos,
800, 600, videoEvidence, "youtubeVideo",
"exercise")
    },
];

for (i = 0; i < contentArray.length; i++)
{

```



```
// adds one row each time through loop  
contentArray[i]();
```

```
// add 900 to yPos, between new section  
below
```

```
yPos = yPos + space + 200;  
}
```

```
// spacer sector line  
makeLine(yPos, rowNum, "Cross this  
Line to Load Content");
```

```
rowNum += 1;  
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// containerScience.js
```

```
function containerScience()
{
    let contentArray =
    [
        function()
        {
            // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
            rowOfVideosMultipleFilters(100, yPos,
800, 600, collegeVideos,"youtubeVideo",
"javascript")
        },

        function()
        {
            // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
            rowOfVideosMultipleFilters(100, yPos,
800, 600, collegeVideos,"youtubeVideo",
"papercliptronics")
        },

        function()
        {
            // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
```

```
        rowOfVideosMultipleFilters(100, yPos,
800, 600, collegeVideos, "youtubeVideo",
"breadboard")
    },

    function()
    {
        // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
        makeVideosOnlineStartEnd(100, yPos,
800, 600, techVideos, "youtubeVideo", "electric
motor")
    },

    function()
    {
        // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
        makeVideosOnlineStartEnd(100, yPos,
800, 600, techVideos, "youtubeVideo", "water
pump")
    },

    function()
    {
```

```

        // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
        makeVideosOnlineStartEnd(100, yPos,
800, 600, techVideos, "youtubeVideo", "air
pump")
    },

```

```

function()
{
    // theX, theY, sizeX, sizeY, whichArray,
criteria1, criteria2
    makeVideosOnlineStartEnd(100, yPos,
800, 600, techVideos, "youtubeVideo",
"microphone")
}
];

```

```

for (i = 0; i < contentArray.length; i++)
{
    // adds one row each time through loop
contentArray[i]();

    // add 900 to the yPos, between new
section below
    yPos = yPos + space + 200;
}

```

```
// spacer sector 1 line  
makeLine(yPos, rowNum, "Cross this  
Line to Load Content");
```

```
    rowNum += 1;  
}
```

```
//--//
```

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

```
// contentContainersLineCrossed.js
```

```
// load Content based on the player's position.  
let sector = 1;
```

```
function wasLineCrossed(whichId)  
{  
    let playerElement = ge(whichId);
```

```
// if player exists
if (playerElement)
{
    // player speed can be very fast, so we
    allow a large range of Y value to be able to
    trigger the event

    if (sector == 1)
    {
        if (Math.floor(positionGet(whichId).y) >
yPos && Math.floor(positionGet(whichId).y) <
yPos + 70)
        {
            ge('infoDiv').innerHTML = yPos + "
Crossed" + '<br>' + 'Sector: ' + sector;

            // loads the content
            containerNews();

            // removes the line
            removeElement("sector1");

            // only allow content to be added one
time
            sector = 2;
        }
    }
}
```

```

    }

    if (sector == 2)
    {
        if (Math.floor(positionGet(whichId).y) >
yPos && Math.floor(positionGet(whichId).y) <
yPos + 70)
        {
            ge('infoDiv').innerHTML = yPos + '
Crossed' + '<br>' + 'Sector: ' + sector;

            containerBooks();
            //containerScience();

            // removes line
            removeElement("sector2");

            // only allow content to be added one
time
            sector = 3;
        }
    }

    if (sector == 3)
    {

```

```

        if (Math.floor(positionGet(whichId).y) >
yPos && Math.floor(positionGet(whichId).y) <
yPos + 70)
        {
            ge('infoDiv').innerHTML = yPos + '
Crossed' + '<br>' + 'Sector: ' + sector;

            // loads the content
            containerScience();

            // removes the line
            removeElement("sector3");

            //only allow content to be added one
time
            sector = 4;
        }
    }

    if (sector == 4)
    {
        if (Math.floor(positionGet(whichId).y) >
yPos && Math.floor(positionGet(whichId).y) <
yPos + 70)
        {

```



```

    ge('infoDiv').innerHTML = yPos + '
Crossed' + '<br>' + 'Sector: ' + sector;

```

```

    // loads the content
    containerNewsVideos();

```

```

    // removes the line
    removeElement("sector4");

```

```

    // only allow content to be added one
time

```

```

    sector = 5;

```

```

    }

```

```

}

```

```

if (sector == 5)
{

```

```

    if (Math.floor(positionGet(whichId).y) >
yPos && Math.floor(positionGet(whichId).y) <
yPos + 70)

```

```

{

```

```

    ge('infoDiv').innerHTML = yPos + '
Crossed' + '<br>' + 'Sector: ' + sector;

```

```

    // loads the content
    containerBoxingVideos();

```

```

// removes the line
removeElement("sector5");

// only allow content to be added one
time
sector = 6;
}
}

if (sector == 6)
{
    if (Math.floor(positionGet(whichId).y) >
yPos && Math.floor(positionGet(whichId).y) <
yPos + 70)
    {
        ge('infoDiv').innerHTML = yPos + '
Crossed' + '<br>' + 'Sector: ' + sector + '\n' +
'More Content Soon';

// containerVi();

// removes the line
removeElement("sector6");

```

```
// only allow content to be added one  
time
```

```
    sector = 7;  
    }  
  }  
}
```

```
//--//
```

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

```
// downloadData.js
```

```
// creates a text file named structureData.js.  
function downloadStructureData()  
{  
    // get the data from the Text Area  
    // let data =  
ge("textAreaStructureData").value;
```

```
// get data from structuresArray format it
let ourData = "structuresArray=" +
JSON.stringify(structuresArray, null, ' ') + ";" +
"\n\n";

// create an anchor element
let ourElement = ce("a");

// we can choose .txt .html or in our case, we
use .js
ourElement.download = "structuresArray.js";

// we choose the data and kind
let theKind = new Blob([ourData],
{
    type: "text/plain"
});

ourElement.href =
window.URL.createObjectURL(theKind);

ourElement.click();
}

function downloadPlayerData()
{
```

```
// get the data from the Text Area
// let data = ge("textAreaPlayerData").value;

// get data from playerArray, format it
let ourData = "playerArray = " +
JSON.stringify(playerArray, null, ' ') + ";" + "\n\n";

// create an anchor element
let ourElement = ce("a");

// we can choose .txt .html or in our case, we
use .js
ourElement.download = "playerData.js";

// we choose the data and kind
let theKind = new Blob([ourData],
{
  type: "text/plain"
});

ourElement.href =
window.URL.createObjectURL(theKind);

ourElement.click();
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// playerLoadData.js
```

```
// playerArray is used to load player settings  
function playerLoadData(whichId)
```

```
{
```

```
    let playerElement = ge(whichId);
```

```
    if (playerElement)
```

```
    {
```

```
        if (playerArray[0].texture != "none")
```

```
        {
```

```
            // texture the player
```

```
            playerElement.style.backgroundImage =
```

```
playerArray[0].texture;
```

```
        }
```

```
    else
```

```
{  
    // else if texture is not used  
    // background Color of the player  
    playerElement.style.background =  
playerArray[0].colorFill;  
}  
  
    // opacity  
    playerElement.style.opacity =  
playerArray[0].opacity;  
  
    // text color  
    playerElement.style.color =  
playerArray[0].colorText;  
  
    // X position  
    playerElement.style.left = playerArray[0].x;  
  
    // Y position  
    playerElement.style.top = playerArray[0].y;  
  
    // width  
    playerElement.style.width =  
playerArray[0].width;  
  
    // height
```

```
        playerElement.style.height =  
playerArray[0].height;  
    }  
}
```

```
//--//
```

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

```
// recordDelete.js
```

```
function deleteThisRecord(whichDiv)  
{  
    let warningDelete = confirm("DELETE ONLY  
This Record?");  
  
    if (warningDelete == true)  
    {  
        ge(whichDiv).outerHTML = "";  
    }  
    else if (warningDelete == false)
```



```
{  
    ge('infoDiv').textContent = "Canceled";  
}  
}
```

```
//--//
```

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

```
// showPlayerData.js
```

```
// Show the playerArray data in a Text Area that  
// is scrollable
```

```
function showPlayerData()  
{  
    ge("textAreaPlayerData").value =  
    "playerArray = " + JSON.stringify(playerArray,  
    null, ' ') + ";" + "\n\n" ;  
}
```

//--//

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// showStructureData.js

// show the structuresArray data

// in a Text Area that is scrollable

function showStructureData()

{

ge("textAreaStructureData").value =

"structuresArray = " +

JSON.stringify(structuresArray, null, ' ') + ";" +

"\n\n" ;

}

//--//

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

```
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalian

// sortByDate.js

// Ascending order uses a.date - b.date
// Descending order uses b.date - a.date
// ascending or descending
function sortByDate(whichArray, direction)
{
    // ASCENDING is a.date - b.date
    if (direction == "up")
    {
        whichArray.sort(function(a, b)
        {
            return new Date(a.date) - new
Date(b.date);
        });
    }
    // DESCENDING is b.date - a.date
    else if (direction == "down")
    {
        whichArray.sort(function(a, b)
        {
```

```

        return new Date(b.date) - new
Date(a.date);
    });
}
}

```

// examples

```

// Sort by date
// sortByDate(theNotes, "up"); // Ascending
// sortByDate(theNotes, "down"); //
Descending// sortByNumber.js

```

```

function sortByNumber(whichArray, direction)
{
    if (direction == "up") // ASCENDING
    {
        whichArray.sort(function(a, b)
        {
            return
a.number.localeCompare(b.number);
        });
    }
    else if (direction == "down") // DESCENDING
    {
        whichArray.sort(function(b, a)

```

```
    {  
        return  
a.number.localeCompare(b.number);  
    });  
}  
}
```

// examples

```
// Sort by number  
// sortByNumber(theNotes, "up"); // Ascending  
// sortByNumber(theNotes, "down"); //  
Descending// sortData.js
```

// we include which arrays we want to sort

```
function sortData()  
{  
    // ascending  
    sortByDate(booksPdf, "up");  
  
    // ascending  
    sortByDate(petitoGabrielleInvestigation,  
"up");  
  
    // descending  
    sortByDate(theNews, "down");
```

```
// descending  
sortByDate(evidence, "down");
```

```
// descending  
sortByDate(videoEvidence, "down");
```

```
// ascending  
sortByDate(socialMediaEvidence, "up");
```

```
// descending  
sortByDate(boxingVideos, "down");
```

```
// descending  
sortByDate(collegeVideos, "down");
```

```
// ascending  
sortByDate(ourVideoFiles, "up");
```

```
// sort by Multiple Criteria, category and  
name
```

```
    javascriptLanguage.sort(function(a, b)  
    {  
        return  
a.category.localeCompare(b.category) ||  
b.name - a.name;
```

```
});
```

```
// sort by Multiple Criteria, category and  
name
```

```
lindenLanguage.sort(function(a, b)  
{  
    return  
a.category.localeCompare(b.category) ||  
b.name - a.name;  
});
```

```
// sort by one number only, number
```

```
theNotes.sort(function(a, b)  
{  
    return a.number.localeCompare(b.number,  
'en-US', {numeric:"true"});  
    // || b.note - a.note;  
});
```

```
// sort by one number only, probability
```

```
socialMediaEvidence.sort(function(a, b)  
{  
    return  
a.probability.localeCompare(b.probability, 'en-  
US', {numeric:"true"}); // || b.title - a.title;  
});
```

**// to sort by additional criteria, just use
another ||**

**// sort By Date - ASCENDING
scienceImages.sort(function(a,b)
{
 return new Date(a.date) - new Date(b.date);
});
}**

//--//

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

// cameraFollowsPlayer.js

**// camera starts active
let cameraFlag = 1;**

// camera Follows the Player


```
function cameraFollowsPlayer(whichId)  
{  
    if (cameraFlag == 1)  
    {  
        let playerElement = ge(whichId);  
  
        if (playerElement)  
        {  
            playerElement.scrollToView(  
                {  
                    block: "center", inline: "center"  
                });  
        }  
    }  
  
    if (cameraFlag == 0)  
    {  
        // get position of where the person clicks  
their mouse arrow  
        mouseOn();  
  
        return;  
    }  
}
```

//--//

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

```
// gameLoop.js
```

```
function gameLoop(whichId)
{
    playerMotion(activePlayer);

    cameraFollowsPlayer(activePlayer);

    keepPlayerInWorld(activePlayer);

    wasLineCrossed(activePlayer);

    // getDistanceXY('thePlayer',
'Structure1');

    requestAnimationFrame(gameLoop);
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// keepPlayerInWorld.js
```

```
function keepPlayerInWorld(whichId)
```

```
{
```

```
    if (ge(whichId))
```

```
    {
```

```
        // if left
```

```
        if (ourPlayer.x <= 0)
```

```
        {
```

```
            ourPlayer.x = 0;
```

```
        }
```

```
        // if right
```

```
        if (ourPlayer.x > backgroundSizeX)
```

```
        {
```

```
            ourPlayer.x = backgroundSizeX;
```

```
        }
```

```
// if top
if (ourPlayer.y <= 0)
{
    ourPlayer.y = 0;
}

// if bottom
if (ourPlayer.y >= backgroundSizeY)
{
    ourPlayer.y = backgroundSizeY;
}
}
```

//--//

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

// keyboardControl.js

```
// used when a person is done with a textarea
function keyboardControlOn()
{
    controlFlag = 1;
}
```

```
// used when person is typing in a textarea
function keyboardControlOff()
{
    controlFlag = 0;
}
```

```
function hideShow(whichId)
{
    ge(whichId).style.transform = 'translate(-
50px, 0px)';

    setTimeout(function()
    {
        ge(whichId).style.transform =
'translate(0px, 0px)';
    }, 1000);
}
```

```
function theControls(e)
{
```

```
if (controlFlag ==1)
{
    // Letter z
    if (e.keyCode == 90)
    {
        // open player menu
        menuPlayer();
    }

    // Letter x
    if (e.keyCode == 88)
    {
        // listen for mouse clicks
        window.addEventListener("click",
mousePos, false);

        ge('infoDivPlayer').innerHTML = "Left
Click for <br>Mouse Position";
    }

    // Letter v
    if (e.keyCode == 86)
    {
        // remove listen for mouse clicks
        window.removeEventListener("click",
mousePos, false);
    }
}
```

```

    ge('infoDivPlayer').innerHTML = "";
}

// SPEED

// tilda
if(e.keyCode == 192)
{
    ourPlayer.speedMultiplier = 0.5;

    ge('infoDivPlayer').innerHTML = "Speed
Tilda";

    hideShow('infoDivPlayer');
}

// number 1, or numpad 1
if (e.keyCode == 49 || e.keyCode == 97)
{
    ourPlayer.speedMultiplier = 1;

    ge('infoDivPlayer').innerHTML = "Speed
1";

    hideShow('infoDivPlayer');
}

```

```
}
```

```
// number 2
```

```
if (e.keyCode == 50 || e.keyCode == 98)
```

```
{
```

```
    ourPlayer.speedMultiplier = 2;
```

```
    ge('infoDivPlayer').innerHTML = "Speed  
2";
```

```
    hideShow('infoDivPlayer');
```

```
}
```

```
// number 3
```

```
if (e.keyCode == 51 || e.keyCode == 99)
```

```
{
```

```
    ourPlayer.speedMultiplier = 5;
```

```
    ge('infoDivPlayer').innerHTML = "Speed  
3";
```

```
    hideShow('infoDivPlayer');
```

```
}
```

```
//number 4
```

```
if (e.keyCode == 52 || e.keyCode == 100)
```



```
{  
    ourPlayer.speedMultiplier = 10;  
    ge('infoDivPlayer').innerHTML = "Speed  
4";  
  
    hideShow('infoDivPlayer');  
}
```

```
// number 5  
if (e.keyCode == 53 || e.keyCode == 101)  
{  
    ourPlayer.speedMultiplier = 15;  
    ge('infoDivPlayer').innerHTML = "Speed  
5";  
  
    hideShow('infoDivPlayer');  
}
```

```
// number 6  
if (e.keyCode == 54 || e.keyCode == 102)  
{  
    ourPlayer.speedMultiplier = 20;  
    ge('infoDivPlayer').innerHTML = "Speed  
6";  
  
    hideShow('infoDivPlayer');
```

```
}
```

```
// number 7
```

```
if (e.keyCode == 55 || e.keyCode == 103)
```

```
{
```

```
    ourPlayer.speedMultiplier = 25;
```

```
    ge('infoDivPlayer').innerHTML = "Speed
```

```
7";
```

```
    hideShow('infoDivPlayer');
```

```
}
```

```
// number 8
```

```
if (e.keyCode == 56 || e.keyCode == 104)
```

```
{
```

```
    ourPlayer.speedMultiplier = 30;
```

```
    ge('infoDivPlayer').innerHTML = "Speed
```

```
8";
```

```
    hideShow('infoDivPlayer');
```

```
}
```

```
// number 9
```

```
if (e.keyCode == 57 || e.keyCode == 105)
```

```
{
```

```
    ourPlayer.speedMultiplier = 40;
```

```

ge('infoDivPlayer').innerHTML = "Speed
9";

hideShow('infoDivPlayer');
}

// number 0
if (e.keyCode == 48 || e.keyCode == 96)
{
    ourPlayer.speedMultiplier = 50;
    ge('infoDivPlayer').innerHTML = "Speed
0";

    hideShow('infoDivPlayer');
}

// dash
if (e.keyCode == 173)
{
    ourPlayer.speedMultiplier = 100;
    ge('infoDivPlayer').innerHTML = "Speed
-";

    hideShow('infoDivPlayer');
}

```

```
// +
if (e.keyCode == 61)
{
    ourPlayer.speedMultiplier = 200;
    ge('infoDivPlayer').innerHTML = "Speed
+";

    hideShow('infoDivPlayer');
}

// Back Space button
if (e.keyCode == 8)
{
    ourPlayer.speedMultiplier = 400;
    ge('infoDivPlayer').innerHTML = "Speed
Backspace";

    hideShow('infoDivPlayer');
}

// Insert Button
if (e.keyCode == 45)
{
    ourPlayer.speedMultiplier = 600;
    ge('infoDivPlayer').innerHTML = "Speed
Insert";
```

```
        hideShow('infoDivPlayer');
    }

    // Home Button
    if (e.keyCode == 36)
    {
        ourPlayer.speedMultiplier = 800;
        ge('infoDivPlayer').innerHTML = "Speed
Home";

        hideShow('infoDivPlayer');
    }

    // PageUp Button
    if (e.keyCode == 33)
    {
        ourPlayer.speedMultiplier = 1600;
        ge('infoDivPlayer').innerHTML = "Speed
PageUp";

        hideShow('infoDivPlayer');
    }

    //-//
```

```
// Letter b to add Structure
if (e.keyCode == 66)
{
    audioPlay("sfx_blip_001", 1.0);

    ge('infoDivPlayer').innerHTML =
"STRUC- <br>TURE<br> MODE<br>ON";

    // when left clicked, make a structure at
    the location of the mouse arrow
    window.addEventListener("click",
structuresCreate, false);
}

// Esc Button to exit Structure mode
// escape button
if (e.keyCode == 27)
{
    audioPlay("sfx_blip_001", 1.0);

    ge('infoDivPlayer').innerHTML =
"STRUC- <br>TURE<br> MODE<br>OFF";

    // Remove the event listener, that listens
    for mouse clicks. This deactivates structure
    creation mode
```

```
window.removeEventListener("click",  
structuresCreate);  
}  
  
// letter j  
if (e.keyCode == 74)  
{  
    // theX, theY, sizeX, sizeY, whichArray  
    structuresRow(ourPlayer.x, ourPlayer.y,  
125, 200, structuresArray);  
  
    audioPlay("sfx_blip_001", 1.0);  
}  
  
// Delete Button  
if (e.keyCode == 46)  
{  
    // audioPlay("sfx_blip_001", 1.0);  
  
    // we move the structures row offscreen,  
as a simple way of clearing the screen of the  
row.  
    structuresRow(-1000, -1000, 10, 10,  
structuresArray)  
  
    structuresRemoveAll();
```

```
ge('infoDivPlayer').innerHTML =  
"Removed <br>All<br> Structures";  
}
```

```
// STRUCTURE MODE
```

```
// Shift + 1 brings you to structure 1  
if(event.getModifierState("Shift") &&  
e.keyCode == 49)  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
    // get the position  
    let newPosition =  
positionGet(structuresArray[0].id);
```

```
    // update player position  
    ourPlayer.x = newPosition.x;  
    ourPlayer.y = newPosition.y;  
  
    movePlayer(activePlayer);  
}
```

```
// Shift + 2 brings you to structure 2
```



```
    if(event.getModifierState("Shift") &&
e.keyCode == 50)
    {
        audioPlay("sfx_blip_001", 1.0);

        // get the position
        let newPosition =
positionGet(structuresArray[1].id);

        // update player position
        ourPlayer.x = newPosition.x;
        ourPlayer.y = newPosition.y;

        movePlayer(activePlayer);
    }

    // Shift + 3 brings you to structure 3
    if(event.getModifierState("Shift") &&
e.keyCode == 51)
    {
        audioPlay("sfx_blip_001", 1.0);

        // get the position
        let newPosition =
positionGet(structuresArray[2].id);
```

```
// update player position  
ourPlayer.x = newPosition.x;  
ourPlayer.y = newPosition.y;
```

```
    movePlayer(activePlayer);  
}
```

```
// Shift + 4 brings you to structure 4  
if (event.getModifierState("Shift") &&  
e.keyCode == 52)  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
    // get the position  
    let newPosition =  
positionGet(structuresArray[3].id);
```

```
// update player position  
ourPlayer.x = newPosition.x;  
ourPlayer.y = newPosition.y;
```

```
    movePlayer(activePlayer);  
}
```

```
// Shift + 5 brings you to structure 5
```

```
    if (event.getModifierState("Shift") &&
e.keyCode == 53)
    {
        audioPlay("sfx_blip_001", 1.0);

        // get the position
        let newPosition =
positionGet(structuresArray[4].id);

        // update player position
        ourPlayer.x = newPosition.x;
        ourPlayer.y = newPosition.y;

        movePlayer(activePlayer);
    }

    // Shift + 6 brings you to structure 6
    if (event.getModifierState("Shift") &&
e.keyCode == 54)
    {
        audioPlay("sfx_blip_001", 1.0);

        // get the position
        let newPosition =
positionGet(structuresArray[5].id);
```

```
// update player position  
ourPlayer.x = newPosition.x;  
ourPlayer.y = newPosition.y;
```

```
    movePlayer(activePlayer);  
}
```

```
// Shift + 7 brings you to structure 7  
if(event.getModifierState("Shift") &&  
e.keyCode == 55)  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
    // get the position  
    let newPosition =  
positionGet(structuresArray[6].id);
```

```
// update player position  
ourPlayer.x = newPosition.x;  
ourPlayer.y = newPosition.y;
```

```
    movePlayer(activePlayer);  
}
```

```
// Shift + 8 brings you to structure 8
```

```
    if(event.getModifierState("Shift") &&
e.keyCode == 56)
    {
        audioPlay("sfx_blip_001", 1.0);

        // get the position
        let newPosition =
positionGet(structuresArray[7].id);

        // update player position
        ourPlayer.x = newPosition.x;
        ourPlayer.y = newPosition.y;

        movePlayer(activePlayer);
    }

    // Shift + 9 brings you to structure 9
    if (event.getModifierState("Shift") &&
e.keyCode == 57)
    {
        audioPlay("sfx_blip_001", 1.0);

        // get the position
        let newPosition =
positionGet(structuresArray[8].id);
```

```
// update player position  
ourPlayer.x = newPosition.x;  
ourPlayer.y = newPosition.y;
```

```
    movePlayer(activePlayer);  
}
```

```
// Shift + 0 brings you to structure 10  
if (event.getModifierState("Shift") &&  
e.keyCode == 48)  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
    // get the position  
    let newPosition =  
positionGet(structuresArray[9].id);
```

```
// update player position  
ourPlayer.x = newPosition.x;  
ourPlayer.y = newPosition.y;
```

```
    movePlayer(activePlayer);  
}
```

```
// if Control + F is pressed make  
cameraFlag 0
```

```
    if (event.getModifierState("Control") &&  
e.keyCode == 70)  
    {  
        cameraFlag = 0;  
    }
```

// TRANSPARENCY

// Letter m

```
if (e.keyCode == 77)  
{  
    ge(activePlayer).style.opacity = "1.0";  
}
```

// Letter n

```
if (e.keyCode == 78)  
{  
    ge(activePlayer).style.opacity = "0.2";  
  
    ge('infoDivPlayer').innerHTML= "";  
}
```

// Letter p for player color

```
if (e.keyCode == 80)  
{  
    menuPlayerColor();  
}
```

```
    audioPlay("sfx_blip_001", 1.0);
}

// Letter t for player texture
if (e.keyCode == 84)
{
    menuPlayerTexture();
    audioPlay("sfx_blip_001", 1.0);
}

// Letter o for player opacity
if (e.keyCode == 79)
{
    menuPlayerOpacity();
    audioPlay("sfx_blip_001", 1.0);
}

// Letter g clears player text
if (e.keyCode == 71)
{
    ge('infoDivPlayer').innerHTML = "";
    menuPlayer();
}
}
```



```
let theKeyCode = e.keyCode || e.which;

if (e.type === 'keydown')
{
    keyboard[theKeyCode] = true;
}
else if (e.type === 'keyup')
{
    keyboard[theKeyCode] = false;
}
}

//--//

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

// keyboardMotion.js

// keyboard letter codes being pressed
let keyboard = { };
```

```
keyboard.UP = 87; // 87 w, 38 up arrow  
keyboard.DOWN = 83; // 83 s, 40 down arrow  
keyboard.LEFT = 65; // 65 a, 37 left arrow  
keyboard.RIGHT = 68; // 68 d, 39 right arrow  
keyboard.SHIFT = 16; // 16 shift
```

// Player Controls

```
function playerMotion(whichId)  
{  
    let playerElement = ge(whichId);  
  
    // if person is not typing in a textarea,  
    controls are ON  
    if (controlFlag == 1)  
    {  
        if (keyboard[keyboard.LEFT])  
        {  
            movePlayer(whichId, -1, 0);  
  
            ge(ourPlayer.id).style.transform =  
'scaleX(-1)';  
        }  
        if (keyboard[keyboard.RIGHT])  
        {  
            movePlayer(whichId, 1, 0);
```

```
        ge(ourPlayer.id).style.transform =  
'scaleX(1)';  
    }  
    if (keyboard[keyboard.UP])  
    {  
        movePlayer(whichId, 0, -1);  
    }  
    if (keyboard[keyboard.DOWN])  
    {  
        movePlayer(whichId, 0, 1);  
    }  
  
    if (keyboard[keyboard.LEFT] &&  
keyboard[keyboard.SHIFT])  
    {  
        movePlayer(whichId, -2, 0);  
    }  
    if (keyboard[keyboard.RIGHT] &&  
keyboard[keyboard.SHIFT])  
    {  
        movePlayer(whichId, 2, 0);  
    }  
    if (keyboard[keyboard.UP] &&  
keyboard[keyboard.SHIFT])  
    {
```

```
        movePlayer(whichId, 0, -2);
    }
    if (keyboard[keyboard.DOWN] &&
keyboard[keyboard.SHIFT])
    {
        movePlayer(whichId, 0, 2);
    }
}
}
```

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

```
// movePlayer.js
```

```
function movePlayer(whichId, theX, theY)
{
    let playerElement = ge(whichId);

    if (playerElement)
    {
```

```
    ourPlayer.x += (theX || 0) *  
    ourPlayer.speedMultiplier;
```

```
    ourPlayer.y += (theY || 0) *  
    ourPlayer.speedMultiplier;
```

```
    playerElement.style.left = ourPlayer.x +  
'px';
```

```
    playerElement.style.top = ourPlayer.y +  
'px';  
    }  
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// get.js
```

```
function get(whichId, whichProperty)
```

```
{  
  let theId = ge(whichId);  
  
  if (whichProperty == 'name')  
  {  
    let theProperty = playerArray[0].name;  
    return theProperty;  
  }  
  
  if (whichProperty == 'innerHTML')  
  {  
    let theProperty = ge(whichId).innerHTML;  
    return theProperty;  
  }  
  
  if (whichProperty == 'id')  
  {  
    let theProperty = ge(whichId).id;  
    return theProperty;  
  }  
  
  if (whichProperty == 'speedMultiplier')  
  {  
    let theProperty =  
playerArray[0].speedMultiplier;  
  }  
}
```

```
    return theProperty;
}

if (whichProperty == 'size')
{
    let theRect =
theId.getBoundingClientRect();

    let sizeXY =
    {
        x: parseFloat(theRect.width).toFixed(3),
        y: parseFloat(theRect.height).toFixed(3)
    };

    return sizeXY;
}

if (whichProperty == 'pos')
{
    let theRect =
theId.getBoundingClientRect();

    let posXY = {
        x: theRect.x + theRect.width / 2 +
window.scrollX,
```

```
        y: theRect.top + theRect.height / 2 +  
window.scrollY  
    };  
  
    return posXY;  
}  
  
else  
{  
    let theProperty =  
window.getComputedStyle(theId).getPropertyV  
alue(whichProperty);  
  
    return theProperty;  
}  
}
```

```
// 'color' is text color  
// 'background-color' is bg color  
// 'opacity' is transparency  
// 'background-image' is texture name  
// 'left' is x position  
// 'top' is y position  
// 'name' gets element's name  
// 'width' gets width  
// 'height' gets height
```


// 'font-size' gets font size
// 'z-index' gets the layer index
// can get any css property value using its css name

// example
// get('thePlayer', 'name');

// example
// get('thePlayer', 'opacity');

// example
// get('thePlayer', 'z-index');

// allows you to get any css property value by using its css name

//--//

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

```
// makeBackground.js
```

```
function makeBackground()
{
    let bgDiv = ce('div');
    bgDiv.id = 'bgDiv';
    bgDiv.style.position = 'absolute';
    bgDiv.style.left = '0px';
    bgDiv.style.top = '0px';
    bgDiv.style.width = backgroundSizeX + 'px';
    bgDiv.style.height = backgroundSizeY + 'px';
    bgDiv.style.backgroundColor = 'rgb(0, 0, 0)';

    if (online == true)
    {
        bgDiv.style.background =
'url("https://collegeofscripting.weebly.com/uplo
ads/6/4/4/8/64482293/stars-nice-
edited_orig.png");'
    }
    else if (online == false)
    {
        bgDiv.style.background =
'url("src/media/textures/stars-nice-
edited_orig.png");'
    }
}
```

```
    bgDiv.style.zIndex = 0;  
    ba(bgDiv);  
}
```

```
//--//
```

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

```
// makeBottomToolbar.js
```

```
function activateBuildMode()  
{  
    setTimeout(function()  
    {  
        window.addEventListener("click",  
structuresCreate, false);  
    }, 500);  
}
```

```
let bottomToolbarArray =  
[
```

```

{
    name: "Build",
    functionName: `activateBuildMode();`
},
{
    name: "Player",
    functionName: `menuPlayer()`,
},
{
    name: "GoTo",
    functionName:
`structuresRow(ourPlayer.x, ourPlayer.y, 125,
200, structuresArray);`,
},
{
    name: "MouseOn",
    functionName: `mouseOn()`,
},
{
    name: "MouseOff",
    functionName: `mouseOff()`,
}
];

```

```

function makeBottomToolbar(whichArray)
{

```

```
let theDetails = ce('details');  
theDetails.setAttribute('open', '');  
theDetails.style.position = 'fixed';  
theDetails.style.left = '2px';  
theDetails.style.bottom = '0px';  
theDetails.style.zIndex = '9000000000000';  
theDetails.style.color = 'rgb(100, 100, 100)';  
ba(theDetails);
```

```
//-//
```

```
let theSummary = ce('summary');  
theSummary.innerHTML = "";  
theDetails.append(theSummary);
```

```
//-//
```

```
let toolBarMainDiv = ce("div");  
toolBarMainDiv.id = "toolBarMainDiv";  
toolBarMainDiv.title =  
"makeBottomToolbar()";  
theDetails.append(toolBarMainDiv);
```

```
//-//
```

```
let toolBarSubDiv = ce("div");
```

```
toolBarSubDiv.id = "toolBarSubDiv";  
toolBarSubDiv.style.display = ' flex';  
toolBarSubDiv.style.justifyContent = ' space-around';  
toolBarSubDiv.style.color = 'white';  
toolBarMainDiv.append(toolBarSubDiv);
```

```
//-//
```

```
for (let x = 0; x < whichArray.length; x++) {  
    let theButton = ce("button");  
    theButton.id = whichArray[x].name;  
    theButton.className = "ourButton";  
    theButton.style.backgroundColor = "rgb(0,  
0, 0)";  
    theButton.style.color = "rgb(100, 100,  
100)";  
    theButton.innerHTML =  
whichArray[x].name;  
  
    theButton.onclick = function()  
    {  
        eval(' ' + whichArray[x].functionName + '  
');  
    };  
};
```

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

//--//

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

// makeDedication.js

function makeDedication()
{
  let dedication = ce(`div`);
  dedication.id = `dedication`;
  dedication.className = `glowBlue`;
  dedication.style.transition = `opacity 5.5s
ease`;
  dedication.style.opacity = 0;
  dedication.style.whiteSpace = 'nowrap';
  dedication.innerHTML = `Dedicated to God
the Father`;
```

```
setTimeout(function()
{
    dedication.style.opacity = 1.0;
}, 1000);

ba(dedication);
}

//--//

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

// makeEngineTitle.js

function makeEngineTitle()
{
    let gameEngineTitle = ce("div");
    gameEngineTitle.id = "gameEngineTitle";
    gameEngineTitle.className = "glowRed";
```



```
gameEngineTitle.innerHTML = "Topalian  
Game Engine Data Navigator";  
gameEngineTitle.style.transition = `opacity  
5.5s ease`;  
gameEngineTitle.style.opacity = 0;  
gameEngineTitle.style.whiteSpace =  
'nowrap';  
setTimeout(function()  
{  
    gameEngineTitle.style.opacity = 1.0;  
}, 4000);  
ba(gameEngineTitle);  
}  
  
/--/  
  
// Dedicated to God the Father  
// All Rights Reserved Christopher Andrew  
// Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
  
// makeInfoDiv.js
```

```
function makeInfoDiv()
```

```
{  
  let infoDiv = ce('div');  
  infoDiv.id = 'infoDiv';  
  infoDiv.className = 'divStyle001';  
  infoDiv.title = 'makeInfoDiv()';  
  infoDiv.innerHTML = 'Info';  
  infoDiv.style.position = 'fixed';  
  infoDiv.style.right = '0px';  
  infoDiv.style.bottom = '10px';  
  infoDiv.style.zIndex = 1000;  
  infoDiv.style.fontSize = '15px';  
  infoDiv.style.textShadow = '1px 1px rgb(0, 0,  
0)';  
  infoDiv.style.transition = `transform 0.5s  
ease`;  
  ba(infoDiv);  
}
```

//--//

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

// makeInfoDivPlayer.js

```
function makeInfoDivPlayer()
{
    let infoDivPlayer = ce('div');
    infoDivPlayer.id = 'infoDivPlayer';
    infoDivPlayer.className = 'divStyle001';
    infoDivPlayer.title = 'makeInfoDivPlayer()';
    infoDivPlayer.style.position = 'fixed';
    infoDivPlayer.style.right = '0px';
    infoDivPlayer.style.bottom = '60px';
    infoDivPlayer.style.zIndex = 1000;
    infoDivPlayer.style.fontSize = '15px';
    infoDivPlayer.style.textShadow = '1px 1px
rgb(0, 0, 0)';
    infoDivPlayer.style.transition = `transform
0.5s ease`;
    infoDivPlayer.innerHTML = 'infoDivPlayer';
    ba(infoDivPlayer);
}
```

//--//

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// makeInstructions.js
```

```
function makeInstructions()  
{  
    if (ge('instructions'))  
    {  
        removeElement('instructions');  
    }  
  
    audioPlay("sfx_blip_001", 1.0);  
  
    //-//  
  
    let mainDiv = ce("div");  
    mainDiv.id = "instructions";  
    mainDiv.title = "makeInstructions()";  
    mainDiv.style.position = "absolute";  
    mainDiv.style.left = ourPlayer.x + 100 + 'px';  
    mainDiv.style.top = ourPlayer.y - 20 + 'px';  
    mainDiv.style.width = 400 + 'px';
```

```
mainDiv.style.height = 200 + 'px';  
mainDiv.style.display = "flex";  
mainDiv.style.flexDirection = "column";  
ba(mainDiv);
```

```
//-//
```

```
let subDiv = ce("div");  
mainDiv.append(subDiv);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.className = "ourButton";  
closeButton.textContent = "X";  
closeButton.onclick = function()  
{  
    removeElement("instructions");  
};  
subDiv.append(closeButton);
```

```
//-//
```

```
subDiv.append(ce('hr'));
```

```
//-//
```

```
let dedication = ce("div");  
dedication.textContent = "Dedicated to God  
the Father";  
subDiv.append(dedication);
```

```
//-//
```

```
let hrUnderDedication = ce('hr');  
hrUnderDedication.style.marginBottom =  
'10px';  
subDiv.append(hrUnderDedication);
```

```
//-//
```

```
let instruction1 = ce("div");  
instruction1.textContent = "W is Up, S is  
Down, A is Left, D is Right";  
subDiv.append(instruction1);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction2 = ce("div");  
instruction2.textContent = "Numbers 1 to 9  
Changes Speed";  
subDiv.append(instruction2);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction3 = ce("div");  
instruction3.textContent = "0 - + BackSpace  
Home End for FASTER SPEEDS";  
subDiv.append(instruction3);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction4 = ce("div");  
instruction4.textContent = "Z for Player  
Menu or Double Click on Player for Player  
Menu";
```

```
subDiv.append(instruction4);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction5 = ce("div");  
instruction5.textContent = "X to start Mouse  
Position Mode";  
subDiv.append(instruction5);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction6 = ce("div");  
instruction6.textContent = "Left Click to find  
Mouse Position";  
subDiv.append(instruction6);
```

```
//-//
```



```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction7 = ce("div");  
instruction7.textContent = "V to end Mouse  
Position mode";  
subDiv.append(instruction7);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction8 = ce("div");  
instruction8.textContent = "B for Structure  
Mode";  
subDiv.append(instruction8);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction9 = ce("div");  
instruction9.textContent = "Left Click to  
Place a Structure";  
subDiv.append(instruction9);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction10 = ce("div");  
instruction10.textContent = "ESC to Exit  
Structure Mode";  
subDiv.append(instruction10);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction11 = ce("div");  
instruction11.textContent = "Left Click any  
Structure to EDIT that Structure";  
subDiv.append(instruction11);
```

//-//

subDiv.append(ce("br"));

//-//

**let instruction12 = ce("div");
instruction12.textContent = "Delete to
Remove Structures";
subDiv.append(instruction12);**

//-//

subDiv.append(ce("br"));

//-//

**let instruction13 = ce("div");
instruction13.textContent = "J to Create a
Structures GoTo Menu";
subDiv.append(instruction13);**

//-//

subDiv.append(ce("br"));

//-//

```
let instruction14 = ce("div");  
instruction14.textContent = "Load Button to  
Load Structures Data";  
subDiv.append(instruction14);
```

//-//

```
subDiv.append(ce("br"));
```

//-//

```
let instruction15 = ce("div");  
instruction15.textContent = "P for Color  
Player Menu";  
subDiv.append(instruction15);
```

//-//

```
subDiv.append(ce("br"));
```

//-//

```
let instruction16 = ce("div");
```

```
instruction16.textContent = "T for Texture  
Player Menu";  
subDiv.append(instruction16);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction17 = ce("div");  
instruction17.textContent = "O for Player  
Opacity Menu";  
subDiv.append(instruction17);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let instruction18 = ce("div");  
instruction18.textContent = "N for  
Transparent, M for Solid";  
subDiv.append(instruction18);
```

//-//

subDiv.append(ce("br"));

//-//

**let instruction19 = ce("div");
instruction19.textContent = "G for Clearing
Player Text";
subDiv.append(instruction19);**

//-//

subDiv.append(ce("br"));

//-//

**let instruction20 = ce("div");
instruction20.textContent = "Control + F to
Search, then Left Click to Travel to Search
Result";
subDiv.append(instruction20);**

//-//

subDiv.append(ce("br"));

```
//-//
```

```
let linkToCode = ce("a");
linkToCode.href =
"https://github.com/ChristopherTopalian/Topali
anGameEngineDataNavigator";
linkToCode.target = "_blank";
linkToCode.textContent = "Christopher
Topalian Data Navigator Code Here";
subDiv.append(linkToCode);
}
```

```
//--//
```

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

```
// makeLine.js
```

```
// Make a Message Box
// with a message of the Section name
```

```
function makeLine(theYPos, sector,
ourMessage)
{
    let makeLine = ce("div");
    makeLine.id = "sector" + sector;
    makeLine.className = "divStyle001";
    makeLine.style.position = "absolute";
    makeLine.style.left = 0 +'px';
    makeLine.style.top = theYPos + 'px';
    makeLine.style.width = 1000 +'px';
    makeLine.style.height = 25 +'px';
    makeLine.innerHTML = ourMessage;
    ba(makeLine);
}

//--//

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

// makeLogo.js
```



```
function makeLogo()
{
    let mainDiv = ce("div");
    mainDiv.id = "mainDiv";
    mainDiv.className = "divStyle001";
    mainDiv.title = "makeLogo()";
    mainDiv.style.position = "absolute";
    mainDiv.style.left = "900px";
    mainDiv.style.top = "20px";
    mainDiv.style.width = "420px";
    mainDiv.style.height = "155px";
    mainDiv.style.borderRadius = "10px";
    ba(mainDiv);

    //-//

    let dedication = ce("div");
    dedication.style.fontFamily = "Arial";
    dedication.style.fontSize = "16pt";
    dedication.textContent = "Dedicated to God
the Father";
    mainDiv.append(dedication);

    //-//

    mainDiv.append(ce('hr'));
```

```
//-//
```

```
let appTitle = ce("div");  
appTitle.style.color = "rgb(126, 203, 255)";  
appTitle.innerHTML = "TOPALIAN GAME  
ENGINE<br>DATA NAVIGATOR";  
mainDiv.append(appTitle);
```

```
//-//
```

```
let brUnderAppTitle = ce('br');  
brUnderAppTitle.style.lineHeight = '7px';  
mainDiv.append(brUnderAppTitle);
```

```
//-//
```

```
let credit = ce("div");  
credit.style.fontFamily = "arial";  
credit.style.fontStyle = "normal";  
credit.style.lineHeight = "20px";  
credit.textContent = "by Christopher  
Topalian";  
mainDiv.append(credit);
```

```
//-//
```

```
let brUnderCredit = ce('br');  
brUnderCredit.style.lineHeight = '7px';  
mainDiv.append(brUnderCredit);
```

```
//-//
```

```
let appLink = ce("a");  
appLink.href =  
"https://github.com/ChristopherTopalian/Topali  
anGameEngineDataNavigator";  
appLink.target = "_blank";  
appLink.textContent =  
"GitHub.com/ChristopherTopalian";  
mainDiv.append(appLink);  
}
```

```
//--//
```

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

// makePlayer.js

```
function makePlayer()
{
    let player = ce("div");
    player.id = "thePlayer";
    player.style.position = 'absolute';
    player.style.left = ourPlayer.x;
    player.style.top = ourPlayer.y;
    player.style.backgroundSize = 'cover';
    player.style.backgroundPosition = 'center
center';
    player.style.backgroundRepeat = 'no-repeat';
    player.style.objectFit = 'cover';
    player.style.zIndex = '2000';
    player.ondblclick = function()
    {
        menuPlayer();
    };
    player.onclick = function()
    {
        keyboardControlOn();
    };
    ba(player);
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// makePlayerTextArea.js
```

```
function makePlayerTextArea()
```

```
{
```

```
    let mainDiv = ce("div");
```

```
    mainDiv.id = "mainDivPlayerTextArea";
```

```
    mainDiv.title = 'makePlayerTextArea()';
```

```
    mainDiv.style.position = "absolute";
```

```
    mainDiv.style.left = "400px";
```

```
    mainDiv.style.top = "200px";
```

```
    mainDiv.style.display = "flex";
```

```
    mainDiv.style.flexDirection = "column";
```

```
    ba(mainDiv);
```

```
//--//
```

```
let theTextArea = ce("textarea");
```

```
theTextArea.id = "textAreaPlayerData";
theTextArea.className = "divStyle001";
theTextArea.style.width = "325px";
theTextArea.style.height = "300px";
theTextArea.style.zIndex = "1";
theTextArea.style.fontSize = '17px';
theTextArea.style.fontWeight = 'normal';
theTextArea.style.color = 'rgb(150, 150, 150)';
theTextArea.style.textAlign = "left";
theTextArea.onclick = function()
{
    showPlayerData();
};
theTextArea.onkeydown = function()
{
    keyboardControlOff();
};
theTextArea.onblur = function()
{
    keyboardControlOn();
};
mainDiv.append(theTextArea);

//-
```

```
let controlsDiv = ce("div");
```

```
controlsDiv.style.display = "flex";  
controlsDiv.style.flexDirection = "row";  
mainDiv.append(controlsDiv);
```

```
//-//
```

```
let copyButton = ce("button");  
copyButton.className = "ourButton";  
copyButton.textContent = "Copy";  
copyButton.onclick = function()  
{  
    copyIt(theTextArea.id);  
};  
controlsDiv.append(copyButton);
```

```
//-//
```

```
let downloadButton = ce("button");  
downloadButton.className = "ourButton";  
downloadButton.textContent = "Download";  
downloadButton.onclick = function()  
{  
    downloadPlayerData();  
};  
controlsDiv.append(downloadButton);  
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// makeStructureTextArea.js
```

```
// Get All Structure Attributes
```

```
function makeStructureTextArea()
```

```
{
```

```
    let mainDiv = ce("div");
```

```
    mainDiv.id = "structureTextAreald";
```

```
    mainDiv.title = 'makeStructureTextArea()';
```

```
    mainDiv.style.position = "absolute";
```

```
    mainDiv.style.left = "20px";
```

```
    mainDiv.style.top = "200px";
```

```
    mainDiv.style.display = "flex";
```

```
    mainDiv.style.flexDirection = "column";
```

```
    ba(mainDiv);
```

```
//--//
```



```
let theTextArea = ce("textarea");
theTextArea.id = "textAreaStructureData";
theTextArea.className = "divStyle001";
theTextArea.style.width = "325px";
theTextArea.style.height = "300px";
theTextArea.style.fontSize = '17px';
theTextArea.style.fontWeight = 'normal';
theTextArea.style.color = 'rgb(150, 150, 150)';
theTextArea.style.textAlign = "left";
theTextArea.onclick = function()
{
    showStructureData();
};
theTextArea.onkeydown = function()
{
    keyboardControlOff();
};
theTextArea.onblur = function()
{
    keyboardControlOn();
};
mainDiv.append(theTextArea);

//-
```

```
let controlsDiv = ce("div");
controlsDiv.style.display = "flex";
controlsDiv.style.flexDirection = "row";
mainDiv.append(controlsDiv);
```

```
//-//
```

```
let copyButton = ce("button");
copyButton.className = "ourButton";
copyButton.style.color = "rgb(255, 255,
255)";
copyButton.textContent = "Copy";
copyButton.onclick = function()
{
    copyIt(theTextArea.id);
};
controlsDiv.append(copyButton);
```

```
//-//
```

```
let downloadButton = ce("button");
downloadButton.className = "ourButton";
downloadButton.style.color = "rgb(255, 255,
255)";
downloadButton.style.zIndex = "10";
downloadButton.textContent = "Download";
```

```
downloadButton.onclick = function()
{
    downloadStructureData();
};
controlsDiv.append(downloadButton);
}

//--//

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

// makeVideo.js

function makeVideo(whichVideold)
{
    let videold = whichVideold;

    let youTubeVideo = ce("iframe");
    youTubeVideo.style.position = "absolute";
    youTubeVideo.style.left =
    positionGet(activePlayer).x;
```

```
    youTubeVideo.style.top =  
positionGet(activePlayer).y;  
    youTubeVideo.width = '560';  
    youTubeVideo.height = '315';  
    youTubeVideo.src =  
`https://www.youtube.com/embed/${videoid}`;  
    youTubeVideo.frameBorder = '0';  
    youTubeVideo.allowFullscreen = true;  
    ba(youTubeVideo);  
}
```

//--//

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

// menuPlayer.js

// player menu activated when player presses
the letter z

function menuPlayer()

```
{  
  if (ge('menuPlayer'))  
  {  
    removeElement('menuPlayer');  
  }  
  
  //-//  
  
  let theMenu = ce("div");  
  theMenu.id = "menuPlayer";  
  theMenu.className = "divStyle001";  
  theMenu.title = "menuPlayer()";  
  theMenu.style.position = "absolute";  
  theMenu.style.left = ourPlayer.x - 50 +'px';  
  theMenu.style.top = ourPlayer.y + 'px';  
  theMenu.style.width = 250 +'px';  
  theMenu.style.height = 200 +'px';  
  theMenu.style.overflowY = 'scroll';  
  ba(theMenu);  
  
  // prevent menu leaving screen on left side  
  if (positionGet("menuPlayer").x <= 100)  
  {  
    theMenu.style.left = 50;  
  }  
}
```

```
// prevent menu being positioned off screen  
top side
```

```
if (positionGet("menuPlayer").y <= 200)  
{  
    theMenu.style.top = 50;  
}
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.textContent = "X";  
closeButton.className = "ourButton";  
closeButton.onclick = function()  
{  
    removeElement("menuPlayer");  
};  
closeButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let instructionsText = ce("span");  
instructionsText.id = 'instructionsText';  
instructionsText.className =  
"textCompact";  
instructionsText.textContent =  
"Instructions";  
theMenu.append(instructionsText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let instructionsButton = ce("button");  
instructionsButton.id = 'instructionsButton';  
instructionsButton.className =  
"ourButton";  
instructionsButton.textContent =  
'Instructions';  
instructionsButton.onmouseover =  
function()  
{  
    audioPlay('sfx_warp_001', 1.0);
```

```
};  
instructionsButton.onclick = function()  
{  
    makeInstructions();  
};  
theMenu.append(instructionsButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let goToText = ce("span");  
goToText.textContent = "Go To";  
goToText.className = "textCompact";  
theMenu.append(goToText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let sections =  
[
```



```
{  
  name: 'Scripting',  
  class: 'ourButton',  
  whenClicked: `travelTo('scripting');`  
},  
  
{  
  name: 'Evidence',  
  class: 'ourButton',  
  whenClicked: `travelTo('evidence');`  
},  
  
{  
  name: 'Books',  
  class: 'ourButton',  
  whenClicked: `travelTo('books');`  
},  
  
{  
  name: 'News',  
  class: 'ourButton',  
  whenClicked: `travelTo('news');`  
}  
];
```

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

```
{  
    let travelToButton = ce("button");  
    travelToButton.textContent =  
sections[i].name;  
    travelToButton.className =  
sections[i].class;  
    travelToButton.onmouseover = function()  
    {  
        audioPlay('sfx_warp_001', 1.0);  
    };  
    travelToButton.onclick = function()  
    {  
        audioPlay('sfx_blip_001', 1.0);  
        eval(sections[i].whenClicked);  
    };  
    theMenu.append(travelToButton);  
}
```

//-//

```
theMenu.append(ce('hr'));
```

//-//

```
let audioText = ce("span");  
audioText.textContent = "Audio";
```

```
audioText.className = "textCompact";  
theMenu.append(audioText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let songButton = ce("button");  
songButton.textContent = "Songs";  
songButton.className = "ourButton";  
songButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
songButton.onclick = function()  
{  
    songChoicesMenu();  
};  
theMenu.append(songButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let createStructuresText = ce("span");  
createStructuresText.textContent = "Create  
Structures";  
createStructuresText.className =  
"textCompact";  
theMenu.append(createStructuresText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let howToButton = ce("button");  
howToButton.textContent = "How To";  
howToButton.className = 'ourButton';  
howToButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
howToButton.onclick = function()  
{  
    structuresMenu();  
};
```

```
theMenu.append(howToButton);
```

```
//-//
```

```
let createButton = ce("button");  
createButton.textContent = "Create";  
createButton.className = "ourButton";  
createButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
createButton.onclick = function()  
{  
    activateBuildMode();  
};  
theMenu.append(createButton);
```

```
//-//
```

```
let goToButton = ce("button");  
goToButton.textContent = "Go To";  
goToButton.className = "ourButton";  
goToButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};
```

```
goToButton.onclick = function()
{
    structuresRow(ourPlayer.x, ourPlayer.y,
125, 200, structuresArray);
};
theMenu.append(goToButton);
```

//-//

```
let loadButton = ce("button");
loadButton.textContent = "Load";
loadButton.id = "loadStructureButtonId";
loadButton.className = "ourButton";
loadButton.onmouseover = function()
{
    audioPlay('sfx_warp_001', 1.0);
};
loadButton.onclick = function()
{
    loadStructureScript('structuresArray.js');
};
theMenu.append(loadButton);
```

//-//

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let playerText = ce("span");  
playerText.textContent = "Player";  
playerText.className = "textCompact";  
theMenu.append(playerText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let nameButton = ce("button");  
nameButton.textContent = "Name";  
nameButton.className = "ourButton";  
nameButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
nameButton.onclick = function()  
{  
    menuPlayerName('thePlayer');  
};  
theMenu.append(nameButton);
```

```
//-//
```

```
let loadScriptButton = ce("button");  
loadScriptButton.textContent = "Load";  
loadScriptButton.className = "ourButton";  
loadScriptButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
loadScriptButton.onclick = function()  
{  
    loadPlayerScript('playerData.js');  
};
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let playerAppearanceText = ce("span");  
playerAppearanceText.textContent = "Player  
Appearance";  
playerAppearanceText.className =  
"textCompact";
```



```
theMenu.append(playerAppearanceText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let textColorButton = ce("button");  
textColorButton.textContent = "Text Color";  
textColorButton.className = "ourButton";  
textColorButton.title =  
'menuPlayerColorText';  
textColorButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
textColorButton.onclick = function()  
{  
    menuPlayerColorText();  
};  
theMenu.append(textColorButton);
```

```
//-//
```

```
let textureButton = ce("button");
```

```
textureButton.textContent = "Texture";
textureButton.className = "ourButton";
textureButton.onmouseover = function()
{
    audioPlay('sfx_warp_001', 1.0);
};
textureButton.onclick = function()
{
    menuPlayerTexture();
};
theMenu.append(textureButton);

//-//
```

```
let opacityButton = ce("button");
opacityButton.textContent = "Opacity";
opacityButton.className = "ourButton";
opacityButton.onmouseover = function()
{
    audioPlay('sfx_warp_001', 1.0);
};
opacityButton.onclick = function()
{
    menuPlayerOpacity();
};
theMenu.append(opacityButton);
```

//-//

```
let colorButton = ce("button");
colorButton.textContent = "Color";
colorButton.className = "ourButton";
colorButton.onmouseover = function()
{
    audioPlay('sfx_warp_001', 1.0);
};
colorButton.onclick = function()
{
    menuPlayerColor();
};
theMenu.append(colorButton);
```

//-//

```
theMenu.append(ce('hr'));
```

//-//

```
let playerSpeedText = ce("span");
playerSpeedText.textContent = "Player
Speed";
```

```
playerSpeedText.className =  
"textCompact";  
theMenu.append(playerSpeedText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let playerSpeedButton = ce("button");  
playerSpeedButton.textContent = "Speed";  
playerSpeedButton.className =  
"ourButton";  
playerSpeedButton.onmouseover =  
function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
playerSpeedButton.onclick = function()  
{  
    menuPlayerSpeed();  
};  
theMenu.append(playerSpeedButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let playerPosText = ce("span");  
playerPosText.textContent = "Player  
Position";  
playerPosText.className = "textCompact";  
theMenu.append(playerPosText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let getPosButton = ce("button");  
getPosButton.textContent = "Get";  
getPosButton.className = "ourButton";  
getPosButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
getPosButton.onclick = function()  
{
```

```
    menuPlayerPositionGet();  
};  
theMenu.append(getPosButton);  
  
//-//
```

```
let setPosButton = ce("button");  
setPosButton.textContent = "Set";  
setPosButton.className = "ourButton";  
setPosButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
setPosButton.onclick = function()  
{  
    menuPlayerPositionSet();  
};  
theMenu.append(setPosButton);  
  
//-//
```

```
theMenu.append(ce('hr'));  
  
//-//
```

```
let playerSizeText = ce("span");
```

```
playerSizeText.textContent = "Player Size";  
playerSizeText.className = "textCompact";  
theMenu.append(playerSizeText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let playerSizeGetButton = ce("button");  
playerSizeGetButton.textContent = "Get";  
playerSizeGetButton.className =  
"ourButton";  
playerSizeGetButton.onmouseover =  
function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
playerSizeGetButton.onclick = function()  
{  
    menuPlayerSizeGet();  
};  
theMenu.append(playerSizeGetButton);
```

```
//-//
```

```
let playerSizeSetButton = ce("button");
playerSizeSetButton.textContent = "Set";
playerSizeSetButton.className =
"ourButton";
playerSizeSetButton.onmouseover =
function()
{
    audioPlay('sfx_warp_001', 1.0);
};
playerSizeSetButton.onclick = function()
{
    menuPlayerSizeSet();
};
theMenu.append(playerSizeSetButton);

//-
```

```
theMenu.append(ce('hr'));
```

```
//-
```

```
let mousePosText = ce("span");
mousePosText.textContent = "Mouse
Position";
mousePosText.className = "textCompact";
```



```
theMenu.append(mousePosText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let mouseOnButton = ce("button");  
mouseOnButton.textContent = "On";  
mouseOnButton.className = "ourButton";  
mouseOnButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
mouseOnButton.onclick = function()  
{  
    mouseOn();  
    audioPlay("blipQuiet", 1.0);  
};  
theMenu.append(mouseOnButton);
```

```
//-//
```

```
let mouseOffButton = ce("button");  
mouseOffButton.textContent = "Off";
```

```
mouseOffButton.className = "ourButton";  
mouseOffButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
mouseOffButton.onclick = function()  
{  
    mouseOff();  
    audioPlay("sfx_blip_001", 1.0);  
};  
theMenu.append(mouseOffButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let distanceText = ce("span");  
distanceText.textContent = "Distance";  
distanceText.className = "textCompact";  
theMenu.append(distanceText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let distanceFinderButton = ce("button");  
distanceFinderButton.textContent =  
"Distance Finder";  
distanceFinderButton.className =  
"ourButton";  
distanceFinderButton.onmouseover =  
function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
distanceFinderButton.onclick = function()  
{  
    menuPlayerDistance();  
};  
theMenu.append(distanceFinderButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let findText = ce("span");
```

```
findText.textContent = "Find";  
findText.className = "textCompact";  
theMenu.append(findText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let searchButton = ce("button");  
searchButton.textContent = "Search";  
searchButton.className = "ourButton";  
searchButton.onmouseover = function()  
{  
    audioPlay('sfx_warp_001', 1.0);  
};  
searchButton.onclick = function()  
{  
    menuSearch();  
};  
theMenu.append(searchButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// menuPlayerColor.js
```

```
function menuPlayerColor()
```

```
{
```

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

```
//--//
```

```
if (ge('menuColorPlayer'))
```

```
{
```

```
    removeElement('menuColorPlayer');
```

```
}
```

```
//--//
```

```
let theMenu = ce("div");
theMenu.id = "menuColorPlayer";
theMenu.className = "divStyle001";
theMenu.style.position = "absolute";
theMenu.style.left = ourPlayer.x + 150 + 'px';
theMenu.style.top = ourPlayer.y + 'px';
theMenu.style.width = 200 + 'px';
theMenu.style.height = 250 + 'px';
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce('button');
closeButton.textContent = 'X';
closeButton.className = 'ourButton';
closeButton.onclick = function()
{
    removeElement('menuColorPlayer');
    keyboardControlOn();
};
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let colorPlayerText = ce("span");  
colorPlayerText.textContent = "Color  
Player";  
theMenu.append(colorPlayerText);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let redTextArea = ce("textarea");  
redTextArea.id = "theRed";  
redTextArea.className = "textInput";  
redTextArea.onkeydown = function()  
{  
    keyboardControlOff();  
};
```

```
//-//
```

```
// colorRText
```

```
let colorRText = ce("span");  
colorRText.textContent = "R";
```

```
colorRText.style.marginRight = "5px";  
theMenu.append(colorRText);  
theMenu.append(redTextArea);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
// colorGText
```

```
let colorGText = ce("span");  
colorGText.textContent = "G";  
colorGText.style.marginRight = "5px";  
theMenu.append(colorGText);
```

```
//-//
```

```
let greenTextArea = ce("textarea");  
greenTextArea.id = "theGreen";  
greenTextArea.className = "textInput";  
greenTextArea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(greenTextArea);
```



```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
// colorBText
```

```
let colorBText = ce("span");
```

```
colorBText.textContent = "B";
```

```
colorBText.style.marginRight = "5px";
```

```
theMenu.append(colorBText);
```

```
//-//
```

```
let blueTextArea = ce("textarea");
```

```
blueTextArea.id = "theBlue";
```

```
blueTextArea.className = "textInput";
```

```
blueTextArea.onkeydown = function()
```

```
{
```

```
    keyboardControlOff();
```

```
};
```

```
theMenu.append(blueTextArea);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let enterButton = ce("button");
enterButton.textContent = "Enter";
enterButton.className = "ourButton";
enterButton.onclick = function()
{
    // remove texture
    ge(activePlayer).style.backgroundImage =
'none';

    // color player
    ge(activePlayer).style.backgroundColor =
`rgb(
    ${ ge("theRed").value },
    ${ ge("theGreen").value },
    ${ ge("theBlue").value }
)`;

    keyboardControlOn();
    enterButtonSound();
};
theMenu.append(enterButton);
```

//-//

```
let buttonContainer = ce('div');  
buttonContainer.style.display = 'grid';  
buttonContainer.style.gridTemplateColumns  
= 'repeat(3, 1fr);'  
buttonContainer.style.gap = '0.5px';  
theMenu.append(buttonContainer);
```

//-//

```
for (let z = 0; z < colorChoices.length; z++)  
{  
    let colorChoiceButton = ce("button");  
    colorChoiceButton.textContent =  
colorChoices[z].name;  
    colorChoiceButton.className =  
'ourButton';  
    colorChoiceButton.onclick = function()  
    {  
        ge(activePlayer).style.backgroundColor  
= "rgb("+colorChoices[z].color+"");  
  
        // split color value to array of rgb  
        let rgbValues =  
colorChoices[z].color.split(',');  

```

```
        ge("theRed").value =  
rgbValues[0].trim();  
        ge("theGreen").value =  
rgbValues[1].trim();  
        ge("theBlue").value =  
rgbValues[2].trim();  
  
        updatePlayer();  
    };  
    theMenu.append(colorChoiceButton);  
}
```

//-//

```
buttonContainer.append(ce("br"));
```

//-//

```
let customColorButton = ce("button");  
customColorButton.textContent = "Custom";  
customColorButton.className =  
"ourButton";  
customColorButton.onclick = function()  
{  
    customColor();  
}
```

```
};  
theMenu.append(customColorButton);  
  
//-//  
  
    showPlayerData();  
}  
  
function customColor()  
{  
    audioPlay("sfx_blip_001", 1.0);  
  
    // color is only applied if NO texture is on the  
    player  
    ge(activePlayer).style.backgroundImage =  
    'none';  
  
    let whichColor = prompt("Enter RGB as  
    255,255,255")  
  
    let theColorFormatted =  
    'rgb('+whichColor+')';  
  
    // color of player  
    ge(activePlayer).style.backgroundColor =  
    theColorFormatted;
```

```
let colorSeparated = whichColor.split(',');

// fill in textarea, show value of choice
ge("theRed").value = colorSeparated[0];
ge("theGreen").value = colorSeparated[1];
ge("theBlue").value = colorSeparated[2];

// update playerArray to show changed value
updatePlayer();
}

//--//

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

// menuPlayerColorText.js

function menuPlayerColorText()
{
    audioPlay("sfx_blip_001", 1.0);
```

```
//-//
```

```
if (ge("menuColorPlayerText"))  
{  
    removeElement("menuColorPlayerText");  
}
```

```
//-//
```

```
let theMenu = ce("div");  
theMenu.id = "menuColorPlayerText";  
theMenu.className = "divStyle001";  
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 100 + 'px';  
theMenu.style.top = ourPlayer.y - 75 + 'px';  
theMenu.style.width = '400px';  
theMenu.style.height = '255px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";
```

```
closeButton.onclick = function()
{
    removeElement("menuColorPlayerText");
    keyboardControlOn();
};
theMenu.append(closeButton);
```

```
//-//
```

```
let containerDiv = ce("div");
theMenu.append(containerDiv);
```

```
//-//
```

```
containerDiv.append(ce('hr'));
```

```
//-//
```

```
let titleSpan = ce("span");
titleSpan.style.color = "rgb(255, 255, 255)";
titleSpan.textContent = "COLOR TEXT";
containerDiv.append(titleSpan);
containerDiv.append(ce("br"));
```

```
//-//
```



```
let redLabel = ce('div');  
redLabel.textContent = "Red ";  
containerDiv.append(redLabel);
```

```
//-//
```

```
let redTextArea = ce("textarea");  
redTextArea.id = "theRedFont";  
redTextArea.className = 'textInput';  
redTextArea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
containerDiv.append(redTextArea);
```

```
//-//
```

```
containerDiv.append(ce("br"));
```

```
//-//
```

```
let greenLabel = ce('div');  
greenLabel.textContent = "Green ";  
containerDiv.append(greenLabel);
```

```
//-//
```

```
let greenTextArea = ce("textarea");
greenTextArea.id = "theGreenFont";
greenTextArea.className = 'textInput';
greenTextArea.onkeydown = function()
{
    keyboardControlOff();
};
containerDiv.append(greenTextArea);
```

```
//-//
```

```
containerDiv.append(ce("br"));
```

```
//-//
```

```
let blueLabel = ce('div');
blueLabel.textContent = "Blue ";
containerDiv.append(blueLabel);
```

```
//-//
```

```
let blueTextArea = ce("textarea");
blueTextArea.id = "theBlueFont";
blueTextArea.className = 'textInput';
blueTextArea.onkeydown = function()
```

```
{  
    keyboardControlOff();  
};  
containerDiv.append(blueTextArea);
```

```
//-//
```

```
containerDiv.append(ce("br"));
```

```
//-//
```

```
let enterButton = ce("button");  
enterButton.className = 'ourButton';  
enterButton.textContent = 'Enter';  
enterButton.onclick = function()  
{  
    customPlayerTextColorFromTextInput();  
    keyboardControlOn();  
    enterButtonSound();  
};  
containerDiv.append(enterButton);
```

```
//-//
```

```
containerDiv.append(ce("hr"));
```

//-//

```
let buttonContainer = ce('div');  
buttonContainer.style.display = 'grid';  
buttonContainer.style.gridTemplateColumns  
= 'repeat(3, 1fr)';  
containerDiv.append(buttonContainer);
```

//-//

```
for (let z = 0; z < colorChoices.length; z++)  
{  
    let colorChoiceButton = ce("button");  
    colorChoiceButton.textContent =  
colorChoices[z].name;  
    colorChoiceButton.className =  
'ourButton';  
    colorChoiceButton.onclick = function()  
    {  
        ge(activePlayer).style.color =  
"rgb("+colorChoices[z].color+"");  
  
        // split color value to array of rgb  
        let rgbValues =  
colorChoices[z].color.split(',');  

```

```
        ge("theRedFont").value =  
rgbValues[0].trim();  
        ge("theGreenFont").value =  
rgbValues[1].trim();  
        ge("theBlueFont").value =  
rgbValues[2].trim();  
  
        updatePlayer();  
    };  
    theMenu.append(colorChoiceButton);  
}
```

//-//

```
buttonContainer.append(ce("br"));
```

//-//

```
containerDiv.append(ce("br"));
```

//-//

```
let customButton = ce("button");  
customButton.className = 'ourButton';  
customButton.textContent = 'Custom';  
customButton.onclick = function()
```

```
{  
    customFontColor();  
};  
containerDiv.append(customButton);  
}
```

```
function customFontColor()  
{  
    let theColor = prompt("Enter RGB as 255,  
255, 255")  
  
    let theColorFormatted = 'rgb('+theColor+')';  
  
    // color of text on player object  
    ge(activePlayer).style.color =  
theColorFormatted;  
  
    let colorSeparated = theColor.split(',');  
  
    // fill in the textarea, to show the value of our  
choice  
    ge("theRedFont").value = colorSeparated[0];  
    ge("theGreenFont").value =  
colorSeparated[1];  
    ge("theBlueFont").value = colorSeparated[2];  
}
```

```
// update playerArray to show changed value
updatePlayer();
}

// Read the 3 Textarea, for the red, green, blue
values of our player text color
// Set the text color of our player
function
customPlayerTextColorFromTextInput()
{
    let r = ge("theRedFont").value;
    let g = ge("theGreenFont").value;
    let b = ge("theBlueFont").value;

    ge(activePlayer).style.color=
    "rgb("+r+", "+g+", "+b+)";

    // fill textarea, show value of choice
    ge("theRedFont").value = r;
    ge("theGreenFont").value = g;
    ge("theBlueFont").value = b;

    // update playerArray to show changed value
    updatePlayer();
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// menuPlayerDistance.js
```

```
function menuPlayerDistance()
```

```
{
```

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

```
//--//
```

```
if (ge('menuPositionPlayer'))
```

```
{
```

```
    removeElement('menuPositionPlayer');
```

```
}
```

```
let theMenu = ce("div");
```

```
theMenu.id = "menuPositionPlayer";
```

```
theMenu.className = "divStyle001";
```

```
theMenu.style.position = "absolute";
```



```
theMenu.style.left = ourPlayer.x + 150 + 'px';  
theMenu.style.top = ourPlayer.y + 'px';  
theMenu.style.width = '400px';  
theMenu.style.height = '160px';  
ba(theMenu);
```

```
//-//
```

```
let removeButton = ce("button");  
removeButton.textContent = 'X';  
removeButton.className = 'ourButton';  
removeButton.style.marginBottom = '-4px';  
removeButton.onclick = function()  
{  
    removeElement("menuPositionPlayer");  
    keyboardControlOn();  
};  
theMenu.append(removeButton);
```

```
//-//
```

```
let hr001 = ce("hr");  
theMenu.append(hr001);
```

```
//-//
```

```
let distanceSection = ce("div");  
theMenu.append(distanceSection);
```

```
//-//
```

```
let distanceLabel = ce("span");  
distanceLabel.style.color = 'rgb(255, 255,  
255)';  
distanceLabel.textContent = 'Distance';  
distanceSection.append(distanceLabel);
```

```
//-//
```

```
let hr002 = ce("hr");  
distanceSection.append(hr002);
```

```
//-//
```

```
let idLabel = ce("span");  
idLabel.style.color = 'rgb(255, 255, 255)';  
idLabel.textContent = 'Id of Object';  
distanceSection.append(idLabel);
```

```
//-//
```

```
let idTextarea = ce("textarea");
```

```
idTextarea.id = 'idToFindDistance';  
idTextarea.className = 'textInput';  
idTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(idTextarea);  
  
//-//
```

```
let enterButton = ce("button");  
enterButton.textContent = 'Enter';  
enterButton.className = 'ourButton';  
enterButton.onclick = function()  
{  
    keepGettingXYCustom();  
    keyboardControlOn();  
    enterButtonSound();  
};  
theMenu.append(enterButton);  
  
//-//
```

```
let hr003 = ce("hr");  
theMenu.append(hr003);
```

//-//

```
let cancelButton = ce("button");  
cancelButton.textContent = 'Cancel';  
cancelButton.className = 'ourButton';  
cancelButton.onclick = function()  
{  
    removeTimer(ourTimer);  
};  
theMenu.append(cancelButton);  
}
```

//--//

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

// menuPlayerName.js

```
function menuPlayerName(whichId)  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
// if menu exists, remove it, and then create a  
new menu wherever the player is  
if (ge("menuNamePlayer"))  
{  
    removeElement("menuNamePlayer");  
}
```

```
//-//
```

```
let theMenu = ce("div");  
theMenu.id = "menuNamePlayer";  
theMenu.className = "divStyle001";  
theMenu.title =  
"menuPlayerName(whichId);";  
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 150 + 'px';  
theMenu.style.top = ourPlayer.y + 'px';  
theMenu.style.width = '250px';  
theMenu.style.height = '155px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.textContent = "X";
```

```
closeButton.className = "ourButton";
closeButton.style.marginBottom = "-4px";
closeButton.onclick = function()
{
    removeElement("menuNamePlayer");
    keyboardControlOn();
};
theMenu.append(closeButton);
```

```
//-//
```

```
let hr = ce("hr");
hr.style.height = "1px";
hr.style.border = "none";
hr.style.backgroundColor = "rgb(50, 50, 50)";
hr.style.color = "rgb(50, 50, 50)";
theMenu.append(hr);
```

```
//-//
```

```
let titleSpan = ce("span");
titleSpan.textContent = "NAME OF PLAYER";
titleSpan.style.color = "rgb(255, 255, 255)";
theMenu.append(titleSpan);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let nameLabel = ce('div');  
nameLabel.textContent = "Name ";  
theMenu.append(nameLabel);
```

```
//-//
```

```
let nameTextArea = ce("textarea");  
nameTextArea.id = "thePlayerName";  
nameTextArea.className = "textInput";  
nameTextArea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(nameTextArea);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let enterButton = ce("button");
enterButton.textContent = "Enter";
enterButton.className = "ourButton";
enterButton.onclick = function()
{
    customPlayerNameFromTextInput();
    keyboardControlOn();
    enterButtonSound();
};
theMenu.append(enterButton);
```

//-//

```
let hr2 = ce("hr");
hr2.className = "hrCompact";
theMenu.append(hr2);
```

//-//

```
let customButton = ce("button");
customButton.textContent = "Custom";
customButton.className = "ourButton";
customButton.onclick = customPlayerName;
theMenu.append(customButton);
}
```



```
function customPlayerName()
{
    let theName = prompt("Enter Name")

    // name of Player
    ge(activePlayer).style.innerHTML = theName;

    // update textarea, to show name entered
    ge("thePlayerName").value = theName;

    playerArray[0].name = theName;

    ge(activePlayer).innerHTML = theName;

    showPlayerData();

    ge("ourPlayerName").innerHTML = '<hr class
= "hrCompact">' + theName + '<hr class =
"hrCompact">';
}

// Read the 3 Textarea, for the Name value of
our Player
// Set the Name on our Player
```

```
function customPlayerNameFromTextInput()
```

```
{  
    let theName = ge("thePlayerName").value;  
  
    ge(activePlayer).style.innerHTML= theName;  
  
    playerArray[0].name = theName;  
  
    ge(activePlayer).innerHTML = theName;  
  
    showPlayerData();  
}
```

```
//--//
```

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

```
// menuPlayerOpacity.js
```

```
function menuPlayerOpacity()  
{  
    audioPlay("sfx_blip_001", 1.0);  
}
```

```
// if menu exists, remove it
if (ge("menuOpacityPlayer"))
{
    removeElement('menuOpacityPlayer');
}
```

```
//-//
```

```
let theMenu = ce("div");
theMenu.id = "menuOpacityPlayer";
theMenu.className = "divStyle001";
theMenu.style.position = "absolute";
theMenu.style.left = ourPlayer.x + 100 + 'px';
theMenu.style.top = ourPlayer.y - 55 + 'px';
theMenu.style.width = '440px';
theMenu.style.height = '193px';
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");
closeButton.textContent = 'X';
closeButton.className = 'ourButton';
closeButton.style.marginBottom = '-4px';
closeButton.onclick = function()
```

```
{
  removeElement("menuOpacityPlayer");
  keyboardControlOn();
};
theMenu.append(closeButton);

//--//

theMenu.append(ce('hr'));

//--//

let titleSpan = ce("span");
titleSpan.style.color = 'rgb(255, 255, 255)';
titleSpan.textContent = 'OPACITY OF
PLAYER';
theMenu.append(titleSpan);

//--//

theMenu.append(ce('br'));

//--//

let textNode = ce('div');
textNode.textContent = 'Alpha ';
```

```
theMenu.append(textNode);
```

```
//-//
```

```
let opacityTextarea = ce("textarea");  
opacityTextarea.id = 'theOpacity';  
opacityTextarea.className = 'textInput';  
opacityTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(opacityTextarea);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let enterButton = ce("button");  
enterButton.textContent = 'Enter';  
enterButton.className = 'ourButton';  
enterButton.onclick = function()  
{  
    let theOpacity = ge("theOpacity").value;
```

```
ge(activePlayer).style.opacity =  
theOpacity;
```

```
keyboardControlOn();
```

```
enterButtonSound();
```

```
updatePlayer();  
};  
theMenu.append(enterButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let opacityButtons = [  
    "Default", 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,  
0.9, 1.0  
];
```

```
for (let z = 0; z < opacityButtons.length; z++)  
{  
    let opacityButton = ce("button");
```

```
opacityButton.textContent =  
opacityButtons[z];  
opacityButton.className = 'ourButton';  
opacityButton.onclick = function()  
{  
    ge(activePlayer).style.opacity =  
opacityButtons[z];  
    ge("theOpacity").value =  
opacityButtons[z];  
  
    updatePlayer();  
};  
theMenu.append(opacityButton);  
}
```

//-//

```
theMenu.append(ce('br'));
```

//-//

```
let customButton = ce("button");  
customButton.textContent = 'Custom';  
customButton.className = 'ourButton';  
customButton.onclick = function()  
{
```

```
    customOpacity();  
};  
theMenu.append(customButton);
```

```
//-//
```

```
    theMenu.append(ce('br'));  
}
```

```
//-//
```

```
function customOpacity()  
{  
    let theOpacity = prompt("Enter Opacity as  
0.0 to 1.0")  
  
    ge(activePlayer).style.opacity = theOpacity;  
    ge("theOpacity").value = theOpacity;  
  
    updatePlayer();  
}
```

```
//--//
```

```
// Dedicated to God the Father
```



```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
  
// menuPlayerPositionGet.js
```

```
let currentPosTimer;
```

```
function menuPlayerPositionGet()  
{  
    audioPlay("sfx_blip_001", 1.0);  
  
    if (ge('menuPositionPlayerGet'))  
    {  
        removeElement('menuPositionPlayerGet');  
    }  
}
```

```
//-//
```

```
let theMenu = ce("div");  
theMenu.id = "menuPositionPlayerGet";  
theMenu.className = "divStyle001";  
theMenu.title = "menuPlayerPositionGet()";  
theMenu.style.position = "absolute";
```

```
theMenu.style.left = ourPlayer.x + 100 + 'px';  
theMenu.style.top = ourPlayer.y - 55 + 'px';  
theMenu.style.width = 400 + 'px';  
theMenu.style.height = 220 + 'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.onclick = function()  
{
```

```
removeElement("menuPositionPlayerGet");  
    keyboardControlOn();  
};  
closeButton.innerHTML = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let span = ce("span");  
span.style.color = "rgb(255, 255, 255)";  
span.innerHTML = "GET PLAYER  
POSITION";  
theMenu.append(span);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let buttonContainer = ce('div');  
theMenu.append(buttonContainer);
```

```
//-//
```

```
let playerPosButton = ce("button");  
playerPosButton.className = "ourButton";  
playerPosButton.onclick = function()  
{  
    playerPositionGet();  
};  
playerPosButton.innerHTML = "GET POS";  
buttonContainer.append(playerPosButton);
```

```
//-//
```

```
buttonContainer.append(ce("br"));
```

```
//-//
```

```
let playerPosRepeatButton = ce("button");  
playerPosRepeatButton.className =  
"ourButton";  
playerPosRepeatButton.onclick = function()  
{  
    playerPositionGetRepeat();  
};  
playerPosRepeatButton.innerHTML = "GET  
POS REPEAT";
```

```
buttonContainer.append(playerPosRepeatButt  
on);
```

```
//-//
```

```
buttonContainer.append(ce("br"));
```

```
//-//
```

```
let clearPosTimeButton = ce("button");
```

```
clearPosTimeButton.className =  
"ourButton";  
clearPosTimeButton.onclick = function()  
{  
    clearCurrentPosTimer();  
};  
clearPosTimeButton.innerHTML = "Cancel  
Get Pos";  
  
buttonContainer.append(clearPosTimeButton);  
  
//-//  
  
buttonContainer.append(ce("br"));  
  
//-//  
  
buttonContainer.append(ce("br"));  
  
//-//  
  
theMenu.append(ce("hr"));  
  
//-//  
  
let xPosLabel = ce("span");
```

```
xPosLabel.innerHTML = "X POS &nbsp;";  
theMenu.append(xPosLabel);
```

```
//-//
```

```
let xPosInput = ce("textarea");  
xPosInput.id = "gotX";  
xPosInput.className = "textInput";  
xPosInput.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(xPosInput);
```

```
//-//
```

```
let yPosLabel = ce("span");  
yPosLabel.innerHTML = "Y POS &nbsp;";  
theMenu.append(yPosLabel);
```

```
//-//
```

```
let yPosInput = ce("textarea");  
yPosInput.id = "gotY";  
yPosInput.className = "textInput";  
yPosInput.onkeydown = function()
```

```
{  
    keyboardControlOff();  
};  
theMenu.append(yPosInput);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let cancelButton = ce("button");  
cancelButton.innerHTML = "Cancel Timer";  
cancelButton.onclick = function()  
{  
    clearInterval(currentPosTimer);  
};  
theMenu.append(cancelButton);  
}
```

```
//-//
```

```
function playerPositionGet()  
{  
    let currentPosition =  
positionGet("thePlayer");
```

```
ge(activePlayer).innerHTML =  
currentPosition.x.toFixed(0) + "<br>" +  
currentPosition.y.toFixed(0) + "<br>";
```

```
ge('gotX').value = currentPosition.x;  
ge('gotY').value = currentPosition.y;  
}
```

```
function playerPositionGetRepeat()  
{  
    currentPosTimer = setInterval(function()  
    {  
        playerPositionGet();  
    }, 1000);  
}
```

```
function clearCurrentPosTimer()  
{  
    clearInterval(currentPosTimer);  
}
```

```
//--//
```

```
// Dedicated to God the Father
```



```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// menuPlayerPositionSet.js
```

```
function menuPlayerPositionSet()  
{  
    audioPlay("sfx_blip_001", 1.0);  
  
    //-//  
  
    if (ge('menuPositionPlayerSet'))  
    {  
        removeElement('menuPositionPlayerSet');  
    }  
  
    let theMenu = ce("div");  
    theMenu.id = "menuPositionPlayerSet";  
    theMenu.className = "divStyle001";  
    theMenu.style.position = "absolute";  
    theMenu.style.left = ourPlayer.x + 100 + 'px';  
    theMenu.style.top = ourPlayer.y - 80 + 'px';  
    theMenu.style.width = 400 + 'px';
```

```
theMenu.style.height = 200 + 'px';  
theMenu.style.resize = 'both';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.onclick = function()  
{
```

```
removeElement("menuPositionPlayerSet");  
    keyboardControlOn();  
};  
closeButton.innerHTML = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let span = ce("span");  
span.style.color = "rgb(255, 255, 255)";
```

```
span.innerHTML = "PLAYER POSITION  
OPTIONS";
```

```
theMenu.append(span);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let buttonContainer = ce('div');  
theMenu.append(buttonContainer);
```

```
//-//
```

```
let positionValues = [  
    1, 2, 3, 4, 5, 6, 7, 8, 9, 10  
];
```

```
for (let x = 0; x < positionValues.length; x++)  
{  
    let button1 = ce("button");  
    button1.className = "ourButton";  
    button1.onclick = function()  
    {
```

```
        playerMoveTo(activePlayer,  
positionValues[x]);  
    };  
    button1.innerHTML = positionValues[x];  
    buttonContainer.append(button1);  
}  
  
//-//  
  
buttonContainer.append(ce("br"));  
  
//-//  
  
theMenu.append(ce("hr"));  
  
//-//  
  
let customButton = ce("button");  
customButton.className = "ourButton";  
customButton.onclick = function()  
{  
    customPosition();  
};  
customButton.innerHTML = "Custom Player  
Position";  
theMenu.append(customButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let label1 = ce("span");  
label1.innerHTML = "X POS &nbsp;";  
theMenu.append(label1);
```

```
//-//
```

```
let textArea1 = ce("textarea");  
textArea1.id = "customPosX";  
textArea1.className = "textInput";  
textArea1.onkeydown = function() {  
    keyboardControlOff();  
};  
theMenu.append(textArea1);
```

```
//-//
```

```
theMenu.append(ce('br'));
```

```
//-//
```

```
let label2 = ce("span");  
label2.innerHTML = "Y POS &nbsp;";  
theMenu.append(label2);
```

```
//-//
```

```
let textArea2 = ce("textarea");  
textArea2.id = "customPosY";  
textArea2.className = "textInput";  
textArea2.onkeydown = function() {  
    keyboardControlOff();  
};  
theMenu.append(textArea2);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let enterButton = ce("button");  
enterButton.className = "ourButton";  
enterButton.onclick = function()  
{  
    customPositionText();
```

```
    keyboardControlOn();
    enterButtonSound();
};
enterButton.innerHTML = "Enter";
theMenu.append(enterButton);
}

// move player to a preset location, based on
// which button was pressed
function playerMoveTo(whichId, theLocation)
{
    audioPlay("sfx_blip_001", 1.0);

    if (theLocation == '1')
    {
        ourPlayer.x = 100;
        ourPlayer.y = 700;
    }

    if (theLocation == '2')
    {
        ourPlayer.x = 500;
        ourPlayer.y = 700;
    }

    if (theLocation == '3')
```

```
{  
    ourPlayer.x = 1000;  
    ourPlayer.y = 700;  
}
```

```
if (theLocation == '4')  
{  
    ourPlayer.x = 100;  
    ourPlayer.y = 200;  
}
```

```
if (theLocation == '5')  
{  
    ourPlayer.x = 500;  
    ourPlayer.y = 200;  
}
```

```
if (theLocation == '6')  
{  
    ourPlayer.x = 1000;  
    ourPlayer.y = 200;  
}
```

```
if (theLocation == '7')  
{  
    ourPlayer.x = 700;
```



```
    ourPlayer.y = 700;
}

if (theLocation == '8')
{
    ourPlayer.x = 50;
    ourPlayer.y = 900;
}

if (theLocation == '9')
{
    ourPlayer.x = 50;
    ourPlayer.y = 100;
}

if (theLocation == '0')
{
    ourPlayer.x = 300;
    ourPlayer.y = 700;
}

movePlayer(activePlayer);

// update the playerArray posX
playerArray[0].posX = ourPlayer.x;
```

```
// update the playerArray posY  
playerArray[0].posY = ourPlayer.y;
```

```
// update the textArea of the posX and posY  
ge("customPosX").value = ourPlayer.x;  
ge("customPosY").value = ourPlayer.y
```

```
// show the playerArray data in the textarea  
that has the id named textAreaPlayerData  
showPlayerData();
```

```
// update playerArray to show the changed  
value  
// updatePlayer();  
}
```

```
function customPosition()  
{
```

```
    let playerX = prompt("Enter X Position");  
    let playerY = prompt("Enter Y Position");
```

```
// if person doesn't input a value, then keep  
the player where they are  
if (playerX == " || playerY == ")  
{  
    ourPlayer.x = ourPlayer.x;
```

```
    ourPlayer.y = ourPlayer.y;
    return;
}

// x location of the player
ourPlayer.x= parseFloat(playerX);

// y location of the player
ourPlayer.y= parseFloat(playerY);

// if person did answer Y position, move
player
if (playerY != "")
{
    movePlayer(activePlayer);
}

// update the playerArray posX
playerArray[0].posX = ourPlayer.x;

// update the playerArray posY
playerArray[0].posY = ourPlayer.y;

// show the playerArray data in the textarea
that has the id named textAreaPlayerData
showPlayerData();
```

```
ge("customPosX").value = ourPlayer.x;
ge("customPosY").value = ourPlayer.y;

// update playerArray to show changed value
// updatePlayer();
}

function customPositionText()
{
    let playerX = ge("customPosX").value;
    let playerY = ge("customPosY").value;

    // if person doesn't input a value, then keep
    person where they are
    if (playerY == "")
    {
        ourPlayer.x = ourPlayer.x;
        ourPlayer.y = ourPlayer.y;
        return;
    }

    // x location of the player
    ourPlayer.x= parseFloat(playerX);

    // y location of the player
```

```
ourPlayer.y= parseFloat(playerY);  
  
// if person did answer Y position, move  
player  
if (playerY != "")  
{  
    movePlayer(activePlayer);  
}  
  
// update the playerArray posX  
playerArray[0].posX = ourPlayer.x;  
  
// update the playerArray posY  
playerArray[0].posY = ourPlayer.y;  
  
// show the playerArray data in the textarea  
that has the id named textAreaPlayerData  
showPlayerData();  
  
// update playerArray to show changed value  
// updatePlayer();  
}  
  
//--//
```

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// menuPlayerSizeGet.js
```

```
function menuPlayerSizeGet()  
{  
    audioPlay("sfx_blip_001", 1.0);  
  
    //-//  
  
    if (ge("menuSizePlayerGet"))  
    {  
        removeElement("menuSizePlayerGet");  
    }  
  
    let theMenu = ce("div");  
    theMenu.id = "menuSizePlayerGet";  
    theMenu.className = "divStyle001";  
    theMenu.style.position = "absolute";  
    theMenu.style.left = ourPlayer.x + 100 + 'px';  
    theMenu.style.top = ourPlayer.y - 41 + 'px';  
    theMenu.style.width = 400 + 'px';
```

```
theMenu.style.height = 170 + 'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{  
    removeElement("menuSizePlayerGet");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
let titleDiv = ce("div");  
theMenu.append(titleDiv);
```

```
//-//
```

```
let titleSpan = ce("span");  
titleSpan.style.color = "rgb(255, 255, 255)";  
titleSpan.textContent = "GET PLAYER SIZE";
```

```
titleDiv.append(titleSpan);
```

```
//-//
```

```
titleDiv.append(ce("hr"));
```

```
//-//
```

```
titleDiv.append(ce("hr"));
```

```
//-//
```

```
let getSizeButton = ce("button");  
getSizeButton.onclick = function()  
{  
    playerSizeGet();  
};  
getSizeButton.className = "ourButton";  
getSizeButton.textContent = "GET SIZE";  
theMenu.append(getSizeButton);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```



```
let xSizeLabel = ce('div');  
xSizeLabel.textContent = "X Size ";  
theMenu.append(xSizeLabel);
```

```
//-//
```

```
let xSizeTextarea = ce("textarea");  
xSizeTextarea.id = "gotSizeX";  
xSizeTextarea.className = "textInput";  
xSizeTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(xSizeTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let ySizeLabel = ce('div');  
ySizeLabel.textContent = "Y Size ";  
theMenu.append(ySizeLabel);
```

//-//

```
let ySizeTextarea = ce("textarea");
ySizeTextarea.id = "gotSizeY";
ySizeTextarea.className = "textInput";
ySizeTextarea.onkeydown = function()
{
    keyboardControlOff();
};
theMenu.append(ySizeTextarea);
}
```

//-//

```
function playerSizeGet()
{
    let currentSize = sizeGet("thePlayer");

    ge('gotSizeX').value = currentSize.x;
    ge('gotSizeY').value = currentSize.y;

    ge(activePlayer).innerHTML = currentSize.x +
    "<br>" + currentSize.y + "<br>";
}
```

//--//

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

```
// menuPlayerSizeSet.js
```

```
function menuPlayerSizeSet()
{
    audioPlay("sfx_blip_001", 1.0);

    if (ge("menuSizePlayerSet"))
    {
        removeElement("menuSizePlayerSet");
    }

    let theMenu = ce("div");
    theMenu.id = "menuSizePlayerSet";
    theMenu.className = "divStyle001";
    theMenu.style.position = "absolute";
    theMenu.style.left = ourPlayer.x + 100 + 'px';
    theMenu.style.top = ourPlayer.y - 50 + 'px';
    theMenu.style.width = 400 + 'px';
```

```
theMenu.style.height = 190 + 'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{  
    removeElement("menuSizePlayerSet");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let title = ce("span");  
title.style.color = "rgb(255, 255, 255)";  
title.textContent = " SET PLAYER SIZE ";  
theMenu.append(title);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let defaultButton = ce("button");  
defaultButton.onclick =  
playerSizeSetDefault;  
defaultButton.className = "ourButton";  
defaultButton.textContent = "Default";  
theMenu.append(defaultButton);
```

```
//-//
```

```
let xSizeLabel = ce("div");  
xSizeLabel.textContent = "X SIZE ";  
let xSizeInput = ce("textarea");  
xSizeInput.id = "customSizeX";  
xSizeInput.className = "textInput";  
xSizeInput.onkeydown =  
keyboardControlOff;  
xSizeLabel.append(xSizeInput);  
theMenu.append(xSizeLabel);
```

//-//

```
let ySizeLabel = ce("div");
ySizeLabel.textContent = "Y SIZE ";
let ySizeInput = ce("textarea");
ySizeInput.id = "customSizeY";
ySizeInput.className = "textInput";
ySizeInput.onkeydown =
keyboardControlOff;
ySizeLabel.append(ySizeInput);
theMenu.append(ySizeLabel);
```

//-//

```
let enterButton = ce("button");
enterButton.onclick = function()
{
    playerSizeSet();
    keyboardControlOn();
    enterButtonSound();
};
enterButton.className = "ourButton";
enterButton.textContent = "Enter";
theMenu.append(enterButton);
}
```

```
function playerSizeSet()
{
    // let currentSize = sizeGet("thePlayer");

    let theSizeX;
    let theSizeY;

    theSizeX = ge('customSizeX').value;
    theSizeY = ge('customSizeY').value;

    ge(activePlayer).innerHTML = theSizeX +
    "<br>" + theSizeY + "<br>";

    ge(activePlayer).style.width = theSizeX;
    ge(activePlayer).style.height = theSizeY;

    // update playerArray to show changed value
    updatePlayer();
}
```

```
function playerSizeSetDefault()
{
    // let currentSize = sizeGet("thePlayer");

    let theSizeX = 70;
    let theSizeY = 70;
```

```
ge('customSizeX').value = theSizeX;  
ge('customSizeY').value = theSizeY;
```

```
ge(activePlayer).innerHTML = theSizeX +  
"<br>" + theSizeY + "<br>";
```

```
ge(activePlayer).style.width = theSizeX;  
ge(activePlayer).style.height = theSizeY;
```

```
// update playerArray to show changed value  
updatePlayer();  
}
```

```
//--//
```

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

```
function menuPlayerSpeed()
```



```
{
  audioPlay("sfx_blip_001", 1.0);

  // if player menu already exists
  // remove it and create a new one
  if (ge("menuSpeedPlayer"))
  {
    removeElement("menuSpeedPlayer");
  }

  //-//

  let theMenu = ce("div");
  theMenu.id = "menuSpeedPlayer";
  theMenu.className = "divStyle001";
  theMenu.style.position = "absolute";
  theMenu.style.left = ourPlayer.x + 100 + 'px';
  theMenu.style.top = ourPlayer.y - 50 + 'px';
  theMenu.style.width = 400 + 'px';
  theMenu.style.height = 205 + 'px';
  ba(theMenu);

  //-//

  let closeButton = ce("button");
  closeButton.onclick = function()
```

```
{  
    removeElement("menuSpeedPlayer");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let title = ce("span");  
title.style.color = "rgb(255, 255, 255)";  
title.textContent = "SPEED OPTIONS";  
theMenu.append(title);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let speedValues = [
    1, 2, 3, 4, 5, 6, 7, 8, 9, 10
];

for (let z = 0; z < speedValues.length; z++)
{
    let speedButton = ce("button");
    speedButton.textContent =
speedValues[z];
    speedButton.className = 'ourButton';
    speedButton.onclick = function()
    {
        ourPlayer.speedMultiplier =
speedValues[z];

        ge("textSpeed").value = speedValues[z];

        updatePlayer();
    };
    theMenu.append(speedButton);
}

//--//

let customButton = ce("button");
customButton.onclick = customSpeed;
```

```
customButton.className = "ourButton";  
customButton.textContent = "Custom";  
theMenu.append(customButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let speedLabel = ce("div");  
speedLabel.textContent = "SPEED ";  
let speedInput = ce("textarea");  
speedInput.id = "textSpeed";  
speedInput.className = "textInput";  
speedInput.onkeydown =  
keyboardControlOff;  
speedLabel.append(speedInput);  
theMenu.append(speedLabel);
```

```
//-//
```

```
let enterButton = ce("button");  
enterButton.onclick = function()  
{  
    customSpeedFromTextInput();
```

```
        keyboardControlOn();
        enterButtonSound();
    };
    enterButton.className = "ourButton";
    enterButton.textContent = "Enter";
    theMenu.append(enterButton);
}

function customSpeed()
{
    let theSpeed = prompt("Enter Speed 1 to
1000 etc")

    ourPlayer.speedMultiplier = theSpeed;

    if (theSpeed == " " || theSpeed == null)
    {
        ourPlayer.speedMultiplier = 10;
    }

    ge("textSpeed").value =
ourPlayer.speedMultiplier;

    ge('infoDiv').innerHTML = 'Speed: ' +
ourPlayer.speedMultiplier;
```

```
// update playerArray to show changed value
updatePlayer();
}

function customSpeedFromTextInput()
{
    let theSpeed = ge("textSpeed").value;

    ourPlayer.speedMultiplier = theSpeed;

    ge('infoDiv').innerHTML = 'Speed: ' +
ourPlayer.speedMultiplier;

    // update playerArray to show changed value
    updatePlayer();
}

//--//

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

```
// menuPlayerTexture.js
```

```
let textureArray =
```

```
[  
  {  
    name: "Default",  
    textureOffline: "None",  
    textureOnline: "None"  
  },  
  {  
    name: "Concrete 1",  
    textureOffline:  
"src/media/textures/concrete1.jpg",  
    textureOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/concrete1.jpg"  
  },  
  {  
    name: "Wood 1",  
    textureOffline:  
"./src/media/textures/wood1.jpg",  
    textureOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/wood1.jpg"  
  },  
  {  

```

```
    name: "Zebra 1",  
    textureOffline:  
"src/media/textures/zebra1.jpg",  
    textureOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/zebra1.jpg"  
},  
{  
    name: "Boxing Glove",  
    textureOffline:  
"src/media/textures/boxingglove.png",  
    textureOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/boxingglove.png"  
},  
{  
    name: "Jane",  
    textureOffline:  
"src/media/textures/jane.png",  
    textureOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/jane.png"  
}  
];
```

function menuPlayerTexture()


```
{  
  audioPlay("sfx_blip_001", 1.0);  
  
  if (ge("menuTexturePlayer"))  
  {  
    removeElement("menuTexturePlayer");  
  }  
  
  //-//  
  
  let theMenu = ce("div");  
  theMenu.id = "menuTexturePlayer";  
  theMenu.className = "divStyle001";  
  theMenu.style.position = "absolute";  
  theMenu.style.left = ourPlayer.x + 100 + 'px';  
  theMenu.style.top = ourPlayer.y - 50 + 'px';  
  theMenu.style.width = 400 + 'px';  
  theMenu.style.height = 205 + 'px';  
  ba(theMenu);  
  
  //-//  
  
  let closeButton = ce("button");  
  closeButton.onclick = function()  
  {  
    removeElement("menuTexturePlayer");
```

```
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let theTitle = ce("span");  
theTitle.style.color = "rgb(255, 255, 255)";  
theTitle.textContent = "TEXTURE OPTIONS";  
theMenu.append(theTitle);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
for (let z = 0; z < textureArray.length; z++)  
{  
    let textureButton = ce("button");
```

```

textureButton.onclick = function()
{
    if (online == true)
    {

ge(activePlayer).style.backgroundImage =
'url('+textureArray[z].textureOnline+');
    }
    else if (online == false)
    {

ge(activePlayer).style.backgroundImage =
'url('+textureArray[z].textureOffline+');
    }
    updatePlayer();
};
textureButton.className = "ourButton";
textureButton.textContent =
textureArray[z].name;
theMenu.append(textureButton);

// - //

theMenu.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
```

```
// menuSearch.js
```

```
function menuSearch()
```

```
{
```

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

```
//--//
```

```
if (ge("menuSearchId"))
```

```
{
```

```
    removeElement("menuSearchId");
```

```
}
```

```
let theMenu = ce("div");
```

```
theMenu.id = "menuSearchId";
```

```
theMenu.className = "divStyle001";
```

```
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 150 + 'px';  
theMenu.style.top = ourPlayer.y + 'px';  
theMenu.style.width = 300 + 'px';  
theMenu.style.height = 180 + 'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{  
    removeElement(theMenu.id);  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let theTitle = ce("span");
```

```
theTitle.style.color = "rgb(255, 255, 255)";  
theTitle.innerHTML = "<b> SEARCH </b>";  
theMenu.append(theTitle);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let instruction1 = ce("b");  
instruction1.textContent = "Control + F";  
theMenu.append(instruction1);
```

```
//-//
```

```
let instruction2 = ce("span");  
instruction2.textContent = " to Search for  
Word or Phrase";  
theMenu.append(instruction2);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let instruction3 = ce("b");  
instruction3.textContent = "Left Click";  
theMenu.append(instruction3);
```

```
//-//
```

```
let instruction4 = ce("span");  
instruction4.textContent = " to Place Player  
Near Found Word or Phrase";  
theMenu.append(instruction4);  
}
```

```
//--//
```

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

```
// menuStructureColor.js
```

```
let theColorID;
```

```
function menuStructureColor(theStructureId)
{
    audioPlay("sfx_blip_001", 1.0);

    theColorID = ge(theStructureId);

    if (ge('menuColorStructure'))
    {
        removeElement('menuColorStructure');
    }

    let theMenu = ce("div");
    theMenu.id = "menuColorStructure";
    theMenu.className = "divStyle001";
    theMenu.style.position = "absolute";
    theMenu.style.left = ourPlayer.x + 100 + 'px';
    theMenu.style.top = ourPlayer.y - 75 + 'px';
    theMenu.style.width = 400 + 'px';
    theMenu.style.height = 225 + 'px';
    ba(theMenu);

    //-//

    let closeButton = ce("button");
    closeButton.onclick = function()
    {
```



```
removeElement("menuColorStructure");  
keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);  
  
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let theTitle = ce("span");  
theTitle.style.color = "rgb(255, 255, 255)";  
theTitle.textContent = "COLOR  
STRUCTURE";  
theMenu.append(theTitle);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let redLabel = ce("span");  
redLabel.textContent = "R ";  
theMenu.append(redLabel);
```

```
//-//
```

```
let redInput = ce("textarea");  
redInput.id = "theRed";  
redInput.className = "textInput";  
redInput.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(redInput);
```

```
//-//
```

```
let greenLabel = ce("span");  
greenLabel.textContent = "G ";  
theMenu.append(greenLabel);
```

```
//-//
```

```
let greenInput = ce("textarea");  
greenInput.id = "theGreen";  
greenInput.className = "textInput";
```

```
greenInput.onkeydown = function()
{
    keyboardControlOff();
};
theMenu.append(greenInput);

//-
```

```
let blueLabel = ce("span");
blueLabel.textContent = "B ";
theMenu.append(blueLabel);

//-
```

```
let blueInput = ce("textarea");
blueInput.id = "theBlue";
blueInput.className = "textInput";
blueInput.onkeydown = function()
{
    keyboardControlOff();
};
theMenu.append(blueInput);

//-
```

```
let enterButton = ce("button");
```

```
enterButton.onclick = function()
{
    customStructureColorFromTextInput();
    keyboardControlOn();
    enterButtonSound();
};
enterButton.className = "ourButton";
enterButton.textContent = "Enter";
theMenu.append(enterButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
for (let z = 0; z < colorChoices.length; z++)
{
    let colorChoiceButton = ce("button");
    colorChoiceButton.textContent =
colorChoices[z].name;
    colorChoiceButton.className =
'ourButton';
    colorChoiceButton.onclick = function()
    {
```

```
ge(theStructureId).style.backgroundColor =  
"rgb("+colorChoices[z].color+"");
```

```
    // split color value to array of rgb  
    let rgbValues =  
colorChoices[z].color.split(',');
```

```
        ge("theRed").value =  
rgbValues[0].trim();  
        ge("theGreen").value =  
rgbValues[1].trim();  
        ge("theBlue").value =  
rgbValues[2].trim();
```

```
        updateStructures(theStructureId);  
    };  
    theMenu.append(colorChoiceButton);  
}
```

```
//-//
```

```
let customButton = ce("button");  
customButton.onclick = function()  
{  
    customStructureColor();
```

```
};  
customButton.className = "ourButton";  
customButton.textContent = "Custom";  
theMenu.append(customButton);  
}
```

```
function customStructureColor()  
{  
    let theColor = prompt("Enter RGB as  
255,255,255")  
  
    let theColorFormatted = 'rgb('+theColor+')';  
  
    // color of structure  
    theColorID.style.backgroundColor =  
theColorFormatted;  
  
    let colorSeparated = theColor.split(',');  
  
    // fill in the textarea, to show the value of our  
choice  
    ge("theRed").value = colorSeparated[0];  
    ge("theGreen").value = colorSeparated[1];  
    ge("theBlue").value = colorSeparated[2];  
}
```

```
// update structuresArray to show changed
value
    updateStructures(theStructureId);
}

// Read the 3 Textarea, for the red, green, blue
values of our structure
// Set the color on our player
function customStructureColorFromTextInput()
{
    let r = ge("theRed").value;
    let g = ge("theGreen").value;
    let b = ge("theBlue").value;

    theColorID.style.backgroundColor=
    "rgb("+r+", "+g+", "+b+")";

    // update structuresArray to show changed
    value
        updateStructures(theStructureId);
}

//--//

// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// menuStructureColorText.js
```

```
let theTextID;
```

```
function  
menuStructureColorText(ourStructureId)
```

```
{  
    if (ge('menuTextColorStructure'))  
    {
```

```
removeElement('menuTextColorStructure');  
    }
```

```
    theTextID = ge(ourStructureId);
```

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

```
//-//
```

```
let theMenu = ce("div");
```



```
theMenu.id = "menuTextColorStructure";  
theMenu.className = "divStyle001";  
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 100 + 'px';  
theMenu.style.top = ourPlayer.y - 75 + 'px';  
theMenu.style.width = 400 + 'px';  
theMenu.style.height = 225 + 'px';  
ba(theMenu)
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{
```

```
removeElement("menuTextColorStructure");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let theTitle = ce("span");  
theTitle.style.color = "rgb(255, 255, 255)";  
theTitle.textContent = "COLOR TEXT";  
theMenu.append(theTitle);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let redLabel = ce("span");  
redLabel.textContent = "R ";  
theMenu.append(redLabel);
```

```
//-//
```

```
let redInput = ce("textarea");  
redInput.id = "theRed";  
redInput.className = "textInput";  
redInput.onkeydown = function()  
{  
    keyboardControlOff();
```

```
};  
theMenu.append(redInput);
```

```
//-//
```

```
let greenLabel = ce("span");  
greenLabel.textContent = "G ";  
theMenu.append(greenLabel);
```

```
//-//
```

```
let greenInput = ce("textarea");  
greenInput.id = "theGreen";  
greenInput.className = "textInput";  
greenInput.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(greenInput);
```

```
//-//
```

```
let blueLabel = ce("span");  
blueLabel.textContent = "B ";  
theMenu.append(blueLabel);
```

//-//

```
let blueInput = ce("textarea");
blueInput.id = "theBlue";
blueInput.className = "textInput";
blueInput.onkeydown = function()
{
    keyboardControlOff();
};
theMenu.append(blueInput);
```

//-//

```
let enterButton = ce("button");
enterButton.onclick = function()
{
    customStructureTextColorFromTextInput();
    keyboardControlOn();
    enterButtonSound();
};
enterButton.className = "ourButton";
enterButton.textContent = "Enter";
theMenu.append(enterButton);
```

//-//

```
theMenu.append(ce("hr"));

// - //

for (let z = 0; z < colorChoices.length; z++)
{
    let colorChoiceButton = ce("button");
    colorChoiceButton.textContent =
colorChoices[z].name;
    colorChoiceButton.className =
'ourButton';
    colorChoiceButton.onclick = function()
    {
        ge(ourStructureId).style.color =
"rgb("+colorChoices[z].color+"");

        // split color value to array of rgb
        let rgbValues =
colorChoices[z].color.split(',');

        ge("theRed").value =
rgbValues[0].trim();
        ge("theGreen").value =
rgbValues[1].trim();
```

```
    ge("theBlue").value =  
    rgbValues[2].trim();
```

```
        updateStructures(ourStructureId);  
    };  
    theMenu.append(colorChoiceButton);  
}
```

```
//-//
```

```
let customButton = ce("button");  
customButton.className = "ourButton";  
customButton.textContent = "Custom";  
customButton.onclick = function()  
{  
    customStructureFontColor();  
};  
theMenu.append(customButton);  
}
```

```
function customStructureFontColor()  
{  
    let theColor = prompt("Enter RGB as  
255,255,255")
```

```
    let theColorFormatted = 'rgb('+theColor+');
```

```
// color of text on player object
theTextID.style.color = theColorFormatted;

let colorSeparated = theColor.split(',');

// fill in the textarea, to show the value of our
choice
ge("redFontId").value = colorSeparated[0];
ge("greenFontId").value = colorSeparated[1];
ge("blueFontId").value = colorSeparated[2];

// update structuresArray to show the
changed value
updateStructures(ourStructureId);
}

// Read the 3 Textarea, for the red, green, blue
values of our structure text color
// Set the text color of our structure
function
customStructureTextColorFromTextInput()
{
    let r = ge("theRed").value;
    let g = ge("theGreen").value;
    let b = ge("theBlue").value;
```

```
theTextID.style.color = "rgb("+r+", "+g+",  
"+b+")";
```

```
// fill in the textarea, to show the value of our  
choice
```

```
ge("theRed").value = r;  
ge("theRed").value = g;  
ge("theRed").value = b;
```

```
// update structuresArray, show changed  
value
```

```
updateStructures(ourStructureId);  
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// menuStructureDistance.js
```


let theDistanceID;

function

menuStructureDistance(theStructureId)

{

audioPlay("sfx_blip_001", 1.0);

theDistanceID = theStructureId;

if (ge("menuPositionDistanceStructure"))

{

removeElement("menuPositionDistanceStructure");

}

let theMenu = ce("div");

theMenu.id =

"menuPositionDistanceStructure";

theMenu.className = "divStyle001";

theMenu.style.position = "absolute";

theMenu.style.left = ourPlayer.x + 100 + 'px';

theMenu.style.top = ourPlayer.y - 50 + 'px';

theMenu.style.width = 400 + 'px';

theMenu.style.height = 160 + 'px';

ba(theMenu);

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{
```

```
removeElement("menuPositionDistanceStructu  
re");
```

```
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let theTitle = ce("span");  
theTitle.style.color = "rgb(255, 255, 255)";  
theTitle.textContent = "Distance";  
theMenu.append(theTitle);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let idLabel = ce("span");  
idLabel.textContent = "Id of Object ";  
theMenu.append(idLabel);
```

```
//-//
```

```
let idTextArea = ce("textarea");  
idTextArea.id = "idToFindDistance";  
idTextArea.className = "textInput";  
idTextArea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(idTextArea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

//-//

```
let enterButton = ce("button");
enterButton.onclick = function()
{
    keepGettingXYStructureCustom();
    keyboardControlOn();
    enterButtonSound();
};
enterButton.className = "ourButton";
enterButton.textContent = "Enter";
theMenu.append(enterButton);
```

//-//

```
theMenu.append(ce("br"));
```

//-//

```
theMenu.append(ce('hr'));
```

//-//

```
let cancelButton = ce("button");
cancelButton.onclick = function()
{
```

```
    removeTimer(ourDistanceTimer);  
};  
cancelButton.className = "ourButton";  
cancelButton.textContent = "Cancel";  
theMenu.append(cancelButton);
```

```
//-//
```

```
    theMenu.append(ce("br"));  
}
```

```
let ourDistanceTimer;  
function keepGettingXYStructureCustom()  
{
```

```
    let obj1 = theDistanceID;
```

```
    let obj2 = ge("idToFindDistance").value;
```

```
    // clear any previous searches
```

```
    if (ourDistanceTimer)
```

```
    {  
        clearInterval(ourTimer);  
    }
```

```
    ourDistanceTimer = setInterval(function()  
    {
```

```
    getDistanceXY(obj1, obj2);  
    }, 1000);  
}
```

```
//--//
```

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

```
// menuStructureEdit.js
```

```
// when person clicks a structure, an edit menu  
// appears to edit that structure.
```

```
function menuStructureEdit(theStructureId)  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
    // let structureMenuId = ge(theStructureId);
```

```
// remove the text to find the digits only  
which number in the array, based on id number  
- 1 \d find digit
```

```
let theNumberOnlyStructure =  
theStructureId.replace(/\D/g,"") - 1;
```

```
//-//
```

```
// if menu already exists, remove it, and  
create a new menu wherever the player is  
if (ge("menuStructureEdit"))  
{  
    removeElement("menuStructureEdit");  
}
```

```
let theMenu = ce("div");  
theMenu.id = "menuStructureEdit";  
theMenu.className = "divStyle001";  
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 55 + 'px';  
theMenu.style.top = ourPlayer.y + 'px';  
theMenu.style.width = 250 + 'px';  
theMenu.style.height = 200 + 'px';  
theMenu.style.zIndex = 10000;  
theMenu.title = "menuStructureEdit";  
ba(theMenu);
```

```
//-//
```

```
// prevent menu from leaving screen left side  
if (positionGet("menuStructureEdit").x <= 30)  
{  
    theMenu.style.left = '40px';  
}
```

```
// prevent menu from leaving screen top side  
if (positionGet("menuStructureEdit").y <= 200)  
{  
    theMenu.style.top = '100px';  
}
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.className = "ourButton";  
closeButton.textContent = "X";  
closeButton.onclick = function()  
{  
    removeElement("menuStructureEdit");  
};  
theMenu.append(closeButton);
```



```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let idLabel = ce("span");
idLabel.className = "textCompact";
idLabel.innerHTML = `<b> '${theStructureId}'
</b>`;
theMenu.append(idLabel);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let structureNameText = ce("div");
structureNameText.id =
"structureNameText";
structureNameText.innerHTML = `<span
class='textCompact'> '$
{structuresArray[theNumberOnlyStructure].na
me}`;
```

```
theMenu.append(structureNameText);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let nameLabel = ce("span");  
nameLabel.className = "textCompact";  
nameLabel.textContent = "NAME";  
theMenu.append(nameLabel);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let nameButton = ce("button");  
nameButton.className = "ourButton";  
nameButton.textContent = "NAME";  
nameButton.onclick = function()  
{  
    menuStructureName(theStructureId);  
};
```

```
theMenu.append(nameButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let colorLabel = ce("span");  
colorLabel.className = "textCompact";  
colorLabel.textContent = "COLOR";  
theMenu.append(colorLabel);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let colorButton = ce("button");  
colorButton.className = "ourButton";  
colorButton.textContent = "COLOR";  
colorButton.onclick = function()  
{  
    menuStructureColor(theStructureId);  
};
```

```
theMenu.append(colorButton);
```

```
//-//
```

```
let colorTextButton = ce("button");  
colorTextButton.className = "ourButton";  
colorTextButton.textContent = "TEXT";  
colorTextButton.onclick = function()  
{  
    menuStructureColorText(theStructureId);  
};  
theMenu.append(colorTextButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let textureLabel = ce("span");  
textureLabel.className = "textCompact";  
textureLabel.textContent = "TEXTURE";  
theMenu.append(textureLabel);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let textureButton = ce("button");  
textureButton.className = "ourButton";  
textureButton.textContent = "TEXTURE";  
textureButton.onclick = function()  
{  
    menuStructureTexture(theStructureId);  
};  
theMenu.append(textureButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let alphaLabel = ce("span");  
alphaLabel.className = "textCompact";  
alphaLabel.textContent = "ALPHA";  
theMenu.append(alphaLabel);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let opacityButton = ce("button");  
opacityButton.className = "ourButton";  
opacityButton.textContent = "OPACITY";  
opacityButton.onclick = function()  
{  
    menuStructureOpacity(theStructureId);  
};  
theMenu.append(opacityButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let positionLabel = ce("span");  
positionLabel.className = "textCompact";  
positionLabel.textContent = "POSITION";  
theMenu.append(positionLabel);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let getPosButton = ce("button");  
getPosButton.className = "ourButton";  
getPosButton.textContent = "GET POS";  
getPosButton.onclick = function()  
{
```

```
menuStructurePositionGet(theStructureId);  
};  
theMenu.append(getPosButton);
```

```
//-//
```

```
let setPosButton = ce("button");  
setPosButton.className = "ourButton";  
setPosButton.textContent = "SET POS";  
setPosButton.onclick = function()  
{
```

```
menuStructurePositionSet(theStructureId);  
};  
theMenu.append(setPosButton);
```

```
//-//
```

```
theMenu.append(ce("hr"));
```

```
//-//
```

```
let sizeLabel = ce("span");  
sizeLabel.className = "textCompact";  
sizeLabel.textContent = "SIZE";  
theMenu.append(sizeLabel);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let getSizeButton = ce("button");  
getSizeButton.className = "ourButton";  
getSizeButton.textContent = "GET";  
getSizeButton.onclick = function()  
{  
    menuStructureSizeGet(theStructureId);  
};  
theMenu.append(getSizeButton);
```


//-//

```
let setSizeButton = ce("button");
setSizeButton.className = "ourButton";
setSizeButton.textContent = "SET";
setSizeButton.onclick = function()
{
    menuStructureSizeSet(theStructureId);
};
theMenu.append(setSizeButton);
```

//-//

```
theMenu.append(ce("hr"));
```

//-//

```
let distanceLabel = ce("span");
distanceLabel.className = "textCompact";
distanceLabel.textContent = "DISTANCE";
theMenu.append(distanceLabel);
```

//-//

```
theMenu.append(ce("br"));
```

//-//

```
let distanceButton = ce("button");
distanceButton.className = "ourButton";
distanceButton.textContent = "DISTANCE
FINDER";
distanceButton.onclick = function()
{
    menuStructureDistance(theStructureId);
};
theMenu.append(distanceButton);
```

//-//

```
theMenu.append(ce("hr"));
```

//-//

```
showStructureData();
}
```

//--//

// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

```
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalian

// menuStructureName.js

let worldNameId;

let theNameId;

let theNumberOnlyHere;

function menuStructureName(theStructureId)
{
    theNameId = ge(theStructureId);

    worldNameId = theStructureId;

    // remove the text to find the digits only
    // which number in the array, based on id
    number - 1
    // \d find digit
    theNumberOnlyHere =
    theStructureId.replace(/\D/g,"") - 1;
```

```
// console.log(theNumberOnlyHere + " check  
it out");
```

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

```
//-//
```

```
if (ge("menuNameStructure"))  
{  
    removeElement("menuNameStructure");  
}
```

```
let theMenu = ce("div");  
theMenu.id = "menuNameStructure";  
theMenu.className = "divStyle001";  
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 100 +'px';  
theMenu.style.top = ourPlayer.y - 35 + 'px';  
theMenu.style.width = 440 +'px';  
theMenu.style.height = 137 +'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()
```

```
{  
    removeElement("menuNameStructure");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "5px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
let nameLabel = ce('div');  
nameLabel.id = 'nameLabel';  
nameLabel.textContent = 'Name of  
Structure';  
theMenu.append(nameLabel);
```

```
//-//
```

```
let textInputName = ce('textarea');  
textInputName.id = 'theStructureName';  
textInputName.className = 'textInput';  
textInputName.onkeydown = function()  
{  
    keyboardControlOff();  
};
```

```
theMenu.append(textInputName);
```

```
//-//
```

```
let enterButton = ce("button");  
enterButton.onclick = function()  
{  
    customStructureNameFromTextInput();  
    keyboardControlOn();  
    enterButtonSound();  
};  
enterButton.className = "ourButton";  
enterButton.textContent = "Enter";  
theMenu.append(enterButton);  
}
```

```
// Read Textarea, for Name value of our  
structure
```

```
// Set the Name on our structure
```

```
function customStructureNameFromTextInput()  
{  
    let theName = ge("theStructureName").value;  
  
    // theNameID.style.innerHTML = theName;
```

```
structuresArray[theNumberOnlyHere].name  
= theName;
```

```
// Show the name of the structure and also  
the position data
```

```
theNameID.innerHTML = worldNameId +  
"<br>" +  
structuresArray[theNumberOnlyHere].name +  
"<br>" +  
positionGet(worldNameId).x.toFixed(0) +  
"<br>" +  
positionGet(worldNameId).y.toFixed(0);
```

```
ge("structureNameText").innerHTML = '<div  
style = "font-size: 15px">' +  
structuresArray[theNumberOnlyHere].name +  
'</div>';
```

```
showStructureData();  
}
```

```
//--//
```

```
// Dedicated to God the Father  
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalian

// menuStructureOpacity.js

let worldOpacityId;

let theOpacityID;

let theNumberOnly;

function menuStructureOpacity(theStructureId)
{
    theOpacityID = ge(theStructureId);
    worldOpacityId = theStructureId;

    // remove text to find digits only
    let theNumberOnly =
theStructureId.replace(/\D/g,"") - 1;

    //-//

    audioPlay("sfx_blip_001", 1.0);

    //-//
```



```
if (ge("menuOpacityStructure"))  
{  
    removeElement("menuOpacityStructure");  
}
```

```
//-//
```

```
let theMenu = ce("div");  
theMenu.id = "menuOpacityStructure";  
theMenu.className = "divStyle001";  
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 100 + 'px';  
theMenu.style.top = ourPlayer.y - 55 + 'px';  
theMenu.style.width = 440 + 'px';  
theMenu.style.height = 175 + 'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{  
    removeElement("menuOpacityStructure");  
    keyboardControlOn();  
};
```

```
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let opacityLabel = ce("span");  
opacityLabel.style.color = "rgb(255, 255,  
255)";  
opacityLabel.textContent = "OPACITY OF  
STRUCTURE";  
theMenu.append(opacityLabel);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let alphaText = ce("span");  
alphaText.textContent = "Alpha";
```

```
theMenu.append(alphaText);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

```
//-//
```

```
let alphaTextarea = ce("textarea");  
alphaTextarea.id = "theOpacity";  
alphaTextarea.className = "textInput";  
alphaTextarea.onkeydown = function()
```

```
{  
    keyboardControlOff();  
};
```

```
theMenu.append(alphaTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let enterButton = ce("button");  
enterButton.onclick = function()  
{
```

```
    customStructureOpacityFromTextInput();  
    keyboardControlOn();  
    enterButtonSound();  
};  
enterButton.className = "ourButton";  
enterButton.textContent = "Enter";  
theMenu.append(enterButton);  
  
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let opacityValues = [  
    0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,  
1.0  
];  
  
for (let z = 0; z < opacityValues.length; z++)  
{  
    let opacityButton = ce("button");  
    opacityButton.textContent =  
opacityValues[z];  
    opacityButton.className = 'ourButton';  
    opacityButton.onclick = function()
```

```
{
    ge(theStructureId).style.opacity =
opacityValues[z];

    ge("theOpacity").value =
opacityValues[z];

    updateStructures(theStructureId);
};
theMenu.append(opacityButton);
}
```

//-//

```
theMenu.append(ce("br"));
```

//-//

```
let customButton = ce("button");
customButton.onclick = function()
{
    customStructureOpacity();
};
customButton.className = "ourButton";
customButton.textContent = "Custom";
theMenu.append(customButton);
```

```
//-//
```

```
    theMenu.append(ce('br'));  
}
```

```
function customStructureOpacity()  
{
```

```
    let theOpacity = prompt("Enter Opacity as  
0.0 to 1.0")
```

```
    // opacity of structure
```

```
    theOpacityID.style.opacity = theOpacity;
```

```
    // fill in the textarea, to show the value of our  
choice
```

```
    ge("theOpacity").value = theOpacity;
```

```
    // update structuresArray to show the  
changed value
```

```
    updateStructures(worldOpacityId);
```

```
}
```

```
// Read the 3 Textarea, for the opacity value of  
our structure
```

```
// Set the opacity on our structure
```

```
function
customStructureOpacityFromTextInput()
{
    let theOpacity = ge("theOpacity").value;

    theOpacityID.style.opacity = theOpacity;

    // update structuresArray to show the
    changed value
    updateStructures(worldOpacityId);
}

//--//

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

// menuStructurePositionGet.js

let thePositionGetID;
```

```
function
menuStructurePositionGet(theStructureId)
{
    thePositionGetID = theStructureId;

    audioPlay("sfx_blip_001", 1.0);

    //-//

    if (ge("menuPositionStructureGet"))
    {

removeElement("menuPositionStructureGet");

    }

    //-//

    let theMenu = ce("div");
    theMenu.id = "menuPositionStructureGet";
    theMenu.className = "divStyle001";
    theMenu.style.position = "absolute";
    theMenu.style.left = ourPlayer.x + 100 + 'px';
    theMenu.style.top = ourPlayer.y - 55 + 'px';
    theMenu.style.width = 400 + 'px';
    theMenu.style.height = 192 + 'px';
```



```
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{
```

```
removeElement("menuPositionStructureGet");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
let positionLabel = ce("span");  
positionLabel.style.color = "rgb(255, 255,  
255)";  
positionLabel.textContent = "GET  
STRUCTURE POSITION";  
theMenu.append(positionLabel);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let getPosButton = ce("button");  
getPosButton.onclick = function()  
{  
    structurePositionGet();  
    keyboardControlOn();  
};  
getPosButton.className = "ourButton";  
getPosButton.textContent = "GET POS";  
theMenu.append(getPosButton);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let getPosRepeatButton = ce("button");  
getPosRepeatButton.onclick = function()  
{  
    structurePositionGetRepeat();  
    keyboardControlOn();  
};
```

```
};  
    getPosRepeatButton.className =  
"ourButton";  
    getPosRepeatButton.textContent = "GET  
POS REPEAT";  
    theMenu.append(getPosRepeatButton);  
  
//-//  
  
theMenu.append(ce("br"));  
  
//-//  
  
let cancelGetPosButton = ce("button");  
cancelGetPosButton.onclick = function()  
{  
  
removeTimer(currentStructurePositionTimer);  
};  
    cancelGetPosButton.className =  
"ourButton";  
    cancelGetPosButton.textContent = "Cancel  
Get Pos";  
    theMenu.append(cancelGetPosButton);  
  
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let xPosLabel = ce("span");  
xPosLabel.textContent = "X POS";  
theMenu.append(xPosLabel);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

```
//-//
```

```
let xPosTextarea = ce("textarea");  
xPosTextarea.id = "gotX";  
xPosTextarea.className = "textInput";  
xPosTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(xPosTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let yPosLabel = ce("span");  
yPosLabel.textContent = "Y POS";  
theMenu.append(yPosLabel);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

```
//-//
```

```
let yPosTextarea = ce("textarea");  
yPosTextarea.id = "gotY";  
yPosTextarea.className = "textInput";  
yPosTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(yPosTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
}

function structurePositionGet()
{
    let currentPosition =
    positionGet(thePositionGetID);

    thePositionGetID.innerHTML =
    currentPosition.x.toFixed(0) + "<br>" +
    currentPosition.y.toFixed(0) + "<br>";

    ge('gotX').value =
    currentPosition.x.toFixed(0);

    ge('gotY').value =
    currentPosition.y.toFixed(0);
}

let currentStructurePositionTimer;

function structurePositionGetRepeat()
{
    currentStructurePositionTimer =
    setInterval(function()
    {
        structurePositionGet();
```

```
    }, 1000);  
}
```

```
//--//
```

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

```
// menuStructurePositionSet.js
```

```
let thePositionSetID;
```

```
let worldPositionId;
```

```
let numberOnlyStructure;
```

```
function  
menuStructurePositionSet(theStructureId)  
{  
    thePositionSetID = ge(theStructureId);  
  
    worldPositionId = theStructureId;
```

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

// remove the text to find the digits only
// which number in the array, based on id
number - 1
// \d find digit
numberOnlyStructure =
worldPositionId.replace(/\D/g, "") - 1;

//-//

if (ge("menuPositionStructureSet"))
{
removeElement("menuPositionStructureSet");
}

//-//

let theMenu = ce("div");
theMenu.id = "menuPositionStructureSet";
theMenu.className = "divStyle001";
theMenu.style.position = "absolute";
theMenu.style.left = ourPlayer.x + 100 + 'px';
theMenu.style.top = ourPlayer.y - 80 + 'px';
```



```
theMenu.style.width = 400 + 'px';  
theMenu.style.height = 235 + 'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{
```

```
removeElement("menuPositionStructureSet");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let positionLabel = ce("span");
```

```
positionLabel.style.color = "rgb(255, 255,
255)";
positionLabel.textContent = "STRUCTURE
POSITION OPTIONS";
theMenu.append(positionLabel);

//--

theMenu.append(ce('hr'));

//--

let structureValues = [
    1, 2, 3, 4, 5, 6, 7, 8, 9, 10
];

for (let x = 0; x < structureValues.length; x++)
{
    let theButton = ce("button");
    theButton.onclick = function()
    {
        structureMoves(x);
    };
    theButton.className = "ourButton";
    theButton.textContent =
structureValues[x];
```

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

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let customPositionButton = ce("button");  
customPositionButton.onclick = function()  
{  
    customPositionStructure();  
};  
customPositionButton.className =  
"ourButton";  
customPositionButton.textContent =  
"Custom Structure Position";  
theMenu.append(customPositionButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let xPosLabel = ce("span");  
xPosLabel.textContent = "X POS";  
theMenu.append(xPosLabel);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

```
//-//
```

```
let xPosTextarea = ce("textarea");  
xPosTextarea.id = "customPosX";  
xPosTextarea.className = "textInput";  
xPosTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(xPosTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let yPosLabel = ce("span");
```

```
yPosLabel.textContent = "Y POS";  
theMenu.append(yPosLabel);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

```
//-//
```

```
let yPosTextarea = ce("textarea");  
yPosTextarea.id = "customPosY";  
yPosTextarea.className = "textInput";  
yPosTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(yPosTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let enterButton = ce("button");  
enterButton.onclick = function()
```

```
{
    customPositionStructureText();
    keyboardControlOn();
    enterButtonSound();
};
enterButton.className = "ourButton";
enterButton.textContent = "Enter";
theMenu.append(enterButton);
}
```

```
// Buttons move the structure
function structureMoves(theLocation)
{
    audioPlay("sfx_blip_001", 1.0);

    if (theLocation == '1')
    {
        thePositionSetID.x = 100;
        thePositionSetID.y = 100;
    }
    if (theLocation == '2')
    {
        thePositionSetID.x = 100;
        thePositionSetID.y = 200;
    }
    if (theLocation == '3')
```

```
{
    thePositionSetID.x = 100;
    thePositionSetID.y = 300;
}
if (theLocation == '4')
{
    thePositionSetID.x = 100;
    thePositionSetID.y = 400;
}
if (theLocation == '5')
{
    thePositionSetID.x = 100;
    thePositionSetID.y = 500;
}
if (theLocation == '6')
{
    thePositionSetID.x = 100;
    thePositionSetID.y = 600;
}
if (theLocation == '7')
{
    thePositionSetID.x = 100;
    thePositionSetID.y = 700;
}
if (theLocation == '8')
{
```

```
    thePositionSetID.x = 100;
    thePositionSetID.y = 800;
}
if (theLocation == '9')
{
    thePositionSetID.x = 100;
    thePositionSetID.y = 900;
}
if (theLocation == '0')
{
    thePositionSetID.x = 100;
    thePositionSetID.y = 1000;
}

// move the structure
thePositionSetID.style.left =
thePositionSetID.x + 'px';

thePositionSetID.style.top =
thePositionSetID.y + 'px';

// update the structuresArray posX and posY
structuresArray[numberOnlyStructure].posX
= thePositionSetID.x;
```



```
structuresArray[numberOnlyStructure].posY  
= thePositionSetID.y;
```

```
// update the innerHTML of the structure  
// show the structure id, name, posX, and  
posY  
thePositionSetID.innerHTML =  
structuresArray[numberOnlyStructure].id +  
"<br>" +  
structuresArray[numberOnlyStructure].name +  
"<br>" +  
structuresArray[numberOnlyStructure].posX +  
"<br>" +  
structuresArray[numberOnlyStructure].posY;
```

```
// update the textArea of the posX and posY  
ge("customPosX").value =  
structuresArray[numberOnlyStructure].posX;
```

```
ge("customPosY").value =  
structuresArray[numberOnlyStructure].posY;
```

```
updateStructures(worldPositionId);  
}
```

```
function customPositionStructure()
```

```
{  
    let playerX = prompt("Enter X Position");  
    let playerY = prompt("Enter Y Position");  
  
    // if no inputted value, keep player where  
they are  
    if (playerX == " || playerY == ")  
    {  
        thePositionSetID.x = thePositionSetID.x;  
        thePositionSetID.y = thePositionSetID.y;  
        return;  
    }  
  
    thePositionSetID.x = parseInt(playerX);  
  
    // y location of the structure  
    thePositionSetID.y = parseInt(playerY);  
  
    // if person did answer Y position, move  
structure  
    if (playerY != "")  
    {  
        // movePlayer();  
        thePositionSetID.style.left =  
thePositionSetID.x + 'px';  
    }
```

```
    thePositionSetID.style.top =  
thePositionSetID.y + 'px';  
}
```

```
// update the structuresArray posX and posY  
structuresArray[numberOnlyStructure].posX  
= thePositionSetID.x;
```

```
structuresArray[numberOnlyStructure].posY  
= thePositionSetID.y;
```

```
// update the innerHTML of the structure  
// show the structure id, name, posX, and  
posY
```

```
thePositionSetID.innerHTML =  
structuresArray[numberOnlyStructure].id +  
"<br>" +  
structuresArray[numberOnlyStructure].name +  
"<br>" +  
structuresArray[numberOnlyStructure].posX +  
"<br>" +  
structuresArray[numberOnlyStructure].posY;
```

```
// update the textArea of the posX and posY  
ge("customPosX").value =  
structuresArray[numberOnlyStructure].posX;
```

```
ge("customPosY").value =  
structuresArray[numberOnlyStructure].posY;
```

```
// update structuresArray to show the  
changed value  
updateStructures(worldPositionId);  
}
```

```
function customPositionStructureText()  
{  
    let playerX = ge("customPosX").value;  
    let playerY = ge("customPosY").value;  
  
    // if no inputted value, keep person where  
they are  
    if (playerY == "")  
    {  
        thePositionSetID.x = thePositionSetID.x;  
        thePositionSetID.y = thePositionSetID.y;  
        return;  
    }
```

```
thePositionSetID.x = parseInt(playerX);
```

```
// y location of the structure
```

```
thePositionSetID.y = parseInt(playerY);

// if person did answer Y position, move
structure
if (playerY != "")
{
    // movePlayer();
    thePositionSetID.style.left =
thePositionSetID.x + 'px';
    thePositionSetID.style.top =
thePositionSetID.y + 'px';
}

// update the structuresArray posX and posY
structuresArray[numberOnlyStructure].posX
= thePositionSetID.x;

structuresArray[numberOnlyStructure].posY
= thePositionSetID.y;

// update the innerHTML of the structure
// show the structure id, name, posX, and
posY
thePositionSetID.innerHTML =
structuresArray[numberOnlyStructure].id +
"<br>" +
```

```
structuresArray[numberOnlyStructure].name +
"<br>" +
structuresArray[numberOnlyStructure].posX +
"<br>" +
structuresArray[numberOnlyStructure].posY;

    // update the textArea of the posX and posY
    ge("customPosX").value =
structuresArray[numberOnlyStructure].posX;

    ge("customPosY").value =
structuresArray[numberOnlyStructure].posY;

    updateStructures(worldPositionId);
}

//--//

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

// menuStructureSizeGet.js
```

```
let theSizeGetID;
```

```
function menuStructureSizeGet(theStructureId)  
{  
    theSizeGetID = theStructureId;
```

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

```
    //-//
```

```
    if (ge("menuSizeStructureGet"))  
    {  
        removeElement("menuSizeStructureGet");  
    }
```

```
    //-//
```

```
    let theMenu = ce("div");  
    theMenu.id = "menuSizeStructureGet";  
    theMenu.className = "divStyle001";  
    theMenu.style.position = "absolute";  
    theMenu.style.left = ourPlayer.x + 100 + 'px';  
    theMenu.style.top = ourPlayer.y - 40 + 'px';  
    theMenu.style.width = 400 + 'px';  
    theMenu.style.height = 143 + 'px';
```

```
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");  
closeButton.onclick = function()  
{  
    removeElement("menuSizeStructureGet");  
    keyboardControlOn();  
};  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = "X";  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let sizeLabel = ce("span");  
sizeLabel.style.color = "rgb(255, 255, 255)";  
sizeLabel.textContent = "GET STRUCTURE  
SIZE";  
theMenu.append(sizeLabel);
```



```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let getSizeButton = ce("button");  
getSizeButton.onclick = function()  
{  
    structureSizeGet(theSizeGetID);  
};  
getSizeButton.className = "ourButton";  
getSizeButton.textContent = "GET SIZE";  
theMenu.append(getSizeButton);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let xSizeLabel = ce("span");  
xSizeLabel.textContent = "X Size";  
theMenu.append(xSizeLabel);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

```
//-//
```

```
let xSizeTextarea = ce("textarea");  
xSizeTextarea.id = "gotSizeX";  
xSizeTextarea.className = "textInput";  
xSizeTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(xSizeTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let ySizeLabel = ce("span");  
ySizeLabel.textContent = "Y Size";  
theMenu.append(ySizeLabel);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

```
//-//
```

```
let ySizeTextarea = ce("textarea");  
ySizeTextarea.id = "gotSizeY";  
ySizeTextarea.className = "textInput";  
ySizeTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(ySizeTextarea);  
}
```

```
function structureSizeGet()  
{  
    let currentSize = sizeGet(theSizeGetID);  
  
    ge('gotSizeX').value = currentSize.x;  
    ge('gotSizeY').value = currentSize.y;  
  
    theSizeGetID.innerHTML = currentSize.x +  
    "<br>" + currentSize.y + "<br>";  
}
```

//--//

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// menuStructureSizeSet.js

let theSizeSetID;

let worldSizeId;

**function menuStructureSizeSet(theStructureId)
{**

theSizeSetID = ge(theStructureId);

worldSizeId = theStructureId;

// Remove the text to find the digits only

// Which number in the array, based on id

number - 1

// \d find digit

```
numberOnlyStructure =  
worldSizeId.replace(/\D/g, "") - 1;
```

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

```
//-//
```

```
if (ge("menuSizeStructureSet"))  
{  
    removeElement("menuSizeStructureSet");  
}
```

```
//-//
```

```
let theMenu = ce("div");  
theMenu.id = "menuSizeStructureSet";  
theMenu.style.position = "absolute";  
theMenu.className = "divStyle001";  
theMenu.style.left = ourPlayer.x + 100 + 'px';  
theMenu.style.top = ourPlayer.y - 50 + 'px';  
theMenu.style.width = 400 + 'px';  
theMenu.style.height = 165 + 'px';  
ba(theMenu);
```

```
//-//
```

```
let closeButton = ce("button");
closeButton.onclick = function()
{
    removeElement("menuSizeStructureSet");
    keyboardControlOn();
};
closeButton.className = "ourButton";
closeButton.style.marginBottom = "-4px";
closeButton.textContent = "X";
theMenu.append(closeButton);

//-
```

```
theMenu.append(ce('hr'));
```

```
//-
```

```
let sizeLabel = ce("span");
sizeLabel.style.color = "rgb(255, 255, 255)";
sizeLabel.textContent = "SET STRUCTURE
SIZE";
theMenu.append(sizeLabel);
```

```
//-
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let defaultButton = ce("button");  
defaultButton.onclick = function()  
{  
    structureSizeSetDefault();  
};  
defaultButton.className = "ourButton";  
defaultButton.textContent = "Default";  
theMenu.append(defaultButton);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let xSizeLabel = ce("span");  
xSizeLabel.textContent = "X SIZE";  
theMenu.append(xSizeLabel);
```

```
//-//
```

```
theMenu.append(ce("nbsp"));
```

//-//

```
let xSizeTextarea = ce("textarea");  
xSizeTextarea.id = "customSizeX";  
xSizeTextarea.className = "textInput";  
xSizeTextarea.onkeydown = function()  
{  
    keyboardControlOff();  
};  
theMenu.append(xSizeTextarea);
```

//-//

```
theMenu.append(ce("br"));
```

//-//

```
let ySizeLabel = ce("span");  
ySizeLabel.textContent = "Y SIZE";  
theMenu.append(ySizeLabel);
```

//-//

```
theMenu.append(ce("nbsp"));
```

//-//


```
let ySizeTextarea = ce("textarea");
ySizeTextarea.id = "customSizeY";
ySizeTextarea.className = "textInput";
ySizeTextarea.onkeydown = function()
{
    keyboardControlOff();
};
theMenu.append(ySizeTextarea);
```

```
//-//
```

```
theMenu.append(ce("br"));
```

```
//-//
```

```
let enterButton = ce("button");
enterButton.onclick = function()
{
    structureSizeSet();
    keyboardControlOn();
    enterButtonSound();
};
enterButton.className = "ourButton";
enterButton.textContent = "Enter";
theMenu.append(enterButton);
```

```
}
```

```
function structureSizeSet()
```

```
{
```

```
    let theSizeX;
```

```
    let theSizeY;
```

```
    theSizeX = ge('customSizeX').value;
```

```
    theSizeY = ge('customSizeY').value;
```

```
    // theSizeSetID.innerHTML = theSizeX +  
    "<br>" + theSizeY + "<br>";
```

```
    // update the innerHTML of the structure  
    theSizeSetID.innerHTML =  
    structuresArray[numberOnlyStructure].id +  
    "<br>" +  
    structuresArray[numberOnlyStructure].name +  
    "<br>" + "Position" + "<br>" +  
    structuresArray[numberOnlyStructure].posX +  
    "<br>" +  
    structuresArray[numberOnlyStructure].posY +  
    "<br>" + "Size" + "<br>" + theSizeX + "<br>" +  
    theSizeY;
```

```
    theSizeSetID.style.width = theSizeX;
```

```
theSizeSetID.style.height = theSizeY;

updateStructures(worldSizeId);
}

function structureSizeSetDefault()
{
    let theSizeX = 114;
    let theSizeY = 70;

    ge('customSizeX').value = theSizeX;
    ge('customSizeY').value = theSizeY;

    // theSizeSetID.innerHTML = theSizeX +
    "<br>" + theSizeY + "<br>";

    // update the innerHTML of the structure
    theSizeSetID.innerHTML =
    structuresArray[numberOnlyStructure].id +
    "<br>" +
    structuresArray[numberOnlyStructure].name +
    "<br>" + "Position" + "<br>" +
    structuresArray[numberOnlyStructure].posX +
    "<br>" +
    structuresArray[numberOnlyStructure].posY +
```

```
"<br>" + "Size" + "<br>" + theSizeX + "<br>" +  
theSizeY;
```

```
theSizeSetID.style.width = theSizeX;  
theSizeSetID.style.height = theSizeY;
```

```
updateStructures(worldSizeId);  
}
```

```
//--//
```

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

```
// menuStructureTexture.js
```

```
function menuStructureTexture(theStructureId)  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
//-//
```

```
if (ge("menuTextureStructure"))  
{  
    removeElement("menuTextureStructure");  
}
```

//-//

```
let theMenu = ce("div");  
theMenu.id = "menuTextureStructure";  
theMenu.className = "divStyle001";  
theMenu.style.position = "absolute";  
theMenu.style.left = ourPlayer.x + 100 + 'px';  
theMenu.style.top = ourPlayer.y - 50 + 'px';  
theMenu.style.width = 400 + 'px';  
theMenu.style.height = 170 + 'px';  
ba(theMenu);
```

//-//

```
let closeButton = ce("button");  
closeButton.className = "ourButton";  
closeButton.style.marginBottom = "-4px";  
closeButton.textContent = " X ";  
closeButton.onclick = function()  
{  
    removeElement("menuTextureStructure");
```

```
};  
theMenu.append(closeButton);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
let textureOptionsLabel = ce("span");  
textureOptionsLabel.style.color = "rgb(255,  
255, 255)";  
textureOptionsLabel.textContent =  
"TEXTURE OPTIONS";  
theMenu.append(textureOptionsLabel);
```

```
//-//
```

```
theMenu.append(ce('hr'));
```

```
//-//
```

```
for (let z = 0; z < textureArray.length; z++)  
{  
    let textureButton = ce("button");  
    textureButton.onclick = function()
```

```
{
    if (online == true)
    {

ge(theStructureId).style.backgroundImage =
'url('+textureArray[z].textureOnline+');
    }
    else if (online == false)
    {

ge(theStructureId).style.backgroundImage =
'url('+textureArray[z].textureOffline+');
    }
    updateStructures(theStructureId);
};
textureButton.className = "ourButton";
textureButton.textContent =
textureArray[z].name;
theMenu.append(textureButton);

// - //

theMenu.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>

// mouseOff.js

function mouseOff()

{

// remove the listen for mouse clicks

**window.removeEventListener("click",
mousePos, false);**

}

//--//

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew

Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// mouseOn.js

```
function mouseOn()  
{  
    // add a listen for mouse clicks  
    window.addEventListener("click",  
mousePos, false);  
}
```

//--//

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

// mousePos.js

```
function mousePos(whichId)  
{  
    let mouseX = event.pageX;  
    let mouseY = event.pageY;
```

```
// let mousePos = "Mouse" + "<br>" + "X " +  
mouseX + "<br>" + "Y " + mouseY;
```

```
let mousePos = "X " + mouseX + "<br>" +  
"Y " + mouseY;
```

```
console.log(mousePos);
```

```
ge('infoDiv').innerHTML= mousePos;
```

```
if (cameraFlag == 0)  
{  
    ourPlayer.x = mouseX;  
    ourPlayer.y = mouseY;
```

```
    movePlayer(whichId);
```

```
    cameraFlag = 1;
```

```
    }  
}
```

```
//--//
```

```
// Dedicated to God the Father  
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// playerUpdate.js
```

```
function updatePlayer(idName)  
{  
    idName = 'thePlayer';  
  
    playerArray[0].name = playerArray[0].name;  
    playerArray[0].id = idName;  
    playerArray[0].posX = get(idName, 'pos').x;  
    playerArray[0].posY = get(idName, 'pos').y;  
    playerArray[0].sizeX = get(idName, 'size').x;  
    playerArray[0].sizeY = get(idName, 'size').y;  
    playerArray[0].colorFill = get(idName,  
'background-color'),  
    playerArray[0].colorText = get(idName,  
'color'),  
    playerArray[0].opacity = get(idName,  
'opacity');  
    playerArray[0].texture = get(idName,  
'background-image');  
  
    showPlayerData();  
}
```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// positionDistance.js
```

```
function getDistanceX(obj1, obj2)
```

```
{
```

```
    let obj1Position = positionGet(obj1).x;
```

```
    let obj2Position = positionGet(obj2).x
```

```
    let distanceX;
```

```
    if (obj1Position > obj2Position)
```

```
    {
```

```
        distanceX = obj1Position - obj2Position;
```

```
    }
```

```
    else if (obj1Position < obj2Position)
```

```
    {
```

```
    distanceX = obj2Position - obj1Position;
}

// console.log(distanceX);

ge('thePlayer').innerHTML = distanceX;

return distanceX;
}

function getDistanceY(obj1, obj2)
{
    let obj1Position = positionGet(obj1).y;

    let obj2Position = positionGet(obj2).y

    let distanceY;

    if (obj1Position > obj2Position)
    {
        distanceY = obj1Position - obj2Position;
    }
    else if (obj1Position < obj2Position)
    {
        distanceY = obj2Position - obj1Position;
    }
}
```

```
console.log(distanceY);

ge('thePlayer').innerHTML = distanceY;

return distanceY;
}

function getDistanceXY(obj1, obj2)
{
    let obj1PositionX = positionGet(obj1).x;
    let obj2PositionX = positionGet(obj2).x;

    let obj1PositionY = positionGet(obj1).y;
    let obj2PositionY = positionGet(obj2).y

    let distanceX;
    let distanceY;

    if (obj1PositionX > obj2PositionX)
    {
        distanceX = obj1PositionX -
obj2PositionX;
    }
    else if (obj1PositionX < obj2PositionX)
    {
```

```
    distanceX = obj2PositionX -  
obj1PositionX;  
}
```

```
if (obj1PositionY > obj2PositionY)  
{  
    distanceY = obj1PositionY - obj2PositionY;  
}  
else if (obj1PositionY < obj2PositionY)  
{  
    distanceY = obj2PositionY - obj1PositionY;  
}
```

```
ge('thePlayer').innerHTML =  
'<br><br><br><br>' + obj2 + '<br>' + "X " +  
distanceX + "<br>" + "Y " + distanceY;  
}
```

```
function keepGettingX(obj1, obj2)  
{  
    let ourTimer = setInterval(function()  
    {  
        getDistanceX(obj1, obj2);  
    }, 1000);  
}
```

```
function keepGettingY(obj1, obj2)
{
    let ourTimer = setInterval(function()
    {
        getDistanceY(obj1, obj2);
    }, 1000);
}
```

```
function keepGettingXY(obj1, obj2)
{
    let ourTimer = setInterval(function()
    {
        getDistanceXY(obj1, obj2);
    }, 1000);
}
```

```
let ourTimer;
```

```
function keepGettingXYCustom()
{
    let obj1 = "thePlayer";

    let obj2 = ge("idToFindDistance").value;

    // clear any previous searches
    if (ourTimer)
```



```
{
    clearInterval(ourTimer);
}

ourTimer = setInterval(function()
{
    getDistanceXY(obj1, obj2);
}, 1000);
}

function removeTimer(whichTimer)
{
    clearInterval(whichTimer);

    ge(activePlayer).innerHTML = "";
}

// let ourTimer = setInterval(function()
// { getDistanceY('thePlayer', 'STRUCTURE1');},
// 1000);

// examples
// getDistanceX('thePlayer', 'target1');
// getDistanceY('thePlayer', 'target1');

//--//
```

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

```
// positionGet.js
```

```
// get the Position of any Element that has
specified id
```

```
function positionGet(whichId)
{
    let playerElement = ge(whichId);

    if (playerElement)
    {
        let theRect =
playerElement.getBoundingClientRect();

        let posXY = {
            x: theRect.x + theRect.width / 2 +
window.scrollX,
```

```
        y: theRect.top + theRect.height / 2 +  
window.scrollY  
    };
```

```
        return posXY;  
    }  
}
```

```
//--//
```

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

```
// positionGetSizeAndSides.js
```

```
// x, y, width, height, top, right, bottom, left  
function positionGetSizeAndSides(whichId)  
{  
    whichId = ge(whichId);
```

```
    let theRect =  
whichId.getBoundingClientRect();
```

```
    return theRect;  
}
```

// How to Use this Function

```
//  
console.log(JSON.stringify(positionGetSizeAndSides('boxOne')));
```

```
//  
console.log(JSON.stringify(positionGetSizeAndSides('thePlayer')));
```

```
//  
console.log(JSON.stringify(positionGetSizeAndSides(get(activePlayer))));
```

```
/*
```

Here is an example of using this function

```
//  
console.log(JSON.stringify(positionGetSizeAndSides('thePlayer')));
```

Returns the x position, y position, width, height, of the object. It also returns position of the the top, bottom, left and right sides, as shown below.

```
{ "x":374,"y":250.33334350585938,"width":78,"height":78,"top":250.33334350585938,"right":452,"bottom":328.3333435058594,"left":374 }
```

```
*/
```

// The x represents the left top side of the object.

```
/*
```

```
{  
    "x":374,  
    "y":250.33334350585938,  
    "width":78,  
    "height":78,  
    "top":250.33334350585938,  
    "right":452,  
    "bottom":328.3333435058594,  
    "left":374  
}
```

```
*/
```

```
/*
```

**Notice that the x and left are the same.
They both return 374, in our example.**

**Notice the y and top are the same.
They both return 250.33334350585938.**

```
*/
```

```
// example
```

```
//
```

```
console.log(JSON.stringify(positionGetSizeAn  
dSides('thePlayer')));
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// sizeGet.js
```

```
// gets size of any element of a specified id
```

```
function sizeGet(whichId)  
{  
    whichId = ge(whichId);  
  
    let theRect =  
whichId.getBoundingClientRect();  
  
    let sizeXY =  
    {  
        x: parseFloat(theRect.width),  
        y: parseFloat(theRect.height)  
    };  
  
    return sizeXY;  
}
```

```
/*
```

How to Use this Function

```
// get the size of the player object
```

```
console.log(JSON.stringify(sizeGet('thePlayer')));
```

```
// Returns x and y size values, which includes padding  
{"x":78,"y":78}
```

```
// Formatted to show x and y clearer  
{  
  "x":78,  
  "y":78  
}
```

```
/*
```

Important to note is that padding effects size. So the object that is the size x of 70, becomes 78, if the padding of left and right are set to 4.

```
*/
```

```
// example
```

```
//  
console.log(JSON.stringify(sizeGet('thePlayer')));
```


//--//

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// travelTo.js

// Send Player to New Location

**// based on Choice from menuPlayer GoTo
section**

function travelTo(theSection)

```
{  
    if (theSection == "news")  
    {  
        if (!ge("newsSection"))  
        {  
            alert("First, Explore the World\nto  
REVEAL the NEWS Section");  
        }  
    }  
    else
```

```
{
    audioPlay("sfx_blip_001", 1.0);

    ourPlayer.x = 100;

    ourPlayer.y =
positionGet("newsSection").y;
}
}

if (theSection == "scripting")
{
    if (!ge("scriptingSection"))
    {
        alert("First, Explore the World\nto
REVEAL the SCRIPTING Section");
    }
    else
    {
        audioPlay("sfx_blip_001", 1.0);

        ourPlayer.x = 100;

        ourPlayer.y =
positionGet("scriptingSection").y;
    }
}
```

```
}

if (theSection == "books")
{
    if (!ge("collegeBooksSection"))
    {
        alert("First, Explore the World\nto  
REVEAL the COLLEGE BOOKS Section");
    }
    else
    {
        audioPlay("sfx_blip_001", 1.0);

        ourPlayer.x = 100;

        ourPlayer.y =
positionGet("collegeBooksSection").y;
    }
}

if (theSection == "evidence")
{
    if (!ge("evidenceldSection"))
    {
        alert("First, Explore the World\nto  
REVEAL the EVIDENCE Section");
    }
}
```

```
    }  
    else  
    {  
        audioPlay("sfx_blip_001", 1.0);  
  
        ourPlayer.x = 100;  
  
        ourPlayer.y =  
positionGet("evidenceldSection").y;  
    }  
}  
movePlayer(activePlayer);  
}
```

//--//

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

// rowOfBookImages.js

```

function rowOfBookImages(theX, theY, sizeX,
sizeY, whichArray, criteria1, theHeading)
{
    let counter = 0;

    let mainDiv = ce("div");
    mainDiv.id = "displayBooksId";
    mainDiv.className = "bookImageStyle";
    mainDiv.title = "rowOfBookImages(theX,
theY, sizeX, sizeY, whichArray, criteria1,
theHeading)";
    mainDiv.style.position = "absolute";
    mainDiv.style.left = theX;
    mainDiv.style.top = theY;
    mainDiv.style.display = "flex";
    mainDiv.style.flexDirection = "row";
    mainDiv.style.gap = 75 + "px";
    ba(mainDiv);

    //-//

    // used with papercliptronics,
    javascriptBook1, javascriptBook2, logicA,
    logicB, logicC, simulation, science, human,
    mathematics arrays
    for (let x = 0; x < whichArray.length; x++)

```

```
{  
    counter += 1;  
  
    if (whichArray[x].tag == criteria1)  
    {  
        // filter by month  
        //if (whichArray[x].month === '12')  
  
        // filter by multiple months  
        //if (whichArray[x].month === '8' ||  
whichArray[x].month === '9')  
  
        // filter by date  
        // if (whichArray[x].month === '2' &&  
whichArray[x].day === '29' &&  
whichArray[x].year === '2018')  
  
        // filter by multiple dates  
        //if ((whichArray[x].month === '2' &&  
whichArray[x].day === '29' &&  
whichArray[x].year === '2018') ||  
(whichArray[x].month === '2' &&  
whichArray[x].day === '29' &&  
whichArray[x].year === '2020'))  
  
        // filter by day, of any month, of any year
```

```
// if (whichArray[x].day === '2')
```

```
// filter by year
```

```
// if (whichArray[x].year === '2022')
```

```
// filter by multiple years
```

```
// if (whichArray[x].year === '2019' ||  
whichArray[x].year === '2020')
```

```
// filter by month and year
```

```
// if (whichArray[x].month === '2' &&  
whichArray[x].year === '2020')
```

**// we can replace the if statement below,
with one of the options above.**

```
if (whichArray) // filter by all  
{
```

```
    let subDiv = ce("div");
```

```
    subDiv.id = theHeading + counter;
```

```
    subDiv.style.width = sizeX + 'px';
```

```
    subDiv.style.borderStyle = "solid";
```

```
    subDiv.style.borderWidth = 1 + "px";
```

```
    subDiv.style.borderColor = "rgb(0,  
255, 255)";
```

```
    mainDiv.append(subDiv);
```

```
//-//
```

```
    let printButton = ce("button");  
    printButton.className =  
"ourButton";  
    printButton.textContent = "print";  
    printButton.onclick = function()  
    {  
        printOnlyDiv(subDiv.id);  
    };  
    subDiv.append(printButton);
```

```
//-//
```

```
    let theTexture = ce("img");  
    theTexture.id = theHeading + counter;  
    theTexture.src =  
whichArray[x].image;  
    theTexture.className =  
"bookImage";  
    subDiv.append(theTexture);  
    }  
    }  
    }  
}
```



```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// rowOfCollegeBooks.js
```

```
function rowOfCollegeBooks(theX, theY, sizeX,  
sizeY, whichArray)
```

```
{
```

```
    let counter = 0;
```

```
//--//
```

```
    let mainDiv = ce("div");
```

```
    mainDiv.id = "collegeBooksSection";
```

```
    mainDiv.className = "";
```

```
    mainDiv.title = "rowOfCollegeBooks(theX,  
theY, sizeX, sizeY, whichArray)";
```

```
    mainDiv.style.position = "absolute";
```

```
    mainDiv.style.left = theX;
```

```
    mainDiv.style.top = theY;
```

```
mainDiv.style.display = "flex";  
mainDiv.style.flexDirection = "row";  
mainDiv.style.gap = 75 + "px";  
mainDiv.style.overflow = "hidden";  
mainDiv.style.overflowY = "scroll";  
ba(mainDiv);
```

```
//-//
```

```
// used with collegeBooks Array  
for (let x = 0; x < whichArray.length; x++)  
{  
    counter += 1;
```

```
    let subDiv = ce("div");  
    subDiv.className = "divBackground";  
    subDiv.id = "CollegeBooks" + counter;  
    subDiv.style.width = sizeX + 'px';  
    subDiv.style.height = sizeY + 'px';  
    subDiv.style.borderStyle = "solid";  
    subDiv.style.borderWidth = "2px";  
    subDiv.style.borderColor = "rgba(100, 100,  
255, 1.0)";  
    subDiv.style.zIndex = "1";  
    subDiv.style.fontFamily = "arial";  
    subDiv.style.fontWeight = "normal";
```

```
subDiv.style.fontSize = "medium";  
subDiv.style.textAlign = "center";  
subDiv.style.color = "black";
```

```
mainDiv.append(subDiv);
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className = "ourButton";  
printButton.textContent = "print";  
printButton.onclick = function()  
{  
    printOnlyDiv(subDiv.id);  
};  
subDiv.append(printButton);
```

```
//-//
```

```
let theTitle = ce("div");  
theTitle.style.color = "rgb(255, 255, 255)";  
theTitle.innerHTML = "College Books";  
theTitle.innerHTML +=  
whichArray[x].name;  
subDiv.append(theTitle);
```

```
//-//
```

```
let theBreakBelowTitle = ce("hr");  
theBreakBelowTitle.className =  
"hrCompact";  
subDiv.append(theBreakBelowTitle);
```

```
//-//
```

```
let theLink = ce("a");  
theLink.target = "_blank";  
theLink.href = whichArray[x].url;  
theLink.innerHTML = "";  
subDiv.append(theLink);
```

```
//-//
```

```
let theTexture = ce("img");  
theTexture.className = "theImage";  
theTexture.src = whichArray[x].image;  
theLink.append(theTexture);  
}  
}
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// rowOfCollegeLinks.js
```

```
function rowOfCollegeLinks(theX, theY,  
whichArray, sizeX)  
{  
    let counter = 0;  
  
    let mainDiv = ce("div");  
    mainDiv.id = "collegeLinksId";  
    mainDiv.title = "rowOfCollegeLinks(theX,  
theY, whichArray)";  
    mainDiv.style.position = "absolute";  
    mainDiv.style.left = theX + "px";  
    mainDiv.style.top = theY + "px";  
    mainDiv.style.width = '200px';  
    mainDiv.style.display = "flex";  
    mainDiv.style.flexDirection = "row";  
    mainDiv.style.gap = "75px";  
    ba(mainDiv);
```

//-//

```
for (let x = 0; x < whichArray.length; x++) {  
    counter += 1;
```

```
    let subDiv = ce("div");  
    subDiv.id = "CollegeLinks" + counter;  
    subDiv.className = "divBackground";  
    subDiv.style.padding = "10px";  
    subDiv.style.width = sizeX + 'px';  
    subDiv.style.zIndex = 1;  
    subDiv.style.borderStyle = "solid";  
    subDiv.style.borderWidth = "2px";  
    subDiv.style.borderColor = "rgba(100, 100,  
255, 1.0)";  
    subDiv.style.fontFamily = "arial";  
    subDiv.style.fontWeight = "bold";  
    subDiv.style.fontSize = "medium";  
    subDiv.style.color = "aqua";  
    subDiv.style.textAlign = "center";  
    mainDiv.append(subDiv);
```

//-//

```
    let theLink = ce("a");  
    theLink.href = whichArray[x].url;
```

```
    theLink.target = "_blank";  
    theLink.innerHTML = whichArray[x].name;  
    subDiv.append(theLink);  
  }  
}
```

//--//

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

// rowOfDetails.js

**function rowOfDetails(theX, theY, sizeX, sizeY,
whichArray, criteria1)**

```
{  
  let counter = 0;
```

//-//

```
let mainDiv = ce("div");  
mainDiv.id = "detailsId";
```

```
mainDiv.title = "rowOfDetails(theX, theY,  
sizeX, sizeY, whichArray, criteria1)";  
mainDiv.style.position = "absolute";  
mainDiv.style.left = theX + "px";  
mainDiv.style.top = theY + "px";  
mainDiv.style.display = "flex";  
mainDiv.style.flexDirection = "row";  
mainDiv.style.gap = "75px";  
ba(mainDiv);
```

```
//-//
```

```
for (let i = 0; i < whichArray.length; i++) {  
    counter += 1;  
  
    if (whichArray[i].tag == criteria1)  
    {  
        // filter by month  
        if (whichArray[i].date.substring(0, 2) ==  
"07")  
        {  
            if (whichArray[i].probability >= 50)  
            {  
                let subDiv = ce("div");  
                subDiv.className = "divStyle001";  
                mainDiv.append(subDiv);  
            }  
        }  
    }  
}
```



```
//-//
```

```
let theDiv = ce("details");  
theDiv.id = "EvidenceDetails" +  
counter;  
theDiv.style.width = sizeX + 'px';  
theDiv.style.height = sizeY + 'px';  
theDiv.style.color = "rgb(255, 255,  
255)";  
subDiv.append(theDiv);
```

```
//-//
```

```
theDiv.innerText = "Content  
Sensitive - Open at Your Discretion";
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className =  
"ourButton";
```

```
printButton.textContent = "print";  
printButton.onclick = function()  
{  
    printOnlyDiv(theDiv.id);  
};
```

```
subDiv.append(printButton);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let titleText = whichArray[i].title;  
theDiv.append(titleText);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let dateText = whichArray[i].date;  
theDiv.append(dateText);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let infoText = whichArray[i].info;  
theDiv.append(infoText);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let probabilityText = "Probability  
Rating: " + whichArray[i].probability + "%";  
theDiv.append(probabilityText);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let link = ce("a");
```

```
link.href = whichArray[i].url;  
link.target = "_blank";  
link.innerText = "Link";  
theDiv.append(link);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let image = ce("img");  
image.src = whichArray[i].image;  
image.className = "theImage";  
theDiv.append(image);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
//--//
```

```
// filter by MONTH
```

```
// if (whichArray[i].date.substring(0, 2) == 7)
```

// filter by Multiple MONTHS

**// if(whichArray[i].date.substring(0, 2) == 8 ||
whichArray[i].date.substring(0, 2) == 9)**

// filter by DATE

**// if (whichArray[i].date.substring(0, 10) ==
"02/29/2018")**

// filter by Multiple DATES

**// if (whichArray[i].date.substring(0, 10) ==
"02/29/2018" || whichArray[i].date.substring(0,
10) == "02/29/2020")**

// filter by DAY, of any month, of any year

// if (whichArray[i].date.substring(3, 5) == 2)

// filter by YEAR

**// if (whichArray[i].date.substring(6, 10) ==
2019)**

// filter by MULTIPLE YEARS

**// if (whichArray[i].date.substring(6, 10) == 2019
|| whichArray[i].date.substring(6, 10) == 2020)**

// filter by MONTH and YEAR

```
// if (whichArray[i].date.substring(0, 2) == 2 &&  
whichArray[i].date.substring(6, 10) == 2020)
```

```
//--//
```

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

```
function rowOfEvidence(theX, theY, sizeX,  
sizeY, whichArray, criteria1)  
{  
    let counter = 0;
```

```
//--//
```

```
    let mainDiv = ce("div");  
    mainDiv.id = "evidenceldSection";  
    mainDiv.title = "rowOfEvidence(theX, theY,  
sizeX, sizeY, whichArray, criteria1)";  
    mainDiv.style.position = "absolute";  
    mainDiv.style.left = theX + "px";
```

```
mainDiv.style.top = theY + "px";  
mainDiv.style.display = "flex";  
mainDiv.style.flexDirection = "row";  
mainDiv.style.gap = "75px";  
ba(mainDiv);
```

```
//-//
```

```
// used with evidence array  
for (let i = 0; i < whichArray.length; i++) {  
    if (whichArray[i].tag == criteria1)  
    {  
        // filter by MONTH  
        // if (whichArray[i].date.substring(0, 2)  
== 3)  
  
        // filter by Multiple MONTHS  
        // if(whichArray[i].date.substring(0, 2) ==  
8 || whichArray[i].date.substring(0, 2) == 9)  
  
        // filter by DATE  
        // if (whichArray[i].date.substring(0, 10)  
== "02/29/2018")  
  
        // filter by Multiple DATES
```

```
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018" ||  
whichArray[i].date.substring(0, 10) ==  
"02/29/2020")
```

```
// filter by DAY of month  
// if (whichArray[i].date.substring(3, 5)  
== 2)
```

```
// filter by YEAR  
// if (whichArray[i].date.substring(6, 10)  
== 2019)
```

```
// filter by MULTIPLE YEARS  
// if (whichArray[i].date.substring(6, 10)  
== 2019 || whichArray[i].date.substring(6, 10)  
== 2020)
```

```
// filter by MONTH and YEAR  
// if (whichArray[i].date.substring(0, 2)  
== 2 && whichArray[i].date.substring(6, 10) ==  
2020)
```

```
// or show all by using  
// if (whichArray)
```



```
7) // filter by month
    if (whichArray[i].date.substring(0, 2) ==

    {
        counter += 1;

        //-//

        let subDiv = ce("div");
        subDiv.id = "Evidence" + counter;
        subDiv.className = "divStyle001";
        subDiv.style.width = sizeX + 'px';
        mainDiv.append(subDiv);

        //-//

        let hrCompact = ce("hr");
        hrCompact.className = 'hrCompact';
        subDiv.append(hrCompact);

        //-//

        let printButton = ce("button");
        printButton.className =
"ourButton";
        printButton.innerText = "Print";
```

```
printButton.onclick = function()
{
    printOnlyDiv(subDiv.id);
};
subDiv.append(printButton);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let title = ce("span");
title.innerText = whichArray[i].title + "\n";
subDiv.append(title);
```

```
//-//
```

```
let date = ce("span");
date.innerText = whichArray[i].date + "\n";
subDiv.append(date);
```

```
//-//
```

```
let info = ce("span");
info.innerText = whichArray[i].info + "\n";
subDiv.append(info);

//-
```

```
let link = ce("a");
link.href = whichArray[i].url;
link.target = "_blank";
link.innerText = "Link";
subDiv.append(link);
}
}
}
}

//--
```

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

// rowOfEvidenceWithImage.js

```
function rowOfEvidenceWithImage(theX, theY,
sizeX, sizeY, whichArray, criteria1, theHeading)
{
    let counter = 0;

    let mainDiv = ce("div");
    mainDiv.id = "evidenceWithImageId";
    mainDiv.title =
"rowOfEvidenceWithImage(theX, theY, sizeX,
sizeY, whichArray, criteria1, theHeading)";
    mainDiv.style.position = "absolute";
    mainDiv.style.left = theX + "px";
    mainDiv.style.top = theY + "px";
    mainDiv.style.display = "flex";
    mainDiv.style.flexDirection = "row";
    mainDiv.style.gap = "75px";
    ba(mainDiv);
```

//-//

```
// used with evidence array
for (let i = 0; i < whichArray.length; i++) {
    if (whichArray[i].tag == criteria1)
    {
```

```
// filter by MONTH  
// if (whichArray[i].date.substring(0, 2)  
== 3)
```

```
// filter by Multiple MONTHS  
// if(whichArray[i].date.substring(0, 2) ==  
8 || whichArray[i].date.substring(0, 2) == 9)
```

```
// filter by DATE  
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018")
```

```
// filter by Multiple DATES  
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018" ||  
whichArray[i].date.substring(0, 10) ==  
"02/29/2020")
```

```
// filter by DAY of month  
// if (whichArray[i].date.substring(3, 5)  
== 2)
```

```
// filter by YEAR  
// if (whichArray[i].date.substring(6, 10)  
== 2019)
```

```
// filter by MULTIPLE YEARS
// if (whichArray[i].date.substring(6, 10)
== 2019 || whichArray[i].date.substring(6, 10)
== 2020)
```

```
// filter by MONTH and YEAR
// if (whichArray[i].date.substring(0, 2)
== 2 && whichArray[i].date.substring(6, 10) ==
2020)
```

```
// or show all by using
// if (whichArray)
```

```
if (whichArray)
{
    counter += 1;
```

```
//-//
```

```
let subDiv = ce("div");
subDiv.id = "EvidenceWithImage" +
counter;
subDiv.className = "divStyle001";
subDiv.style.width = sizeX + 'px';
subDiv.style.height = sizeY + 'px';
mainDiv.append(subDiv);
```

```
//-//
```

```
    let heading = ce("div");  
    heading.innerHTML = theHeading +  
"<hr class='hrCompact'>";  
    subDiv.append(heading);
```

```
//-//
```

```
    let printButton = ce("button");  
    printButton.className =  
"ourButton";  
    printButton.innerText = "print";  
    printButton.onclick = function()  
    {  
        printOnlyDiv(subDiv.id);  
    };  
    subDiv.append(printButton);
```

```
//-//
```

```
    subDiv.append(ce("br"));
```

```
//-//
```

```
let title = ce("b");  
title.innerHTML = whichArray[i].title +  
"<br>";  
subDiv.append(title);  
  
//-//
```

```
let date = ce("div");  
date.innerHTML = whichArray[i].date  
+ "<br>";  
subDiv.append(date);  
  
//-//
```

```
let info = ce("div");  
info.innerHTML = whichArray[i].info +  
"<br>";  
subDiv.append(info);  
  
//-//
```

```
let link = ce("a");  
link.href = whichArray[i].url;  
link.target = "_blank";  
link.innerHTML = "Link";  
subDiv.append(link);
```



```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let image = ce("img");  
image.src = whichArray[i].image;  
image.className = "theImage";  
subDiv.append(image);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// rowOfInvestigation.js
```

```
function sortInvestigation(whichY,  
whichDirection)  
{  
    let whichArray =  
petitoGabrielleInvestigation;  
  
    if (whichDirection == "up")  
    {  
        whichArray.sort(function(a, b)  
        {  
            return new Date(a.date) - new  
Date(b.date);  
        });  
    }  
    else if (whichDirection == "down")  
    {  
        whichArray.sort(function(b, a)  
        {  
            return new Date(a.date) - new  
Date(b.date);  
        });  
    }  
  
    rowOfInvestigation(100, whichY, 400, 400,  
whichArray, "note");
```

```
}  
  
function removeRow()  
{  
    let name = "Investigation";  
  
    let counter = 0;  
  
    for (x = 0; x <  
petitoGabrielleInvestigation.length; x++)  
    {  
        counter += 1;  
  
        removeElement(name + counter);  
    }  
  
    removeElement("theIdOfButtons");  
}
```

```
function rowOfInvestigation(theX, theY, sizeX,  
sizeY, whichArray, criteria1)  
{  
    let counter = 0;  
  
    let mainDiv = ce("div");  
    mainDiv.id = "evidenceWithImageId";
```

```
mainDiv.title = "rowOfInvestigation(theX,  
theY, sizeX, sizeY, whichArray, criteria1)";  
mainDiv.style.position = "absolute";  
mainDiv.style.left = theX + "px";  
mainDiv.style.top = theY + "px";  
mainDiv.style.display = "flex";  
mainDiv.style.flexDirection = "row";  
mainDiv.style.gap = "75px";  
ba(mainDiv);
```

//-//

```
let theButtons = ce("div");  
theButtons.id = "theIdOfButtons";  
theButtons.className = "divStyle001";  
mainDiv.append(theButtons);
```

//-//

```
let buttonAscending = ce("button");  
buttonAscending.className = "ourButton";  
buttonAscending.onclick = function()  
{  
    sortInvestigation(theY, 'up'); removeRow();  
};  
buttonAscending.innerHTML = "Ascending";
```

```
theButtons.append(buttonAscending);
```

```
//-//
```

```
let buttonDescending = ce("button");  
buttonDescending.className =  
"ourButton";  
buttonDescending.onclick = function()  
{  
    sortInvestigation(theY, 'down');  
removeRow();  
};  
buttonDescending.innerHTML =  
"Descending";  
theButtons.append(buttonDescending);
```

```
//-//
```

```
let buttonRemoveRow = ce("button");  
buttonRemoveRow.className =  
"ourButton";  
buttonRemoveRow.onclick = function()  
{  
    removeRow();  
};
```

```
buttonRemoveRow.innerHTML = "Remove \n
Row";
theButtons.append(buttonRemoveRow);

//--//

for (let i = 0; i < whichArray.length; i++) {
    if (whichArray[i].tag == criteria1)
    {
        // filter by MONTH
        // if (whichArray[i].date.substring(0, 2)
== 7)

        // filter by Multiple MONTHS
        // if(whichArray[i].date.substring(0, 2) ==
8 || whichArray[i].date.substring(0, 2) == 9)

        // filter by DATE
        // if (whichArray[i].date.substring(0, 10)
== "02/29/2018")

        // filter by Multiple DATES
        // if (whichArray[i].date.substring(0, 10)
== "02/29/2018" ||
whichArray[i].date.substring(0, 10) ==
"02/29/2020")
```

```
// filter by DAY, of any month, of any  
year
```

```
// if (whichArray[i].date.substring(3, 5)  
== 2)
```

```
// filter by YEAR  
// if (whichArray[i].date.substring(6, 10)  
== 2019)
```

```
// filter by MULTIPLE YEARS  
// if (whichArray[i].date.substring(6, 10)  
== 2019 || whichArray[i].date.substring(6, 10)  
== 2020)
```

```
// filter by MONTH and YEAR  
// if (whichArray[i].date.substring(0, 2)  
== 2 && whichArray[i].date.substring(6, 10) ==  
2020)
```

```
if (whichArray) // shows all items  
{  
    counter += 1;
```

```
let theDiv = ce("div");  
theDiv.id = "Investigation" + counter;
```

```
theDiv.className = "divStyle001";  
theDiv.style.width = sizeX + 'px';  
theDiv.style.height = sizeY + 'px';  
mainDiv.append(theDiv);
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className =  
"ourButton";  
printButton.style.marginRight =  
"10px";  
printButton.style.zIndex = "20";  
printButton.onclick = function()  
{  
    printOnlyDiv(theDiv.id);  
};  
printButton.textContent = "print";  
theDiv.append(printButton);
```

```
//-//
```

```
let dateSpan = ce("span");  
dateSpan.innerHTML =  
whichArray[i].date;  
theDiv.append(dateSpan);
```



```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let summarySpan = ce("span");  
summarySpan.style.color = "rgb(0,  
255, 255)";  
summarySpan.style.fontWeight =  
"bold";  
summarySpan.innerHTML =  
whichArray[i].summary;  
theDiv.append(summarySpan);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let noteSpan = ce("span");  
noteSpan.innerHTML =  
whichArray[i].note;  
theDiv.append(noteSpan);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let newsArticleLink = ce("a");  
newsArticleLink.href =  
whichArray[i].newsArticleOne;  
newsArticleLink.target = "_blank";  
newsArticleLink.innerHTML =  
whichArray[i].newsArticleOne;  
newsArticleLink.style.color = "aqua";  
theDiv.append(newsArticleLink);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
}
```

```
}
```

```
}
```

```
}
```

```
//--//
```

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

```
// rowOfLanguage.js
```

```
function rowOfLanguage(theX, theY, sizeX,
sizeY, whichArray, criteria1, criteria2)
```

```
{
    let counter = 0;
```

```
//-//
```

```
    let mainDiv = ce("div");
    mainDiv.id = "scriptingCategory";
    mainDiv.title = "rowOfLanguage(theX, theY,
sizeX, sizeY, whichArray, criteria1, criteria2)";
    mainDiv.style.position = "absolute";
    mainDiv.style.left = theX + "px";
    mainDiv.style.top = theY + "px";
    mainDiv.style.display = "flex";
    mainDiv.style.flexDirection = "row";
    mainDiv.style.gap = 75 + "px";
```

```
ba(mainDiv);
```

```
//-//
```

```
// used with lindenLanguage array
```

```
// used with javascriptLanguage array
```

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

```
{
```

```
    if (whichArray[i].category == criteria1 &&  
whichArray[i].tag == criteria2)
```

```
    {
```

```
        counter += 1;
```

```
//-//
```

```
    let subDiv = ce("div");
```

```
    subDiv.id = "JavaScriptCategories_" +  
whichArray[i].category + counter;
```

```
    subDiv.className = "divStyle001";
```

```
    subDiv.style.width = sizeX + 'px';
```

```
    subDiv.style.height = sizeY + 'px';
```

```
    mainDiv.append(subDiv);
```

```
//-//
```

```
let controlPanelDetails = ce("details");
```

```
subDiv.append(controlPanelDetails);
```

```
//-//
```

```
let controlPanelSummary =  
ce("summary");
```

```
controlPanelDetails.append(controlPanelSum  
mary);
```

```
//-//
```

```
let controlPanel = ce("div");  
controlPanel.style.position = "relative";
```

```
controlPanelSummary.append(controlPanel);
```

```
//-//
```

```
subDiv.append(ce('hr'));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className = "ourButton";
```

```
printButton.style.marginRight = 10 +  
"px";  
printButton.onclick = function()  
{  
    printOnlyDiv(subDiv.id);  
};  
printButton.textContent = "print";  
  
controlPanelDetails.append(printButton);  
  
//-//  
  
let titleOfUrlAnchor = ce("a");  
titleOfUrlAnchor.target = "_blank";  
titleOfUrlAnchor.href =  
whichArray[i].url;  
titleOfUrlAnchor.style.marginRight = 10  
+ "px";  
  
controlPanelDetails.append(titleOfUrlAnchor);  
  
//-//  
  
let titleOfUrl = ce("span");  
titleOfUrl.innerHTML = "Link 1";  
titleOfUrlAnchor.append(titleOfUrl);
```

```
//-//
```

```
let titleOfUrlAnchor2 = ce("a");  
titleOfUrlAnchor2.target = "_blank";  
titleOfUrlAnchor2.href =  
whichArray[i].url2;  
  
controlPanelDetails.append(titleOfUrlAnchor2);
```

```
//-//
```

```
let titleOfUrl2 = ce("span");  
titleOfUrl2.innerHTML = "Link 2";  
titleOfUrlAnchor2.append(titleOfUrl2);
```

```
//-//
```

```
subDiv.append(ce('hr'));
```

```
//-//
```

```
let titleOfLanguage = ce("span");  
titleOfLanguage.style.color = "rgb(180,  
180, 180)";
```

```
titleOfLanguage.innerHTML =  
whichArray[i].language;  
subDiv.append(titleOfLanguage);  
  
//-//  
  
subDiv.append(ce('hr'));  
  
//-//  
  
let titleOfCategory = ce("span");  
titleOfCategory.style.color = "rgb(255, 0,  
255)";  
titleOfCategory.innerHTML =  
whichArray[i].category;  
subDiv.append(titleOfCategory);  
  
//-//  
  
subDiv.append(ce('hr'));  
  
//-//  
  
let titleOfFunction = ce("div");  
titleOfFunction.style.fontSize = 30 +  
"px";
```



```
titleOfFunction.style.color = "rgb(0, 255,
255)";
titleOfFunction.innerHTML =
whichArray[i].name;
subDiv.append(titleOfFunction);

//-//

subDiv.append(ce('hr'));

//-//

let titleOfDescription= ce("span");
titleOfDescription.style.color = "rgb(0,
190, 220)";
titleOfDescription.innerHTML =
whichArray[i].description;
subDiv.append(titleOfDescription);

//-//

subDiv.append(ce('hr'));

//-//

let syntax = ce("span");
```

```
        syntax.style.color = "rgb(180, 180,  
180)";  
        syntax.innerHTML =  
whichArray[i].syntax;  
        subDiv.append(syntax);  
  
        //-//  
  
        subDiv.append(ce('hr'));  
  
        //-//  
  
        let useage = ce("span");  
        useage.innerHTML =  
whichArray[i].useage;  
        subDiv.append(useage);  
  
        //-//  
  
        subDiv.append(ce('hr'));  
  
        //-//  
  
        let returnValue = ce("span");  
        returnValue.style.color = "rgb(150, 100,  
200)";
```

```
        returnValue.innerHTML =  
whichArray[i].type;  
        subDiv.append(returnValue);
```

```
//-//
```

```
        subDiv.append(ce('hr'));  
    }  
}  
}
```

```
//--//
```

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

```
// rowOfLanguageShowAll.js
```

```
function rowOfLanguageShowAll(theX, theY,  
sizeX, sizeY, whichArray, criteria1, criteria2) {  
    let counter = 0;
```

//-//

```
let mainDiv = ce("div");
mainDiv.id = "scriptingSection";
mainDiv.style.position = "absolute";
mainDiv.title =
"rowOfLanguageShowAll(theX, theY, sizeX,
sizeY, whichArray, criteria1, criteria2)";
mainDiv.style.position = 'absolute';
mainDiv.style.top = theY + "px";
mainDiv.style.left = theX + "px";
mainDiv.style.display = "flex";
mainDiv.style.flexDirection = "row";
mainDiv.style.gap = "75px";
ba(mainDiv);
```

//-//

```
for (let i = 0; i < whichArray.length; i++) {
    if (whichArray[i].tag == criteria1 &&
whichArray[i].language == criteria2)
    {
        counter += 1;
    }
}
```

//-//

```
let theDiv = ce("div");  
theDiv.id = "JavaScriptAll" + counter;  
theDiv.className = "divStyle001";  
theDiv.style.width = sizeX + "px";  
theDiv.style.height = sizeY + "px";
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let span = ce("span");  
span.style.fontSize = "14pt";  
span.style.lineHeight = "15pt";  
span.textContent =  
whichArray[i].language;  
theDiv.append(span);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let boldAqua = ce("b");
```

```
boldAqua.style.color = "aqua";  
boldAqua.textContent =  
whichArray[i].category;  
theDiv.append(boldAqua);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let boldYellow = ce("b");  
boldYellow.style.color = "yellow";  
boldYellow.textContent =  
whichArray[i].name;  
theDiv.append(boldYellow);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className = "ourButton";  
printButton.style.marginRight = '10px';
```

```
printButton.textContent = "Print";  
printButton.onclick = function()  
{  
    printOnlyDiv(theDiv.id);  
};  
theDiv.append(printButton);
```

```
//-//
```

```
let urlLabel = ce("span");  
urlLabel.style.fontWeight = "bold";  
urlLabel.textContent = "URL: ";  
theDiv.append(urlLabel);
```

```
//-//
```

```
let urlLink = ce("a");  
urlLink.href = whichArray[i].url;  
urlLink.target = "_blank";  
urlLink.textContent = "Link";  
theDiv.append(urlLink);
```

```
//-//
```

```
let url2Label = ce("span");  
url2Label.style.fontWeight = "bold";
```

```
url2Label.textContent = "URL2: ";  
theDiv.append(url2Label);
```

```
//-//
```

```
let url2Link = ce("a");  
url2Link.href = whichArray[i].url2;  
url2Link.target = "_blank";  
url2Link.textContent = "Link 2";  
theDiv.append(url2Link);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let functionNameLabel = ce("span");  
functionNameLabel.style.fontWeight =  
"bold";  
functionNameLabel.style.color =  
"rgb(180, 180, 180)";  
functionNameLabel.innerHTML = "Name  
of Function: <br>";  
theDiv.append(functionNameLabel);
```



```
//-//
```

```
    let functionName = ce("span");  
    functionName.style.fontWeight =  
"bold";  
    functionName.textContent =  
whichArray[i].name;  
    theDiv.append(functionName);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
    let descriptionLabel = ce("span");  
    descriptionLabel.style.fontWeight =  
"bold";  
    descriptionLabel.style.color = "rgb(180,  
180, 180)";  
    descriptionLabel.textContent =  
"Description: ";  
    theDiv.append(descriptionLabel);
```

```
//-//
```

```
let description = ce("div");  
description.innerHTML =  
whichArray[i].description;  
theDiv.append(description);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let syntaxLabel = ce("span");  
syntaxLabel.style.fontWeight = "bold";  
syntaxLabel.style.color = "rgb(180, 180,  
180)";  
syntaxLabel.textContent = "Syntax: ";  
theDiv.append(syntaxLabel);
```

```
//-//
```

```
let syntax = ce("div");  
syntax.innerHTML =  
whichArray[i].syntax;  
theDiv.append(syntax);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let usageLabel = ce("span");  
usageLabel.style.fontWeight = "bold";  
usageLabel.style.color = "rgb(180, 180,  
180)";
```

```
usageLabel.textContent = "Usage: ";  
theDiv.append(usageLabel);
```

```
//-//
```

```
let usage = ce("div");  
usage.innerHTML =  
whichArray[i].usage;  
theDiv.append(usage);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let returnTypeLabel = ce("span");
```

```
returnTypeLabel.style.fontWeight =  
"bold";  
returnTypeLabel.style.color = "rgb(180,  
180, 180)";  
returnTypeLabel.textContent = "Return  
Value Type: ";  
theDiv.append(returnTypeLabel);
```

```
//-//
```

```
let returnType = ce("div");  
returnType.innerHTML =  
whichArray[i].type;  
theDiv.append(returnType);
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let deleteButton = ce("button");  
deleteButton.className = "ourButton";  
deleteButton.textContent = "Delete";  
deleteButton.onclick = function()  
{
```

```
        deleteThisRecord(theDiv.id);  
    };  
    theDiv.append(deleteButton);
```

```
//--//
```

```
theDiv.append(ce("hr"));
```

```
//--//
```

```
mainDiv.append(theDiv);
```

```
}
```

```
}
```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// rowOfNews.js
```

// makes divs in a row - custom for news Data

**function rowOfNews(theX, theY, sizeX, sizeY,
whichArray, criteria1)**

**{
 let counter = 0;**

//-//

**let mainDiv = ce("div");
 mainDiv.id = "newsSection";
 mainDiv.title = "rowOfNews(theX, theY,
sizeX, sizeY, whichArray, criteria1)";
 mainDiv.style.position = "absolute";
 mainDiv.style.left = theX + "px";
 mainDiv.style.top = theY + "px";
 mainDiv.style.display = "flex";
 mainDiv.style.flexDirection = "row";
 mainDiv.style.gap = "75px";
 ba(mainDiv);**

//-//

**for (let i = 0; i < whichArray.length; i++) {
 if (whichArray[i].tag == criteria1)
 {**

```
// filter by MONTH  
// if (whichArray[i].date.substring(0, 2)  
== 3)
```

```
// filter by Multiple MONTHS  
// if(whichArray[i].date.substring(0, 2) ==  
8 || whichArray[i].date.substring(0, 2) == 9)
```

```
// filter by DATE  
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018")
```

```
// filter by Multiple DATES  
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018" ||  
whichArray[i].date.substring(0, 10) ==  
"02/29/2020")
```

```
// filter by DAY of month  
// if (whichArray[i].date.substring(3, 5)  
== 2)
```

```
// filter by YEAR  
// if (whichArray[i].date.substring(6, 10)  
== 2019)
```

```
// filter by MULTIPLE YEARS
// if (whichArray[i].date.substring(6, 10)
== 2019 || whichArray[i].date.substring(6, 10)
== 2020)
```

```
// filter by MONTH and YEAR
// if (whichArray[i].date.substring(0, 2)
== 2 && whichArray[i].date.substring(6, 10) ==
2020)
```

```
// or show all by using
// if (whichArray)
```

```
// if, instead of filter function,
// is 70% faster
```

```
if (whichArray)
{
    counter += 1;
```

```
//-//
```

```
let theDiv = ce("div");
theDiv.id = "News" + counter;
theDiv.className = "divStyle001";
theDiv.style.width = sizeX + 'px';
```



```
theDiv.style.padding = "10px";  
mainDiv.append(theDiv);
```

```
//-//
```

```
let newsHeading = ce("div");  
newsHeading.textContent = "NEWS";  
let hr = ce("hr");  
hr.className = 'hrCompact';  
newsHeading.append(hr);  
theDiv.append(newsHeading);
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className =  
"ourButton";  
printButton.textContent = "Print";  
printButton.onclick = function()  
{  
    printOnlyDiv(theDiv.id);  
};  
theDiv.append(printButton);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let headline = ce("div");
```

```
headline.textContent =
```

```
whichArray[i].headline;
```

```
theDiv.append(headline);
```

```
//-//
```

```
let date = ce("div");
```

```
date.textContent = whichArray[i].date;
```

```
theDiv.append(date);
```

```
//-//
```

```
let link = ce("a");
```

```
link.href = whichArray[i].url;
```

```
link.target = "_blank";
```

```
link.textContent = "Link";
```

```
theDiv.append(link);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// rowOfNotes.js
```

```
function rowOfNotes(theX, theY, sizeX, sizeY,  
whichArray, criteria1)
```

```
{
```

```
    let counter = 0;
```

```
//--//
```

```
    let mainDiv = ce("div");
```

```
    mainDiv.id = "notesId";
```

```
    mainDiv.title = "rowOfNotes(theX, theY,  
sizeX, sizeY, whichArray, criteria1)";
```

```
    mainDiv.style.position = "absolute";
```

```
    mainDiv.style.left = theX + "px";
```

```
    mainDiv.style.top = theY + "px";
```

```
mainDiv.style.display = "flex";  
mainDiv.style.flexDirection = "row";  
mainDiv.style.gap = "75px";  
ba(mainDiv);
```

```
//-//
```

```
for (let i = 0; i < whichArray.length; i++) {  
    if (whichArray[i].tag == criteria1)  
    {  
        // filter by MONTH  
        // if (whichArray[i].date.substring(0, 2)  
== 3)  
  
        // filter by Multiple MONTHS  
        // if(whichArray[i].date.substring(0, 2) ==  
8 || whichArray[i].date.substring(0, 2) == 9)  
  
        // filter by DATE  
        // if (whichArray[i].date.substring(0, 10)  
== "02/29/2018")  
  
        // filter by Multiple DATES  
        // if (whichArray[i].date.substring(0, 10)  
== "02/29/2018" ||
```

```
whichArray[i].date.substring(0, 10) ==  
"02/29/2020")
```

```
// filter by DAY of month  
// if (whichArray[i].date.substring(3, 5)  
== 2)
```

```
// filter by YEAR  
// if (whichArray[i].date.substring(6, 10)  
== 2019)
```

```
// filter by MULTIPLE YEARS  
// if (whichArray[i].date.substring(6, 10)  
== 2019 || whichArray[i].date.substring(6, 10)  
== 2020)
```

```
// filter by MONTH and YEAR  
// if (whichArray[i].date.substring(0, 2)  
== 2 && whichArray[i].date.substring(6, 10) ==  
2020)
```

```
// or show all by using  
// if (whichArray)
```

```
if (whichArray)  
{
```

```
counter += 1;
```

```
//-//
```

```
let theDiv = ce("div");  
theDiv.id = "Notes" + counter;  
theDiv.className = "divStyle001";  
theDiv.style.width = "170px";  
theDiv.style.padding = "12px";  
mainDiv.append(theDiv);
```

```
//-//
```

```
let notesHeading = ce("div");  
notesHeading.textContent = "Notes  
Row";
```

```
//-//
```

```
theDiv.append(ce("hr"));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className =  
"ourButton";
```

```
printButton.textContent = "Print";  
printButton.onclick = function()  
{  
    printOnlyDiv(theDiv.id);  
};  
theDiv.append(printButton);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let date = ce("div");  
date.textContent = whichArray[i].date;  
theDiv.append(date);
```

```
//-//
```

```
let number = ce("div");  
number.textContent =  
whichArray[i].number;  
theDiv.append(number);
```

```
//-//
```

```
        let note = ce("div");
        note.textContent =
whichArray[i].note;
        theDiv.append(note);
    }
}
}
```

//--//

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

// rowOfNotesWithTextArea.js

```
function rowOfNotesWithTextArea(theX, theY,
sizeX, sizeY, whichArray, criteria1)
{
    let counter = 0;
```

//--//


```

let mainDiv = ce("div");
mainDiv.id = "notesWithTextareald";
mainDiv.title =
"rowOfNotesWithTextArea(theX, theY, sizeX,
sizeY, whichArray, criteria1)";
mainDiv.style.position = "absolute";
mainDiv.style.left = theX + "px";
mainDiv.style.top = theY + "px";
mainDiv.style.display = "flex";
mainDiv.style.flexDirection = "row";
mainDiv.style.gap = "75px";
ba(mainDiv);

```

```
//-//
```

```

for (let i = 0; i < whichArray.length; i++) {
  if (whichArray[i].tag == criteria1)
  {
    // filter by MONTH
    // if (whichArray[i].date.substring(0, 2)
== 3)

    // filter by Multiple MONTHS
    // if(whichArray[i].date.substring(0, 2) ==
8 || whichArray[i].date.substring(0, 2) == 9)

```

```
// filter by DATE  
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018")
```

```
// filter by Multiple DATES  
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018" ||  
whichArray[i].date.substring(0, 10) ==  
"02/29/2020")
```

```
// filter by DAY of month  
// if (whichArray[i].date.substring(3, 5)  
== 2)
```

```
// filter by YEAR  
// if (whichArray[i].date.substring(6, 10)  
== 2019)
```

```
// filter by MULTIPLE YEARS  
// if (whichArray[i].date.substring(6, 10)  
== 2019 || whichArray[i].date.substring(6, 10)  
== 2020)
```

```
// filter by MONTH and YEAR
```

```
// if (whichArray[i].date.substring(0, 2)  
== 2 && whichArray[i].date.substring(6, 10) ==  
2020)
```

```
// or show all by using  
// if (whichArray)
```

```
if (whichArray)  
{  
    counter += 1;
```

```
//-//
```

```
let subDiv = ce("div");  
subDiv.className = "divStyle001";  
mainDiv.append(subDiv);
```

```
//-//
```

```
let theTextArea = ce("textarea");  
theTextArea.id = "NotesWithTextArea"  
+ counter;  
theTextArea.className =  
"divStyle001";  
theTextArea.style.width = (sizeX - 30)  
+ 'px';
```

```
theTextArea.style.height = (sizeY - 50)
+ 'px';
theTextArea.style.resize = "none";
theTextArea.setAttribute("readonly",
"true");
```

```
theTextArea.value = "Notes Row with
Text Area Scrolling" + "\n";
theTextArea.value +=
whichArray[i].date + "\n";
theTextArea.value +=
whichArray[i].number + "\n";
theTextArea.value +=
whichArray[i].note + "\n";
subDiv.append(theTextArea);
```

```
//-//
```

```
let printButton = ce("button");
printButton.className =
"ourButton";
printButton.textContent = "Print";
printButton.onclick = function()
{
    printTextArea(theTextArea.id);
};
```

```
        subDiv.append(printButton);
    }
}
}
}

//--//

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

// rowOfSciencelImagesAndLink.js

function rowOfSciencelImagesAndLink(theX,
theY, sizeX, sizeY, whichArray, criteria1,
criteria2) {
    let counter = 0;

    //--//

    let mainDiv = ce("div");
    mainDiv.id = "notesWithTextareald";
```

```
mainDiv.title =  
"rowOfScienceImagesAndLink(theX, theY,  
sizeX, sizeY, whichArray, criteria1, criteria2)";  
mainDiv.style.position = "absolute";  
mainDiv.style.left = theX + "px";  
mainDiv.style.top = theY + "px";  
mainDiv.style.display = "flex";  
mainDiv.style.flexDirection = "row";  
mainDiv.style.gap = "75px";  
ba(mainDiv);
```

```
//-//
```

```
for (let i = 0; i < whichArray.length; i++) {  
    if (whichArray[i].tag == criteria1 ||  
whichArray[i].description == criteria2) {  
        // filter by MONTH  
        // if (whichArray[i].date.substring(0, 2)  
== 3)
```

```
        // filter by Multiple MONTHS  
        // if(whichArray[i].date.substring(0, 2) ==  
8 || whichArray[i].date.substring(0, 2) == 9)
```

```
        // filter by DATE
```

```
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018")
```

```
// filter by Multiple DATES  
// if (whichArray[i].date.substring(0, 10)  
== "02/29/2018" ||  
whichArray[i].date.substring(0, 10) ==  
"02/29/2020")
```

```
// filter by DAY of month  
// if (whichArray[i].date.substring(3, 5)  
== 2)
```

```
// filter by YEAR  
// if (whichArray[i].date.substring(6, 10)  
== 2019)
```

```
// filter by MULTIPLE YEARS  
// if (whichArray[i].date.substring(6, 10)  
== 2019 || whichArray[i].date.substring(6, 10)  
== 2020)
```

```
// filter by MONTH and YEAR  
// if (whichArray[i].date.substring(0, 2)  
== 2 && whichArray[i].date.substring(6, 10) ==  
2020)
```

```
// or show all by using  
// if (whichArray)
```

```
if (whichArray)  
{  
    counter += 1;
```

```
//-//
```

```
    let subDiv = ce("div");  
    subDiv.id =  
"ScienceWithImageAndLink" + counter;  
    subDiv.className = "divStyle001";  
    subDiv.style.width = sizeX + 'px';  
    subDiv.style.height = sizeY + 'px';  
    mainDiv.append(subDiv);
```

```
//-//
```

```
    let theContent = ce("div");  
    theContent.textContent = "Science  
Row with Image and Link";  
    subDiv.append(theContent);
```

```
//-//
```



```
theContent.append(ce("hr"));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.className =  
"ourButton";  
printButton.textContent = "Print";  
printButton.onclick = function()  
{  
    printOnlyDiv(subDiv.id);  
};  
theContent.append(printButton);
```

```
//-//
```

```
theContent.append(ce("br"));
```

```
//-//
```

```
let title = ce("div");  
title.textContent = whichArray[i].title;  
theContent.append(title);
```

```
//-//
```

```
theContent.append(ce("br"));
```

```
//-//
```

```
let date = ce("div");  
date.textContent = whichArray[i].date;  
theContent.append(date);
```

```
//-//
```

```
theContent.append(ce("br"));
```

```
//-//
```

```
let info = ce("div");  
info.textContent = whichArray[i].info;  
theContent.append(info);
```

```
//-//
```

```
theContent.append(ce("br"));
```

```
//-//
```

```
let details = ce("details");
```

```
let link = ce("a");
link.href = whichArray[i].url;
link.target = "_blank";
link.textContent = whichArray[i].url;
details.append(link);
theContent.append(details);
```

```
//-
```

```
theContent.append(ce("br"));
```

```
//-
```

```
let theImage = ce("img");
theImage.src = whichArray[i].image;
theImage.className =
"theSciencelImages";
theContent.append(theImage);
    }
  }
}
}
```

```
//--
```

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// rowOfSocialMediaEvidence.js
```

```
function rowOfSocialMediaEvidence(theX,  
theY, sizeX, sizeY, whichArray, criteria1)
```

```
{  
    let counter = 0;
```

```
    //-//
```

```
    let mainDiv = ce("div");  
    mainDiv.id = "notesWithTextareald";  
    mainDiv.title =  
"rowOfSocialMediaEvidence(theX, theY, sizeX,  
sizeY, whichArray, criteria1)";  
    mainDiv.style.position = "absolute";  
    mainDiv.style.left = theX + "px";  
    mainDiv.style.top = theY + "px";  
    mainDiv.style.display = "flex";  
    mainDiv.style.flexDirection = "row";  
    mainDiv.style.gap = "75px";
```

```
ba(mainDiv);
```

```
//-//
```

```
for (let i = 0; i < whichArray.length; i++) {  
    if (whichArray[i].tag == criteria1 &&  
whichArray[i].date.substring(6, 10) == 2020 &&  
whichArray[i].probability >= 50)  
    {  
        counter += 1;
```

```
//-//
```

```
    let theDiv = ce("div");  
    theDiv.id = "SocialMediaEvidence" +  
counter;  
    theDiv.className = "divStyle001";  
    theDiv.style.width = sizeX + 'px';  
    theDiv.style.height = sizeY + 'px';  
    theDiv.textContent = 'Social Media  
Evidence Rating';  
    mainDiv.append(theDiv);
```

```
//-//
```

```
theDiv.append(ce('hr'));
```

//-//

```
let printButton = ce("button");  
printButton.className = "ourButton";  
printButton.textContent = "print";  
printButton.onclick = function()  
{  
    printOnlyDiv(theDiv.id);  
};  
theDiv.append(printButton);
```

//-//

```
theDiv.append(ce("br"));
```

//-//

```
let title = ce("div");  
title.textContent = whichArray[i].title;  
theDiv.append(title);
```

//-//

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let date = ce("div");  
date.textContent = whichArray[i].date;  
theDiv.append(date);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let info = ce("div");  
info.textContent = whichArray[i].info;  
theDiv.append(info);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let probability = ce("div");  
probability.textContent = "Probability  
Rating: " + whichArray[i].probability + "%";  
theDiv.append(probability);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let theLink = ce("a");  
theLink.href = whichArray[i].url;  
theLink.target = "_blank";  
theLink.textContent = "Link";  
theDiv.append(theLink);
```

```
//-//
```

```
theDiv.append(ce("br"));
```

```
//-//
```

```
let theImage = ce("img");  
theImage.src = whichArray[i].image;  
theImage.className = "theImage";  
theDiv.append(theImage);
```

```
}
```

```
}
```

```
}
```


//--//

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// rowOfVideos.js

**// adds videos to the page. The videos must
allow embed. On youtube, we copy the url in
the Address Bar**

// for example

**[https://www.youtube.com/watch?v=Q4tPG-
cwiWE](https://www.youtube.com/watch?v=Q4tPG-cwiWE)**

// then we just replace watch?v= with embed/

// for example

**[https://www.youtube.com/embed/Q4tPG-
cwiWE](https://www.youtube.com/embed/Q4tPG-cwiWE)**

```
function rowOfVideos(theX, theY, sizeX, sizeY,  
whichArray, criteria)
```

```
{  
    let counter = 0;
```

```
//-//
```

```
    let mainDiv = ce("div");  
    mainDiv.id = "videoEvidenceld";  
    mainDiv.title = "rowOfVideos(theX, theY,  
sizeX, sizeY, whichArray, criteria)";  
    mainDiv.style.position = "absolute";  
    mainDiv.style.left = theX + "px";  
    mainDiv.style.top = theY + "px";  
    mainDiv.style.display = "flex";  
    mainDiv.style.flexDirection = "row";  
    mainDiv.style.gap = "75px";  
    ba(mainDiv);
```

```
//-//
```

```
for (let i = 0; i < whichArray.length; i++)  
{  
    if (whichArray[i].description === criteria)  
    {  
        counter += 1;
```

```
//-//
```

```
let subDiv = ce("div");  
subDiv.id = whichArray[i].title + i;  
subDiv.className = "divStyle001";  
subDiv.style.width = sizeX + 'px';  
mainDiv.append(subDiv);
```

```
//-//
```

```
let titleText = ce("span");  
titleText.textContent =  
whichArray[i].title;  
subDiv.append(titleText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let uploadDateText = ce("span");  
uploadDateText.textContent = "upload  
date " + whichArray[i].date;  
subDiv.append(uploadDateText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let urlText = ce("span");  
urlText.textContent = whichArray[i].url;  
subDiv.append(urlText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let linkElement = ce("a");  
linkElement.href = whichArray[i].url;  
linkElement.target = "_blank";  
linkElement.textContent = "Link";  
subDiv.append(linkElement);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let criteriaText = ce("b");  
criteriaText.style.color = "aqua";  
criteriaText.textContent = criteria;  
subDiv.append(criteriaText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.textContent = "print";  
printButton.className = "ourButton";  
printButton.onclick = function()  
{  
    printOnlyDiv(subDiv.id);  
};  
subDiv.append(printButton);
```

```
//-//
```

```
let makeVideo = ce("iframe");
```

```
makeVideo.src = whichArray[i].url;
makeVideo.style.width = sizeX + 'px';
makeVideo.style.height = sizeY + 'px';
subDiv.append(makeVideo);
    }
}
}

//--//

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

// rowOfVideosMultipleFilters.js

function rowOfVideosMultipleFilters(theX, theY,
sizeX, sizeY, whichArray, criteria1, criteria2)
{
    let counter = 0;

    //--//
```

```
let mainDiv = ce("div");
mainDiv.id = "videoEvidenceMultipleFilters";
mainDiv.title =
"rowOfVideosMultipleFilters(theX, theY, sizeX,
sizeY, whichArray, criteria1, criteria2)";
mainDiv.style.position = "absolute";
mainDiv.style.left = theX + "px";
mainDiv.style.top = theY + "px";
mainDiv.style.display = "flex";
mainDiv.style.flexDirection = "row";
mainDiv.style.gap = "75px";
ba(mainDiv);
```

```
//-//
```

```
for (let i = 0; i < whichArray.length; i++)
{
    if (whichArray[i].tag === criteria1 &&
whichArray[i].description === criteria2)
    {
        counter += 1;
```

```
//-//
```

```
let subDiv = ce("div");
subDiv.id = whichArray[i].title + i;
```

```
subDiv.className = "divStyle001";  
subDiv.style.width = sizeX + 'px';  
mainDiv.append(subDiv);
```

```
//-//
```

```
let criteria1Text = ce("b");  
criteria1Text.style.color = "yellow";  
criteria1Text.textContent = criteria1;  
subDiv.append(criteria1Text);
```

```
//-//
```

```
let separatorText = ce("span");  
separatorText.textContent = " - ";  
subDiv.append(separatorText);
```

```
//-//
```

```
let criteria2Text = ce("b");  
criteria2Text.style.color = "aqua";  
criteria2Text.textContent = criteria2;  
subDiv.append(criteria2Text);
```

```
//-//
```



```
subDiv.append(ce("br"));
```

```
//-//
```

```
let titleText = ce("span");  
titleText.textContent =  
whichArray[i].title;  
subDiv.append(titleText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let uploadDateText = ce("span");  
uploadDateText.textContent = "upload  
date " + whichArray[i].date;  
subDiv.append(uploadDateText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let urlText = ce("span");  
urlText.textContent = whichArray[i].url;  
subDiv.append(urlText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let linkElement = ce("a");  
linkElement.href = whichArray[i].url;  
linkElement.target = "_blank";  
linkElement.textContent = "Link";  
subDiv.append(linkElement);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.textContent = "print";  
printButton.className = "ourButton";  
printButton.onclick = function()
```

```
{
    printOnlyDiv(subDiv.id);
};
subDiv.append(printButton);
```

```
//--//
```

```
let theVideo = ce("iframe");
theVideo.src = whichArray[i].url;
theVideo.style.width = sizeX + 'px';
theVideo.style.height = sizeY + 'px';
subDiv.append(theVideo);
```

```
}
```

```
}
```

```
}
```

```
//--//
```

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

// rowOfVideosOnlineStartEnd.js

**// makes a row of videos, using internet url
embed enabled videos, with a reload button, so
that the video and/or audio can be played again
from the start point.**

**function makeVideosOnlineStartEnd(theX,
theY, sizeX, sizeY, whichArray, criteria1,
criteria2)**

**{
 let counter = 0;**

//-//

**let mainDiv = ce("div");
 mainDiv.id = "makeVideosOnlineStartEnd";
 mainDiv.title =
"makeVideosOnlineStartEnd(theX, theY, sizeX,
sizeY, whichArray, criteria1, criteria2);"
 mainDiv.style.position = "absolute";
 mainDiv.style.left = theX + "px";
 mainDiv.style.top = theY + "px";
 mainDiv.style.display = "flex";
 mainDiv.style.flexDirection = "row";
 mainDiv.style.gap = "75px";
 ba(mainDiv);**

//-//

```
for (let i = 0; i < whichArray.length; i++)  
{  
    if (whichArray[i].tag === criteria1 &&  
whichArray[i].description === criteria2)  
    {  
        counter += 1;
```

//-//

```
let subDiv = ce("div");  
subDiv.id = whichArray[i].title;  
subDiv.className = "divStyle001";  
subDiv.style.width = sizeX + 'px';  
mainDiv.append(subDiv);
```

//-//

```
let criteria1Text = ce("b");  
criteria1Text.style.color = "yellow";  
criteria1Text.textContent = criteria1;  
subDiv.append(criteria1Text);
```

//-//

```
let separatorText = ce("span");  
separatorText.textContent = " - ";  
subDiv.append(separatorText);
```

```
//-//
```

```
let criteria2Text = ce("b");  
criteria2Text.style.color = "aqua";  
criteria2Text.textContent = criteria2;  
subDiv.append(criteria2Text);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let titleText = ce("span");  
titleText.textContent =  
whichArray[i].title;  
subDiv.append(titleText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let uploadDateText = ce("span");  
uploadDateText.textContent = "upload  
date " + whichArray[i].date;  
subDiv.append(uploadDateText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let urlText = ce("span");  
urlText.textContent = whichArray[i].url;  
subDiv.append(urlText);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let theLink = ce("a");  
theLink.href = whichArray[i].url;
```

```
theLink.target = "_blank";  
theLink.textContent = "Link";  
subDiv.append(theLink);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let printButton = ce("button");  
printButton.textContent = "print";  
printButton.className = "ourButton";  
printButton.onclick = function()  
{  
    printOnlyDiv(subDiv.id);  
};  
subDiv.append(printButton);
```

```
//-//
```

```
let theVideo = ce("iframe");  
theVideo.src = whichArray[i].url;  
theVideo.style.width = sizeX + 'px';  
theVideo.style.height = sizeY + 'px';  
subDiv.append(theVideo);
```



```
//-//
```

```
let reloadButton = ce("button");  
reloadButton.className = "ourButton";  
reloadButton.onclick = function()  
{  
    theVideo.src = theVideo.src;  
};  
reloadButton.textContent = "Reload";  
subDiv.append(reloadButton);  
}
```

```
}  
}
```

```
//--//
```

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

// Makes iframes in a row to add content to our level can be used to display Websites or Videos the websites must allow iframe the videos must allow embed

function rowOfWebsitesOrVideos(theX, theY, sizeX, sizeY, whichArray, criteria1)

{

let counter = 0;

//-//

let mainDiv = ce("div");

mainDiv.id = "rowOfWebsitesOrVideos";

mainDiv.title =

"rowOfWebsitesOrVideos(theX, theY, sizeX, sizeY, whichArray, criteria1)";

mainDiv.style.position = "absolute";

mainDiv.style.left = theX + "px";

mainDiv.style.top = theY + "px";

mainDiv.style.display = "flex";

mainDiv.style.flexDirection = "row";

mainDiv.style.gap = "75px";

ba(mainDiv);

//-//

```
for (let i = 0; i < whichArray.length; i++) {  
  if (whichArray[i].tag == criteria1)  
  {  
    counter += 1;
```

```
//-//
```

```
let subDiv = ce("div");  
subDiv.id = whichArray[i].title + i;  
subDiv.className = "divStyle001";  
mainDiv.append(subDiv);
```

```
//-//
```

```
let printButton = ce("button");  
printButton.textContent = "print";  
printButton.className = "ourButton";  
printButton.onclick = function()  
{  
  printOnlyDiv(subDiv.id);  
};  
subDiv.append(printButton);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let boldCriteria = ce("b");  
boldCriteria.style.color = "yellow";  
boldCriteria.textContent = criteria1;  
subDiv.append(boldCriteria);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let boldTitle = ce("b");  
boldTitle.textContent =  
whichArray[i].title;  
subDiv.append(boldTitle);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let uploadDateText = ce("span");
uploadDateText.textContent = "upload
date " + whichArray[i].date;
subDiv.append(uploadDateText);

//-//

subDiv.append(ce("br"));

//-//

let urlText = ce("span");
urlText.textContent = whichArray[i].url;
subDiv.append(urlText);

//-//

subDiv.append(ce("br"));

//-//

let linkElement = ce("a");
linkElement.href = whichArray[i].url;
linkElement.target = "_blank";
linkElement.textContent = "Link";
subDiv.append(linkElement);
```

```
//-//
```

```
subDiv.append(ce("br"));
```

```
//-//
```

```
let theFrame = ce("iframe");  
theFrame.src = whichArray[i].url;  
theFrame.style.width = sizeX + 'px';  
theFrame.style.height = sizeY + 'px';  
subDiv.append(theFrame);
```

```
}
```

```
}
```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// soundsArray.js
```

```
let soundsArray =  
[  
  {  
    name: 'sfx_blip_001',  
    soundFileOffline:  
'src/media/sounds/sfx_blip_001.mp4',  
    soundFileOnline:  
'https://collegeofscripting.weebly.com/uploads/  
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/ChristopherAndrewTopalian
```

```
// songsArray.js
```

```
let songsArray =  
[  
  {  
    name: "Emotional Blockbuster 2",  
    theId: "EmotionalBlockbuster2",  
    soundFileOffline:  
'src/media/songs/emotional_blockbuster_2.mp  
3',  
    soundFileOnline: "https://64482293-  
555277121853380144.preview.editmysite.com/u  
ploads/6/4/4/8/64482293/  
emotional_blockbuster_2.mp3"  
  },  
  {  
    name: "Video Game Blockbuster",  
    theId: "VideoGameBlockbuster",
```



```
    soundFileOffline:
"src/media/songs/video_game_blockbuster.mp
3",
    soundFileOnline: "https://64482293-
555277121853380144.preview.editmysite.com/u
ploads/6/4/4/8/64482293/
video_game_blockbuster.mp3"
},
{
    name: "Ice and Snow",
    theId: "iceAndSnow",
    soundFileOffline:
"src/media/songs/ice_and_snow.mp3",
    soundFileOnline:
"https://collegeofscripting.weebly.com/uploads
/6/4/4/8/64482293/ice_and_snow.mp3"
},
{
    name: "Kings Trailer",
    theId: "kingsTrailer",
    soundFileOffline:
"src/media/songs/kings_trailer.mp3",
    soundFileOnline:
"https://collegeofscripting.weebly.com/uploads
/6/4/4/8/64482293/kings_trailer.mp3"
},
```

```
{  
  name: "Lonely Mountain",  
  theld: "lonelyMontain",  
  soundFileOffline:  
"src/media/songs/lonely_mountain.mp3",  
  soundFileOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/lonely_mountain.mp3"  
},  
{  
  name: "Parhelion",  
  theld: "parehelion",  
  soundFileOffline:  
"src/media/songs/parhelion.mp3",  
  soundFileOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/parhelion.mp3"  
},  
{  
  name: "Night Vigil",  
  theld: "nightVigil",  
  soundFileOffline:  
"src/media/songs/night_vigil.mp3",  
  soundFileOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/night_vigil.mp3"
```

```
},  
{  
    name: "The Abyss",  
    theld: "theAbyss",  
    soundFileOffline:  
"src/media/songs/the_abyss.mp3",  
    soundFileOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/the_abyss.mp3"  
},  
{  
    name: "Burt's Requiem",  
    theld: "burtsRequiem",  
    soundFileOffline:  
"src/media/songs/burts_requiem.mp3",  
    soundFileOnline:  
"https://collegeofscripting.weebly.com/uploads  
/6/4/4/8/64482293/burts_requiem.mp3"  
},  
{  
    name: "Horizon Flare",  
    theld: "horizonFlare",  
    soundFileOffline:  
"src/media/songs/horizon_flare.mp3",
```

```
    soundFileOnline:
    "https://collegeofscripting.weebly.com/uploads
    /6/4/4/8/64482293/horizon_flare.mp3"
    },
    {
        name: "Motions",
        theId: "motions",
        soundFileOffline:
        "src/media/songs/motions.mp3",
        soundFileOnline:
        "https://collegeofscripting.weebly.com/uploads
        /6/4/4/8/64482293/motions.mp3"
    }
];
```

//--//

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

```
function audioPlay(whichId, whichVolume)  
{  
    let audio = ge(whichId);  
  
    if (audio)  
    {  
        audio.volume = whichVolume;  
        audio.play();  
    }  
}
```

```
//--//
```

```
// example
```

```
// audioPlay("blip", 1.0);
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// enterButtonSound.js
```

```
function enterButtonSound()
```

```
{  
  audioPlay("sfx_blip_001", 1.0);  
}
```

```
//--//
```

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

```
// playSong.js
```

```
function playSong(whichSongId)  
{  
  let song = ge(whichSongId);  
  song.volume = songVolume;  
  songChosenId = song.id + "Button";  
  console.log(songChosenId);  
  
  // color button of song chosen  
  ge(songChosenId).style.background =  
  "rgb(128,128,128)";
```

```
// console.log(song.id);

if (songFlag == 0)
{
    song.volume= songVolume;
    song.play();
    songFlag = 1;
}

else if (songFlag == 1)
{
    // color chosen song's Button back to its
    default color
    ge(songChosenId).style.background =
    "rgb(0, 77, 129)";

    let song = ge(whichSongId);
    song.pause();
    songFlag = 0;
}
}

//--//

// example
```

```
// playSong("emotionalBlockbuster2Song",  
1.0);
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// songChoicesMenu.js
```

```
// song Menu Div that is Created when Songs  
Button is clicked
```

```
function songChoicesMenu()  
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
//-//
```

```
if (ge("songMenu"))  
{  
    removeElement("songMenu");  
}
```


//-//

```
let songDiv = ce("div");
songDiv.id = "songMenu";
songDiv.style.position = "absolute";
songDiv.style.left = ourPlayer.x + 150 + 'px';
songDiv.style.top = ourPlayer.y + 'px';
songDiv.style.width = "300px";
songDiv.style.height = "434px";
songDiv.style.zIndex = "998";
songDiv.style.background = "rgb(0,0,0)";
songDiv.style.border = "solid 5px
rgb(255,255,255)";
songDiv.style.borderRadius = "10px";
songDiv.style.fontFamily = "arial";
songDiv.style.fontWeight = "normal";
songDiv.style.fontSize = "medium";
songDiv.style.color = "white";
songDiv.style.textAlign = "center";
songDiv.style.overflowY = "scroll";
ba(songDiv);
```

//-//

```
let closeButton = ce("button");
closeButton.className = "ourButton";
```

```
closeButton.style.marginBottom = "-1px";
closeButton.textContent = "X";
closeButton.onclick = function()
{
    removeElement(songDiv.id);
};
songDiv.append(closeButton);
```

```
//-//
```

```
songDiv.append(ce('hr'));
```

```
//-//
```

```
let volumeTitle = ce("span");
volumeTitle.innerHTML = "<b> VOLUME
</b>";
songDiv.append(volumeTitle);
```

```
//-//
```

```
songDiv.append(ce('hr'));
```

```
//-//
```

```
for (let i = 1; i <= 10; i++)
```

```
{  
  let volumeButton = ce("button");  
  volumeButton.className = "ourButton";  
  volumeButton.textContent = i;  
  volumeButton.onclick = function()  
  {  
    volumeSet((i / 10).toString());  
  };  
  songDiv.append(volumeButton);  
}  
  
//-//  
  
songDiv.append(ce('hr'));  
  
//-//  
  
let songsTitle = ce("span");  
songsTitle.innerHTML = "<b> SONGS </b>";  
songDiv.append(songsTitle);  
  
//-//  
  
songDiv.append(ce('hr'));  
  
//-//
```

```
for (let z = 0; z < songsArray.length; z++) {  
    let songButton = ce("button");  
    songButton.className = "ourButton";  
    songButton.id = songsArray[z].name +  
"Button";  
    songButton.textContent =  
songsArray[z].name;  
    songButton.onclick = function()  
    {  
        playSong(songsArray[z].name);  
    };  
    songDiv.append(songButton);  
}  
  
if (songFlag == 1)  
{  
    // color button of song chosen  
    ge(songChosenId).style.background =  
"rgb(128, 128, 128)";  
}  
  
if (songFlag == 0)  
{  
    // color button of song chosen
```

```
        ge(songChosenId).style.background =  
"rgb(0, 77, 129)";  
    }  
}
```

//--//

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

// sound.js

// PLAY SONG and use a Toggle Button system
for activating and stopping the songs

let songFlag = 0;

let songVolume = 0.3;

let songChosenId;

let songActive = 0;

```
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/ChristopherAndrewTopalian
```

```
// stopSound.js
```

```
function stopSound(whichSongId)  
{  
    let song = ge(whichSongId);  
  
    song.pause();  
}
```

```
//--//
```

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

// volumeSet.js

```
function volumeSet(theVolume)
{
    songVolume = theVolume;
}
```

//--//

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

// structuresColorAll.js

// Color All Structures

```
function structuresColorAll()
{
    let theStructure;

    for (let x = 1; x <= structuresArray.length; x+
+)
```



```
{  
    theStructure = "STRUCTURE" + x;  
  
    ge(theStructure).style.backgroundColor =  
    "rgb(70, 95, 140)";  
}  
}  
  
/--//  
  
// Dedicated to God the Father  
// All Rights Reserved Christopher Andrew  
// Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
  
// structuresCreate.js
```

```
/*  
CREATE a STRUCTURE:  
After the Letter B on the keyboard is pressed,  
or the Create Button is Left Clicked in the  
Player Menu, it activates Structure Create  
Mode, which allows us to place structures in  
the world using the Left Mouse Button.
```

Esc Button is used to deactivate Structure Mode.

***/**

let structureId = 1;

function structuresCreate(theEvent, theX, theY)
{
 audioPlay("sfx_blip_001", 1.0);

//-//

theX = theEvent.pageX;
theY = theEvent.pageY;

//-//

let sizeX = 110;
let sizeY = 45;

//-//

let colorFill = "black";
let colorText = "white";
let structureOpacity = 1.0;
let theTexture = "Default";

```
//-//
```

```
let ourStructure = ce("div");  
ourStructure.style.position = "absolute";
```

// For a Center Position System, use this way below

// We minus the structures size divided by itself.

// This way we place structures according to their center, instead of their top left corner.

```
ourStructure.style.left = theX - (sizeX / 2)  
+'px';
```

```
ourStructure.style.top = theY - (sizeY / 2) +  
'px';
```

```
/*
```

// For a Top Left Coordinate System, use this way below

```
ourStructure.style.left = theX +'px';  
ourStructure.style.top = theY + 'px';  
*/
```

```
// ourStructure.id = "STRUCTURE" +  
structureId;  
  
ourStructure.style.width = sizeX +'px';  
ourStructure.style.height = sizeY +'px';  
  
// id  
ourStructure.id = structuresArray.length;  
  
// ourStructure.setAttribute('onclick',  
'structuresGetInfo(\""+ourStructure.id+"\");');  
  
// Open the Structure Edit Menu for the  
clicked structure  
ourStructure.setAttribute('onclick',  
'menuStructureEdit(\""+ourStructure.id+"\");');  
  
// when clicked activate two functions  
// ourStructure.setAttribute('onclick',  
'menuStructureEdit(\""+ourStructure.id+"\");stru  
cturesGetInfo(\""+ourStructure.id+"\");');  
  
ourStructure.style.zIndex = "1002";  
ourStructure.style.border = "solid 2px  
rgba(100, 100, 255, 1.0)";  
ourStructure.style.background = colorFill;
```

```
ourStructure.style.opacity =  
structureOpacity;
```

```
// ourStructure.style.backgroundImage =  
'url("https://collegeofscripting.weebly.com/uplo  
ads/6/4/4/8/64482293/wood1.jpg")';
```

```
ourStructure.style.backgroundImage=  
theTexture;
```

```
// PADDING
```

```
// ourStructure.style.paddingTop = "2px";
```

```
ourStructure.style.fontFamily = "arial";  
ourStructure.style.fontWeight = "bold";  
ourStructure.style.fontSize = "20px";  
ourStructure.style.color = colorText;  
ourStructure.style.textAlign = "center";
```

```
ourStructure.style.overflowY = "scroll";
```

```
// show the ID, X, and Y, on the STRUCTURE  
ourStructure.innerHTML = ourStructure.id +  
"<br>" + "Name" + "<br>" + theX + ", " + theY;
```

```
// structuresArray[structureId].name
```

```
let structureAttributes =  
{  
  // name: ourStructure.id, //name  
  
  name: "Name", // name  
  id: ourStructure.id, // id  
  posX: theX, // posX  
  posY: theY, //posY  
  sizeX: sizeX, // sizeX  
  sizeY: sizeY, // sizeY  
  colorFill: colorFill, // fill color  
  colorText: colorText, // text color  
  opacity: structureOpacity, //opacity  
  texture: theTexture // texture  
};  
  
// add the new structure's name, id, posX,  
posY, sizeX, and sizeY to our structuresArray  
structuresArray.push(structureAttributes);  
  
console.log(JSON.stringify(structuresArray));  
  
ba(ourStructure);
```

```
    showStructureData();
}

function buttonCreatePressed()
{
    // audioPlay("sfx_blip_001", 0.1);

    ge(activePlayer).style.backgroundColor =
    "rgb(100, 75, 255)";

    ge(activePlayer).innerHTML =
    "<br><br><br><br>STRUC- <br>TURE<br>
    MODE<br>ON";

    // ge("createButton").click();

    // setTimeout is used to ensure that the
    structure is not created prior to the user
    choosing where to place it, after pressing the
    create button
    setTimeout(function()
    {
        addListen('click', structuresCreate);
    }, 1000);

    // addListen('click', structuresCreate);
```

```
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// structuresCreateFromFile.js
```

```
function structuresCreateFromFile()
```

```
{
```

```
//
```

```
console.log(JSON.stringify(structuresArray));
```

```
// the structuresArray that we use in this  
function, is located in structureData.js
```

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

```
    let ourStructure = ce("div");
```

```
    ourStructure.style.position = "absolute";
```


// For a Center Position System, use this way below
// We minus the structures size divided by itself.

// This way we place structures according to their center, instead of their top left corner.

**ourStructure.style.left =
structuresArray[z].posX -
(structuresArray[z].sizeX / 2) + 'px';**

**ourStructure.style.top =
structuresArray[z].posY -
(structuresArray[z].sizeY / 2) + 'px';**

/*

// For a Top Left Coordinate System, use this way below

**ourStructure.style.left =
structuresArray[z].posX + 'px';**

**ourStructure.style.top =
structuresArray[z].posY + 'px';**

***/**

// SIZE

```
ourStructure.style.width =  
structuresArray[z].sizeX +'px';
```

```
ourStructure.style.height =  
structuresArray[z].sizeY +'px';
```

```
ourStructure.style.overflowY = "scroll";
```

```
// PADDING
```

```
// ourStructure.style.paddingTop = "2px";
```

```
// "STRUCTURE" + structureId;
```

```
// id
```

```
ourStructure.id = structuresArray[z].id;
```

```
ourStructure.style.resize = 'both';
```

```
// Show info about the structure, under the  
structure
```

```
// ourStructure.setAttribute('onclick',  
'structuresGetInfo(\"'+ourStructure.id+'\");');
```

```
// when right clicked
```

```
// open structure edit menu
```

```
ourStructure.oncontextmenu = function()
```

```
{
    menuStructureEdit(ourStructure.id);
};

// ourStructure.setAttribute('onclick',
'menuStructureEdit(\"+ourStructure.id+\");stru
cturesGetInfo(\"+ourStructure.id+\");');

ourStructure.style.zIndex = "1002";

if (structuresArray[z].texture != "none")
{
    ourStructure.style.backgroundImage =
structuresArray[z].texture;
}
else
{
    ourStructure.style.background =
structuresArray[z].colorFill;
}
ourStructure.style.border = "solid 2px";
ourStructure.style.borderColor =
"rgba(100, 100, 255, 1.0)";
ourStructure.style.opacity =
structuresArray[z].opacity;
```

```
// FONT
ourStructure.style.fontFamily = "arial";
ourStructure.style.fontWeight = "bold";
ourStructure.style.fontSize = "20px";
ourStructure.style.color =
structuresArray[z].colorText;
ourStructure.style.textAlign = "center";

// Show the id of the structure and also the
simplified position data
ourStructure.innerHTML =
structuresArray[z].id + "<br>" +
structuresArray[z].name + "<br>" +
structuresArray[z].posX.toFixed(0) + "<br>" +
structuresArray[z].posY.toFixed(0);

    ba(ourStructure);
}

//console.log(JSON.stringify(structuresArray));
}
```

// example

// structuresArray data in the console

```
// console.log(JSON.stringify(structuresArray,  
null, ' ') + ";" + "\n\n");
```

```
//--//
```

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

```
// structuresGoTo.js
```

```
// Send the Player to the Chosen Structure  
Location
```

```
function structuresGoTo(whichId)  
{  
    console.log(whichId);  
  
    //let playerElement = ge(whichId);  
  
    //if (playerElement)  
    //{  
        audioPlay("sfx_blip_001", 1.0);
```

```
// Get the position
let newPosition = positionGet(whichId);

// Update player position
ourPlayer.x = newPosition.x + 15;
ourPlayer.y = newPosition.y - 35;

// Move player to the new position
// movePlayer(whichId, ourPlayer.x,
ourPlayer.y);
    ge(activePlayer).style.left = ourPlayer.x +
"px";
    ge(activePlayer).style.top = ourPlayer.y +
"px";
    //}
}
```

//--//

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

```
// structuresMenu.js
```

```
function structuresMenu()
```

```
{  
    audioPlay("sfx_blip_001", 1.0);  
  
    if (ge("structuresMenu"))  
    {  
        removeElement("structuresMenu");  
    }
```

```
    ge('infoDiv').innerHTML = "B to Create  
Structure";
```

```
//-//
```

```
    let structureDiv = ce("div");  
    structureDiv.id = "structuresMenu";  
    structureDiv.style.position = "absolute";  
    structureDiv.style.left = ourPlayer.x + 100 +  
'px';  
    structureDiv.style.top = ourPlayer.y + 'px';  
    structureDiv.style.width = "300px";  
    structureDiv.style.height = "200px";  
    structureDiv.style.paddingLeft = '5px';
```

```
structureDiv.style.paddingRight = '5px';
structureDiv.style.zIndex = "1003";
structureDiv.style.border = "solid 5px";
structureDiv.style.borderRadius = "10px";
structureDiv.style.borderColor = "rgb(255,
255, 255)";
structureDiv.style.background = "rgb(0, 0,
0)";
structureDiv.style.fontFamily = "arial";
structureDiv.style.fontWeight = "normal";
structureDiv.style.fontSize = "medium";
structureDiv.style.color = "white";
structureDiv.style.textAlign = "center";
structureDiv.style.overflowY = 'scroll';
structureDiv.style.resize = 'both';
ba(structureDiv);

//-//
```

```
let closeButton = ce("button");
closeButton.className = "ourButton";
closeButton.style.marginBottom = "-4px";
closeButton.textContent = "X";
closeButton.onclick = function()
{
    removeElement(structureDiv.id);
}
```



```
};  
structureDiv.append(closeButton);
```

```
//-//
```

```
structureDiv.append(ce('hr'));
```

```
//-//
```

```
let titleSpan = ce("span");  
titleSpan.style.color = "rgb(255, 255, 255)";  
titleSpan.innerHTML = "<b> STRUCTURE  
OPTIONS </b>";  
structureDiv.append(titleSpan);
```

```
//-//
```

```
let startStructureSpan = ce("span");  
startStructureSpan.innerHTML = "<b> Letter  
B </b> to Start Structure Mode";  
structureDiv.append(startStructureSpan);
```

```
//-//
```

```
let placeStructureSpan = ce("span");
```

```
placeStructureSpan.innerHTML = "<b> Left  
Click </b> to Place a Structure";  
structureDiv.append(placeStructureSpan);
```

```
//-//
```

```
let gotoStructureSpan = ce("span");  
gotoStructureSpan.innerHTML = "<b> Shift +  
1 </b> to GoTo 1st Structure";  
structureDiv.append(gotoStructureSpan);
```

```
//-//
```

```
let gotoSecondStructureSpan = ce("span");  
gotoSecondStructureSpan.innerHTML =  
"<b> Shift + 2 </b> to GoTo 2nd Structure etc";  
structureDiv.append(gotoSecondStructureSpan);
```

```
//-//
```

```
let rowMenuSpan = ce("span");  
rowMenuSpan.innerHTML = "<b> Letter J  
</b><br> Creates a Row Menu<br>to GoTo Any  
Structure";
```

```
structureDiv.append(rowMenuSpan);
```

```
//-//
```

```
let editStructureSpan = ce("span");  
editStructureSpan.innerHTML = "<b> Left  
Click any Structure </b><br>to EDIT that  
Structure";  
structureDiv.append(editStructureSpan);
```

```
//-//
```

```
let deleteStructureSpan = ce("span");  
deleteStructureSpan.innerHTML = "<b>  
Delete Button </b><br>Removes All  
Structures";  
structureDiv.append(deleteStructureSpan);
```

```
//-//
```

```
let endStructureModeSpan = ce("span");  
endStructureModeSpan.innerHTML = "<b>  
Esc Button </b> to End Structure Mode";  
structureDiv.append(endStructureModeSpan);  
}
```

//--//

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024**

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

//structuresRemoveAll.js

// Remove All Structures

function structuresRemoveAll()

{

let theStructure;

for (let x = 1; x <= structuresArray.length; x+
++)

{

theStructure = "STRUCTURE" + x;

removeElement(theStructure);

}

```
// reset the array  
structuresArray = [ ];  
  
// reset the structureId to 1  
structureId = 1;  
}  
  
/--//  
  
// Dedicated to God the Father  
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
  
// structuresRow.js  
  
/*  
By Pressing the J button, the player creates a  
menu that is populated with entries from each  
Structure made.  
The player can then click on any of the entries  
to be taken to the location of that Structure.
```

The Structures GoTo Menu can also be opened from the playerMenu, by clicking on the GoTo button.

```
*/  
  
function structuresRow(theX, theY, sizeX,  
sizeY, whichArray)  
{  
    if (ge("structureRowMainDiv"))  
    {  
        removeElement("structureRowMainDiv");  
    }  
  
    //-//  
  
    let mainDiv = ce("div");  
    mainDiv.id = "structureRowMainDiv";  
    mainDiv.title = "structuresRow(theX, theY,  
sizeX, sizeY, whichArray)";  
    mainDiv.style.position = "absolute";  
    mainDiv.style.left = theX + "px";  
    mainDiv.style.top = theY + "px";  
    mainDiv.style.display = "flex";  
    mainDiv.style.flexDirection = "row";  
    mainDiv.style.gap = "15px";  
    ba(mainDiv);
```

//-//

```
let closeButton = ce("button");
closeButton.id = "closeButton";
closeButton.className = "ourButton";
closeButton.textContent = "X";
closeButton.onclick = function()
{
    ge("structureRowMainDiv").remove();
};
mainDiv.append(closeButton);
```

//-//

```
for (let i = 0; i < whichArray.length; i++) {
    if (whichArray)
    {
        let theTextArea = ce("div");
        theTextArea.style.width = sizeX + 10 +
'px';
        theTextArea.style.height = sizeY + 5 +
'px';
        theTextArea.style.zIndex = "998";
        theTextArea.style.border = "solid 2px";
```

```
theTextArea.style.borderColor =  
"rgba(0,255,255,1.0)";  
theTextArea.style.background =  
"rgb(0,0,0)";  
theTextArea.style.fontFamily = "arial";  
theTextArea.style.fontWeight =  
"normal";  
theTextArea.style.fontSize = "medium";  
theTextArea.style.color = "white";  
theTextArea.style.textAlign = "center";  
theTextArea.style.overflowY = "scroll";  
theTextArea.style.resize = "none";  
theTextArea.setAttribute("readonly",  
"true");
```

```
//-//
```

```
let goToButton = ce("button");  
goToButton.className = "ourButton";  
goToButton.textContent = "GoTo";  
goToButton.onclick = function()  
{  
    structuresGoTo(whichArray[i].id);  
};  
theTextArea.append(goToButton);
```



```
//-//
```

```
theTextArea.append(ce("br"));
```

```
//-//
```

```
let nameText = ce("div");  
nameText.innerHTML = "Name<br><b>"  
+ whichArray[i].name + "</b><br>";  
theTextArea.append(nameText);
```

```
//-//
```

```
let idText = ce("div");  
idText.innerHTML = "ID<br><b>" +  
whichArray[i].id + "</b><br>";  
theTextArea.append(idText);
```

```
//-//
```

```
let posXText = ce("div");  
posXText.textContent = "X Pos " +  
whichArray[i].posX + "<br>";  
theTextArea.append(posXText);
```

```
//-//
```

```

    let posYText = ce("div");
    posYText.textContent = "Y Pos " +
whichArray[i].posY + "<br>";
    theTextArea.append(posYText);

    //-//

```

```

    let sizeXText = ce("div");
    sizeXText.textContent = "X Size " +
whichArray[i].sizeX + "<br>";
    theTextArea.append(sizeXText);

    //-//

```

```

    let sizeYText = ce("div");
    sizeYText.textContent = "Y Size " +
whichArray[i].sizeY + "<br>";
    theTextArea.append(sizeYText);
    mainDiv.append(theTextArea);
  }
}
}

```

```

//--//

```

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

```
// updateStructures.js
```

```
// update the structuresArray
function updateStructures(idName)
{
    // strips text, leaves only digits
    // to find the array index number
    let theNumberOnly = idName.replace(/\D/g,"")
- 1;

    structureAttributes =
    {
        name:
structuresArray[theNumberOnly].name,
        id: idName,
        posX: get(idName, 'pos').x,
        posY: get(idName, 'pos').y,
        sizeX: get(idName, 'size').x,
        sizeY: get(idName, 'size').y,
```

```
    colorFill: get(idName, 'background-color'),  
    colorText: get(idName, 'color'),  
    opacity: get(idName, 'opacity'),  
    texture: get(idName, 'background-image')  
};
```

```
structuresArray.splice(theNumberOnly,1,structureAttributes);
```

```
    showStructureData();  
}
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// copyIt.js
```

```
function copyIt(whichId)  
{
```

```
let copyText = ge(whichId);
copyText.select();
copyText.setSelectionRange(0, 1000000);
document.execCommand("copy");
}
```

```
// We can copy the Text Area easily
// by clicking the Copy button
```

```
//--//
```

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

```
// listens.js
```

```
// Add a listen used currently to
// start Structure mode
```

```
function addListen(theType, theListen)
{
```

```
window.addEventListener(theType,  
theListen);  
}
```

```
// Remove a listen used currently to end  
Structure mode  
function removeListen(theType, theListen)  
{  
    window.removeEventListener(theType,  
theListen);  
}
```

```
//--//
```

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

```
// loadScript.js
```

```
function loadStructureScript(scriptName)  
{  
    // get address of current document
```

```
let theDocumentUrl = location.href;

// get first 4 letters of address
theDocumentUrl =
theDocumentUrl.substring(0,4);
//results in: file or http

// if the user is using the offline version of
the app
if (theDocumentUrl != "http")
{
    // first, we remove any currently existing
structures
    structuresRemoveAll();

    //-//

    let script = ce('script');

    script.onload = function()
    {
        structuresCreateFromFile();
    };

    script.src = "src/js/2data/" + scriptName;
```

```
//"structuresArray.js";

    document.getElementsByTagName('head')
[0].append(script);
}

else
{
    alert('Online version does NOT currently
support loading scripts');
}
}

function loadPlayerScript(scriptName)
{
    // get address of the current document
    let theDocumentUrl = location.href;

    // get first 4 letters of address
    theDocumentUrl =
theDocumentUrl.substring(0,4);
    //results in: file or http

    // if the user is using the offline version of
the app
    if (theDocumentUrl != "http")
```



```
{  
    let script = ce('script');  
  
    script.onload = function()  
    {  
        playerLoadData(activePlayer);  
    };  
  
    script.src = "src/js/2data/" + scriptName;  
    //"playerData.js";  
  
    document.getElementsByTagName('head')  
[0].append(script);  
}  
  
// else if person is using online version of the  
app  
else  
{  
    alert('Online version does NOT currently  
support loading scripts');  
}  
}  
  
/--//
```

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

```
// preventContextMenu.js
```

```
function preventContextMenu()
{
    document.addEventListener('contextmenu',
function(event)
    {
        event.preventDefault();
    });
}
```

```
//--//
```

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

```
// printing.js
```

```
function printOnlyDiv(whichDiv)
```

```
{  
    audioPlay("sfx_blip_001", 1.0);
```

```
    let mywindow = window.open("", 'PRINT',  
'height = 800, width = 1000 top = 50, left = 100');
```

```
mywindow.document.write('<style>img{width:  
75%;</style></head><body>');
```

```
    mywindow.document.write('<h4>' +  
document.title + '</h4>');
```

```
mywindow.document.write(get(whichDiv).inner  
HTML);
```

```
mywindow.print();
```

```
    // mywindow.close();
```

```
}
```

```
function printTextArea(whichTextArea)
{
    audioPlay("sfx_blip_001", 1.0);

    let mywindow = window.open("", 'PRINT',
'height = 800, width = 1000 top = 50, left = 100');

    mywindow.document.write('<style>img{width:
75%;</style></head><body>');

    mywindow.document.write('<h4>' +
document.title + '</h4>');

    mywindow.document.write(get(whichTextArea).
value);

    mywindow.print();

    // mywindow.close();
}

//--//
```

// Dedicated to God the Father

**// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024
// <https://github.com/ChristopherTopalian>
//
<https://github.com/ChristopherAndrewTopalian>**

// removeElement.js

**function removeElement(whichElement)
{
 // check if the element exists first
 if (ge(whichElement))
 {
 audioPlay("sfx_blip_001", 1.0);

 ge(whichElement).remove();
 }
}**

**// removes any element on the page,
// that has the id that was specified**

//--//

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

```
// whenLoaded.js
```

```
// When the Page Loads  
function whenLoaded()  
{  
    makeBackground();  
  
    makeDedication();  
  
    loadSounds(soundsArray);  
    loadSounds(songsArray);  
  
    makeEngineTitle();  
  
    makePlayer();  
  
    movePlayer(activePlayer, -1, 0);  
  
    makeInfoDiv();
```

```
makeInfoDivPlayer();

preventContextMenu();

let playerElement = ge(activePlayer);

if (playerElement)
{
    // url of current document
    let theDocumentUrl = location.href;

    // first 4 letters of url
    theDocumentUrl =
theDocumentUrl.substring(0, 4);

    // results in: file or http

    // if user is not using http internet version
of app, then we can load scripts from js folder
on their system
    if (theDocumentUrl != "http")
    {
        // load Structures Data

loadStructureScript("structuresArray.js");
```

```
// load Player Data  
loadPlayerScript("playerData.js");  
}  
  
structuresCreateFromFile();  
  
playerLoadData(activePlayer);  
  
makeStructureTextArea();  
  
makePlayerTextArea();  
  
// show structuresArray data in textarea  
setTimeout(function()  
{  
    showStructureData();  
}, 1000);  
  
// show playerArray data in textarea  
setTimeout(function()  
{  
    showPlayerData();  
}, 1000);  
  
// menuPlayer();
```



```
sortData();

containerLanguage();

// makeInstructions();

makeLogo();

makeBottomToolbar(bottomToolbarArray);

// makeVideo("ewDcgG8Q288");

// listen for keydown presses
window.addEventListener("keydown",
theControls, false);

// listen for keyup presses
window.addEventListener("keyup",
theControls, false);
}

gameLoop(activePlayer);
}

/--//
```

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

```
// booksPdf.js
```

```
let booksPdf =
```

```
[  
  { title:`How to Combine JS Files into One`,  
  date:`07/26/2021 12:00 AM`, url:`_ tutorials/How  
  to Combine JS Files into One.pdf`, tag:`pdf`,  
  description:`programming`},  
  
  { title:`How to Publish Our Game Engine`,  
  date:`05/12/2020 12:00 AM`, url:`_ tutorials/How  
  to Publish Our Game Engine.pdf`, tag:`pdf`,  
  description:`programming`}  
];
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// booksWebsite.js
```

```
let booksWebsite =
```

```
[  
  { title:"JavaScript for Humans",  
    date:"07/26/2021 12:00 AM",  
  
    url:"https://collegeofscripting.weebly.com/java  
script-for-humans-book.html", tag:"pdf",  
    description:"programming"},  
  
  { title:"JavaScript Bookmarklet Advanced  
Programming", date:"05/12/2020 12:00 AM",  
  
    url:"https://collegeofscripting.weebly.com/java  
script-bookmarklet-advanced-programming-  
book.html", tag:"pdf",  
    description:"programming"},  
  
  { title:"Node.js for Humans",  
    date:"06/09/2021 12:00 AM",  
  
    url:"https://collegeofscripting.weebly.com/nod  
ejs-for-humans-book.html", tag:"pdf",  
    description:"programming"},  
  
  { title:"Python Language for Humans",  
    date:"02/08/2021 12:00 AM",
```

**url:"https://collegeofscripting.weebly.com/python-language-for-humans-book.html",tag:"pdf",
description:"programming"},**

{ title:"Python Scripting in Blender for Humans", date:"06/12/2021 12:00 AM",

**url:"https://collegeofscripting.weebly.com/python-scripting-in-blender-for-humans-book.html", tag:"pdf",
description:"programming"},**

**{ title:"LibreOffice Scripting for Humans",
date:"07/31/2021 12:00 AM",**

**url:"https://collegeofscripting.weebly.com/libre-office-scripting-for-humans-book.html",
tag:"pdf", description:"programming"},**

**{ title:"C# Language for Humans Using Visual Studio", date:"03/06/2021 12:00 AM",
url:"https://collegeofscripting.weebly.com/c-sharp-language-for-humans-using-visual-studio-book.html",tag:"pdf",
description:"programming"},**

```
{ title:"Blend4Web for Humans",  
  date:"06/10/2021 12:00 AM",  
  
  url:"https://collegeofscripting.weebly.com/blen  
d4web-for-humans-book.html", tag:"pdf",  
  description:"programming"},  
  
{ title:"Papercliptronics", date:"01/08/2020  
12:00 AM",  
  
  url:"https://collegeofscripting.weebly.com/pape  
rcliptronics-electronic-circuits-book.html",  
  tag:"pdf", description:"electronics"},  
  
{ title:"Mind Simulation Science",  
  date:"09/26/2020 12:00 AM",  
  
  url:"https://collegeofscripting.weebly.com/min  
d-simulation-science-book.html", tag:"pdf",  
  description:"science"},  
  
{ title:"Human Safety Manual",  
  date:"06/12/2020 12:00 AM",  
  
  url:"https://collegeofscripting.weebly.com/hum
```

**an-safety-manual-book.html", tag:"pdf",
description:"safety"},**

**{ title:"Real Logic", date:"09/21/2020 12:00
AM",**

**url:"https://collegeofscripting.weebly.com/real-
logic-book.html", tag:"pdf",
description:"logic"},**

**{ title:"The Book of Freewill",
date:"04/19/2021 12:00 AM",**

**url:"https://collegeofscripting.weebly.com/the-
book-of-freewill-book.html", tag:"pdf",
description:"safety"},**

**{ title:"Topalian JavaScript Tools Creator,
Version 3", date:"06/04/2021 12:00 AM",**

**url:"https://collegeofscripting.weebly.com/topa-
lian-javascript-tools-creator-book.html",
tag:"pdf", description:"programming"},**

**{ title:"Topalian Game Engine Data Navigator
- Version 24", date:"11/02/2021 12:00 AM",**

```
url:"https://collegeofscripting.weebly.com/topa  
lian-game-engine-data-navigator-024.html",  
  tag:"pdf",  description:"programming"}  
];
```

```
//--//
```

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

```
let boxingVideos =  
[  
  { title:`Tyson Fury Glove Conspiracy Steps  
Back Strangely HD with Super Slow Motion &  
Zoom`, date:`02/27/2020 12:00 AM`,  
url:`https://www.youtube.com/embed/DutXMLj7  
zkQ`, tag:`boxing`, description:`furyWilder1`},  
  
  { title:`Tyson Fury Glove Conspiracy or Fact?  
You decide.`, date:`02/28/2020 12:00 AM`,
```


**url: `https://www.youtube.com/embed/OFC3U2_
dqno`, tag: `boxing`, description: `furyWilder1`},**

**{ title: `Tyson Fury Glove Conspiracy in Slow
Motion and Zoomed in Dragging Against His
Body`, date: `02/28/2020 12:00 AM`,
url: `https://www.youtube.com/embed/eHMeq04i
V7c`, tag: `boxing`, description: `furyWilder1`},**

**{ title: `Tyson Fury Glove Conspiracy
CAMERA WAS DISCONNECTED`,
date: `03/16/2020 12:00 AM`,
url: `https://www.youtube.com/embed/bYUdaPu
bdj4`, tag: `boxing`, description: `furyWilder1`},**

**{ title: `Proof that Fury Cheated #1 - 2nd
FIGHT - BEHIND THE HEAD`, date: `03/10/2020
12:00 AM`,
url: `https://www.youtube.com/embed/n_9y36vc
KeU`, tag: `boxing`, description: `furyWilder2`},**

**{ title: `Proof that Fury Cheated #2 - 2nd
FIGHT - HEAD INTO OPPONENT`,
date: `03/10/2020 12:00 AM`,
url: `https://www.youtube.com/embed/pqh5iTW
YQZI`, tag: `boxing`, description: `furyWilder2`},**

{ title:`Tyson Fury HOLDING WHILE HITTING`, date:`03/11/2020 12:00 AM`, url:`https://www.youtube.com/embed/vJ7x7lq7b0U`, tag:`boxing`, description:`furyWilder2`},

{ title:`Tyson Fury BEHIND THE HEAD 2`, date:`03/11/2020 12:00 AM`, url:`https://www.youtube.com/embed/G5A6zvTt3Uw`, tag:`boxing`, description:`furyWilder2`},

{ title:`Tyson Fury BEHIND THE HEAD 3, HOLDING WHILE HITTING 2`, date:`03/12/2020 12:00 AM`, url:`https://www.youtube.com/embed/pvph9__goil`, tag:`boxing`, description:`furyWilder2`},

{ title:`Tyson Fury Cheated with 40+ Behind the Head Hits on Wilder`, date:`03/14/2020 12:00 AM`, url:`https://www.youtube.com/embed/W2kQHmK9BMc`, tag:`boxing`, description:`furyWilder2`},

{ title:`Tyson Fury Cheating Glove Conspiracy - Flappy Glove? FIGHT TWO`,

date:`03/15/2020 12:00 AM`,
url:`https://www.youtube.com/embed/b-UyfPg4pKg`, tag:`boxing`,
description:`furyWilder2`},

{ title:`Tyson Fury CAUGHT Cheating =
CONCLUSIVE VIDEO EVIDENCE`,
date:`03/22/2020 12:00 AM`,
url:`https://www.youtube.com/embed/3ftyfqVg24g`, tag:`boxing`, description:`furyWilder2`},

{ title:`Tyson Fury Glove Controversy`,
date:`03/25/2020 12:00 AM`,
url:`https://www.youtube.com/embed/wxM6Yvvs0HU`, tag:`boxing`, description:`furyWilder2`},

{ title:`Fury Cheated and the Ref Watched`,
date:`03/28/2020 12:00 AM`,
url:`https://www.youtube.com/embed/0vq2fFpYbR0`, tag:`boxing`, description:`furyWilder2`}
];

//--//

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
// collegeBooks.js
```

```
let collegeBooks =  
[  
  { name:"Papercliptronics",  
    image:"https://m.media-amazon.com/images/I/  
51cdZyRYXVL._SX260_.jpg",  
    url:"https://www.amazon.com/Papercliptronics-  
Homemade-Electronic-Circuits-Paperclips/dp/  
1658209303/ref=sr_1_2?  
keywords=papercliptronics&qid=1582048776&  
sr=8-2"},  
  
  { name:"LSL Book", image:"https://images-  
na.ssl-images-amazon.com/images/I/41YE35ag  
tNL._SY498_BO1,204,203,200_.jpg",  
    url:"https://www.amazon.com/LSL-Scripting-  
Book-Computer-Programming/dp/  
B084DG7ZFD/ref=sr_1_fkmr0_1?  
keywords=ls+scripting&qid=1582049598&sr=8  
-1-fkmr0"},
```

```
{ name:"Topalian Game Engine Data  
Navigator", image:"https://m.media-  
amazon.com/images/I/41kybfYNoYL.jpg",  
url:"https://www.amazon.com/dp/B08JK7D67L"  
},
```

```
{ name:"JavaScript Bookmarklet Advanced  
Programming", image:"https://images-na.ssl-  
images-amazon.com/images/I/51lot64KYcL._S  
X384_BO1,204,203,200_.jpg",  
url:"https://www.amazon.com/JavaScript-  
Bookmarklet-Advanced-Programming-  
Interfaces/dp/B088JFN2BL"},
```

```
{ name:"JavaScript Video Investigator",  
image:"https://m.media-amazon.com/images/I/  
51EeQ4sNVnL._SY346_.jpg",  
url:"https://www.amazon.com/CREATING-  
JAVASCRIPT-APPLICATIONS-Video-  
Investigator/dp/B088N5G5GV"},
```

```
{ name:"JavaScript Missing Persons  
Database", image:"https://m.media-  
amazon.com/images/I/519eyD-37IL.jpg",  
url:"https://www.amazon.com/CREATING-
```

**JAVASCRIPT-APPLICATIONS-Missing-
Database/dp/B088N4XYDY"},**

**{ name:"True Artificial Intelligence",
image:"https://m.media-amazon.com/images/I/
414nBuW6oaL._SY346_.jpg",
url:"https://www.amazon.com/dp/B08975HFJK"
},**

**{ name:"Real Logic",
image:"https://m.media-amazon.com/images/I/
414T-Y2usWL.jpg",
url:"https://www.amazon.com/Real-Logic-
Evidence-Foundation-Reason-ebook/dp/
B08JQPZT6H/"},**

**{ name:"Mind Simulation Science",
image:"https://m.media-amazon.com/images/I/
41cfXtacGvL.jpg",
url:"https://www.amazon.com/Mind-Simulation-
Science-Foundation-Intelligence-ebook/dp/
B08K7L9F3Q"},**

**{ name:"Human Safety Manual",
image:"https://m.media-amazon.com/images/I/
51RzxUVo7FL.jpg",**

```
url:"https://www.amazon.com/Human-Safety-  
Manual-Christopher-Topalian-ebook/dp/  
B08B515RPL"}  
];
```

```
//--//
```

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

```
let collegeLinks =  
[  
  { name:"Javascript",  
url:"https://collegeofscripting.weebly.com/java  
script.html", },  
  
  { name:"JavaScript Game Design",  
url:"https://collegeofscripting.weebly.com/java  
script-game-design.html", },
```

```
{ name:"JavaScript Bookmarklets",  
url:"https://collegeofscripting.weebly.com/bookmarklets.html", },
```

```
{ name:"LSL Scripting Book",  
url:"https://collegeofscripting.weebly.com/lsl-scripting-book.html", },
```

```
{ name:"Logical Boolean System of A.I.",  
url:"https://collegeofscripting.weebly.com/logic-boolean-system.html", },
```

```
{ name:"Health Science",  
url:"https://collegeofscripting.weebly.com/health-science.html", },
```

```
{ name:"Solar Science",  
url:"https://collegeofscripting.weebly.com/solar-science.html", },
```

```
{ name:"Flat Earth Science",  
url:"https://collegeofscripting.weebly.com/flat-earth-science.html", },
```

```
{ name:"Why NO Video of Mars on Mars?",  
url:"https://collegeofscripting.weebly.com/mar
```


s---why-was-there-no-photographic-video-of-mars-on-mars-until-2021.html", },

{ name:"Computing",
url:"https://collegeofscripting.weebly.com/computing.html", },

{ name:"Papercliptronics",
url:"https://papercliptronics.weebly.com/", },

{ name:"Video Investigator",
url:"https://javascriptbook.weebly.com/video-investigator---video-annotation-software.html",
}
];

//--//

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>
// collegeVideos.js

```

let collegeVideos =
[
    { title:`Not Freezing Detector Circuit on a
Breadboard`, date:`09/25/2020 12:00 AM`,
url:`https://www.youtube.com/embed/wbN4MG-
J_R4`, tag:`youtubeVideo`,
description:`breadboard`},

    { title:`Heat Detector Circuit on a
Breadboard`, date:`09/26/2020 12:00 AM`,
url:`https://www.youtube.com/embed/kQ_mVFj
1u-Y`, tag:`youtubeVideo`,
description:`breadboard`},

    { title:`Evidence Viewer for Investigators`,
date:`09/15/2020 12:00 AM`,
url:`https://www.youtube.com/embed/jMBKRbG
_bXw`, tag:`youtubeVideo`,
description:`javascript`}
];

```

//--//

// Dedicated to God the Father
// All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024

```
// https://github.com/ChristopherTopalian
//
https://github.com/ChristopherAndrewTopalian
//computerSciencePdf.js
```

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

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

    { title:`JavaScript Code Volume 1`,
      url:`src/media/pdf/JavaScript Code Volume 1
by Christopher Topalian.pdf`, tag:`pdf`,
      description:`programming`},
];

//--//
```

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
// evidence.js
```

```
let evidence =
```

```
[  
  { title:`In fiery speech to Congress,  
Netanyahu vows 'total victory' in Gaza and  
denounces US protesters`, date:`2024/07/24  
9:29 PM`,  
url:`https://apnews.com/article/netanyahu-  
congress-gaza-hamas-israel-  
6ea5daf3cd1988b0ad6e874bd450f9bf`,  
info:`Natanyahu gives speech to Congress`,  
image:`https://assets.apnews.com/fa/ba/9258a7  
114f5ba5c7202aaa1bdd66/aplogo.svg`,  
tag:`fact`, description:`ww3`},
```

```
  { title:`Bill Clinton visited Jeffrey Epstein's  
private island, unsealed court documents  
suggest`, date:`07/31/2020 12:00 AM`,  
url:`https://www.foxnews.com/us/bill-clinton-  
visited-jeffrey-epsteins-private-island-
```

unsealed-court-documents-suggest`,
 info:`Witness stated that Bill Clinton Visited
 Jeffrey Epsteins island`,
 image:`https://pbs.twimg.com/media/EeN7Qo5
 X0AAhLyJ?format=png&name=900x900`,
 tag:`fact`, description:`epstein`},

{ title:`Oregon State Police end defense of
 federal courthouse in Portland, DHS reacts`,
 date:`08/14/2020 12:00 AM`,
 url:`https://www.youtube.com/embed/xtRJhUE
 S30c`, info:`youtubeVideo`,
 image:`https://collegeofscripting.weebly.com/u
 ploads/6/4/4/8/64482293/
 oregonstatepolice_orig.png`, tag:`fact`,
 description:`riots`},

{ title:`Portland protesters will no longer be
 prosecuted for non-violent charges`,
 date:`08/12/2020 12:00 AM`,
 url:`https://www.youtube.com/embed/DpIR0oD
 WhdY`, info:`youtubeVideo`,
 image:`https://collegeofscripting.weebly.com/u
 ploads/6/4/4/8/64482293/
 portlandprotestersdaprosecution_orig.png`,
 tag:`fact`, description:`riots`}

```
];
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew
```

```
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// javascriptLanguage.js
```

```
let javascriptLanguage =
```

```
[
```

```
  { category: `array`, name: `at()`, description: `single quote ' and double quote " Returns the indexed element from the array`,
```

```
  syntax: `array.at(index)`, useage: `let ourArray = ourArray.at(index);`,
```

```
  url: `https://www.w3schools.com/jsref/jsref_array_at.asp`,
```

```
  url2: `https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/at`, type: `returns the element of the given position (index) in the array.`, tag: `function`,
```

```
  language: `JavaScript`},
```

```
{ category:"array", name:"concat()",  
description:"Joins two or more arrays without  
changing the existing arrays, but instead  
returns a new array.",  
syntax:"array1.concat(array2, array3);",  
useage:"var combinedArray =  
theNotes.concat(theNews, evidence);",  
url:"https://www.w3schools.com/jsref/jsref_con  
cat_array.asp",  
url2:"https://developer.mozilla.org/en-US/docs/  
Web/JavaScript/Reference/Global_Objects/  
Array/concat", type:"returns a new array",  
tag:"function", language:"JavaScript"},
```

```
{ category:"array", name:"copyWithin()",  
description:"Copies a specific part of an array  
to a different location in the array and then  
returns that array without modifying its  
length.", syntax:"array.copyWithin(target, start,  
end);", useage:"theNotes.copyWithin(2, 0, 2);",  
url:"https://www.w3schools.com/jsref/jsref_cop  
ywithin.asp",  
url2:"https://developer.mozilla.org/en-US/docs/  
Web/JavaScript/Reference/Global_Objects/
```

```
Array/copyWithin", type:"returns the modified  
array", tag:"function", language:"JavaScript"},
```

```
];
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian  
// lindenLanguage.js
```

```
let lindenLanguage =
```

```
[
```

```
  { name:"lIAbs", description:"Tells you  
the absolute value of any integer number",  
    syntax:"lIAbs(integer number);",  
    useage:"lIAbs(-3);",  
    url:"https://wiki.secondlife.com/wiki/LIAbs",
```

```
url2:"https://collegeofscripting.weebly.com/",  
  type:"returns an integer", tag:"function",  
  category:"math", language:"Linden"},
```



```

{   name:"lIAcos", description:"Tells you
the ArcCosine of a number",
    syntax:"lIAcos(float number);",
    useage:"lIAcos(0.7);",

url:"https://wiki.secondlife.com/wiki/LIAcos",

url2:"https://collegeofscripting.weebly.com/",
    type:"returns a float, in radians",
    tag:"function", category:"math",
    language:"Linden"},

{   name:"lIAddToLandBanList",
    description:"Adds person to the land ban
list for a time. Forever is zero.",    syntax:"in
progress",    useage:"in progress",

url:"https://wiki.secondlife.com/wiki/LIAddToLa
ndBanList",

url2:"https://collegeofscripting.weebly.com/",
    type:"in progress",    tag:"function",
    category:"in progress",
    language:"Linden"},
];

```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// booksPdf.js
```

```
let booksPdf =
```

```
[
```

```
  { title:`How to Combine JS Files into One`,  
    date:`07/26/2021 12:00 AM`, url:`_ tutorials/How  
to Combine JS Files into One.p//  
mathematics.js
```

```
let mathematics =
```

```
[
```

```
  { title:`MATHEMATICS`, date:`01/01/2007  
12:00 AM`,  
    url:`https://collegeofscripting.weebly.com`,  
    image:`https://logicscience.weebly.com/upload  
s/1/3/5/6/135653355/mathematics-1.png`,
```

info:`texture`, tag:`book`, description:`math
tutorial`},

{ title:`MATHEMATICS`, date:`01/02/2007
12:00 AM`,
url:`https://collegeofscripting.weebly.com`,
image:`https://logicscience.weebly.com/upload
s/1/3/5/6/135653355/mathematics-2.png`,
info:`texture`, tag:`book`, description:`math
tutorial`},

{ title:`MATHEMATICS`, date:`01/03/2007
12:00 AM`,
url:`https://collegeofscripting.weebly.com`,
image:`https://logicscience.weebly.com/upload
s/1/3/5/6/135653355/mathematics-3.png`,
info:`texture`, tag:`book`, description:`math
tutorial`}
];

//--//

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

```
//  
https://github.com/ChristopherAndrewTopalian  
// ourVideoFiles.js
```

```
let ourVideoFiles =
```

```
[  
  { title:"Central Records Complex Opens to  
House FBI Files", type:"mp4",  
    date:"08/14/2020 12:00 AM",  
    url:"https://www.youtube.com/watch?  
v=WsOFxqB3U3A",
```

```
url2:"https://collegeofscripting.weebly.com/upl  
oads/6/4/4/8/64482293/  
central_records_complex_opens_to_house_fbi  
_files.mp4", tag:"personal",  
  description:"personal",  
  starttime:"00:00:05", endtime:"00:00:10",  
  kind:"video"},
```

```
{ title:"FBI Special Agents Your Expertise  
Fits", type:"mp4", date:"03/10/2020 12:00 AM",  
  url:"https://www.youtube.com/watch?  
v=8C973Mi8caQ",
```

```
url2:"https://collegeofscripting.weebly.com/upl
```

oads/6/4/4/8/64482293/

fbi_special_agents_your_expertise_fits.mp4",
 tag:"personal", description:"personal",
 starttime:"00:00:11", endtime:"00:00:17",
 kind:"video"},

{ title:"Video Simulation Team", type:"mp4",
 date:"04/26/2011 12:00 AM",
 url:"https://www.youtube.com/watch?
 v=84bIBZFFVYk",

url2:"https://collegeofscripting.weebly.com/upl
 oads/6/4/4/8/64482293/
 video_simulation_team.mp4", tag:"personal",
 description:"personal",
 starttime:"00:00:00", endtime:"00:00:20",
 kind:"video"},

{ title:"myGuitarMelody1", type:"mp3",
 date:"10/28/2020 12:00 AM",

url:"https://collegeofscripting.weebly.com/uplo
 ads/6/4/4/8/64482293/myguitarmelody1.mp3",

url2:"https://collegeofscripting.weebly.com/upl
 oads/6/4/4/8/64482293/myguitarmelody1.mp3",

```

tag:"personal", description:"personal",
starttime:"00:00:00", endtime:"00:00:48",
kind:"audio"}
];

//--//

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

let petitoGabrielleInvestigation =
[
    { date:`08/12/2021 12:00 PM`,
note:`<b>August 12th</b> is the same day
Petito and Laundrie had an encounter with
police in Moab, Utah.`, summary:`Police in
Utah pull Brian over with Gabrielle on August
12, 2021`, number:`0`, tag:`note`,
newsArticleOne:`https://newyork.cbslocal.com/
2021/09/18/gabby-petito-search-timeline-of-
notable-dates-and-events/`},

```

{ date:`08/25/2021 12:00 AM`, note:`She last spoke with her family on Aug. 25 from Wyoming's Grand Teton National Park`, summary:`Last time Gabrielle spoke with family`, number:`0`, tag:`note`, newsArticleOne:`https://nypost.com/2021/09/19/gabby-petito-case-police-have-obtained-cell-tracking-data/`},

{ date:`08/30/2021 12:00 PM`, note:`In the last text from her phone to family on Aug. 30, Petito indicated that she ended up in Yosemite, about 900 miles away in California - though her mother is skeptical that she actually sent the message`, summary:`Last text from Gabrielle's phone`, number:`0`, tag:`note`, newsArticleOne:`https://nypost.com/2021/09/19/gabby-petito-case-police-have-obtained-cell-tracking-data/`},

{ date:`09/01/2021 12:00 AM`, note:`Police said Laundrie returned in the van alone to his home in North Port, Florida on Sept 1.` , summary:`Brian returned to his home in the van alone`, number:`0`, tag:`note`,

newsArticleOne:`https://nypost.com/2021/09/19/gabby-petito-case-police-have-obtained-cell-tracking-data/`,

{ date:`09/11/2021 12:00 AM`,
 note:`Meanwhile. Petito's family reported her missing **Sept. 11** after she lost contact with them.`, summary:`Gabrielle reported missing on Sept 11, 2021`, number:`0`,
 tag:`note`,

newsArticleOne:`https://nypost.com/2021/09/19/gabby-petito-case-police-have-obtained-cell-tracking-data/`,

{ date:`09/18/2021 12:00 PM`,
 note:`Laundrie's family has refused to let authorities talk to him and they claimed on **Saturday (09/18/2021)** that he had gone missing himself days earlier from their home.`,
 summary:`Brian reported missing`, number:`0`,
 tag:`note`,

newsArticleOne:`https://nypost.com/2021/09/19/gabby-petito-case-police-have-obtained-cell-tracking-data/`,


```
{ date:`09/19/2021 12:00 PM`, note:`A body
consistent with the description of Gabby
Petito, the 22-year-old woman who went
missing while on a cross-country road trip with
her boyfriend, was discovered in the Bridger-
Teton National Forest in Wyoming.` ,
summary:`Body found matching description of
Gabrielle on Sept 19, 2021`, number:`0`,
tag:`note`,
newsArticleOne:`https://www.yahoo.com/gma/
unidentified-body-found-near-area-
213900218.html`},
];
```

```
//--//
```

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

```
let rickyHattonInterview =
[
```

```
{  title:"ABSOLUTE INSULT!` RICKY
HATTON RESPONDS TO DEONTAY WILDERS
CLAIMS HE HELPED TYSON FURY CHEAT!",
  date:"11/06/2020 12:00 AM",

url:"https://www.youtube.com/embed/5HWFJxT
y4R0",  tag:"youtubeVideo",
  startTime:"00:00:33", endTime:"00:00:57",
  words:"Timecode <br> 00:00:33 to 00:00:57
<br> <br> Deontay Wilder is heard speaking
on a video that Ricky Hatton is watching. <br>
Deontay Wilder is heard saying: <br> <b>So in
the first fight, when Ricky Hatton was pulling
down your glove, to put your fist in an
improper position-</b><br><br> Ricky Hatton
says in response:<br> <b>I wasn't pulling
down the glove.<br> I was pulling his gloves
on.<br> Cause Tyson flicks the punches.<br>
He flicks his shots.<br> Pulling the glove,
pulling the glove back on for him.</b>",
  name:"Ricky Hatton",
  description:"Interview", },
```

```
{  title:"`ABSOLUTE INSULT!` RICKY
HATTON RESPONDS TO DEONTAY WILDER'S
CLAIMS HE HELPED TYSON FURY CHEAT!",
```

date:"11/06/2020 12:00 AM",

url:"https://www.youtube.com/embed/5HWFJxTy4R0", tag:"youtubeVideo",

startTime:"00:06:33", endTime:"00:07:07",

words:"Timecode
 00:06:33

Ricky Hatton says
 So, that`s why Tyson, when Tyson, you know, he went, pull me glove up.

 Said pull me glove up.

 He didn`t say get that horseshoe out and stick that in it man.
 You know what I mean?

 He said pull me glove up, because he doesn`t like it as fastened right tightly round his wrist, because he, he likes to flick the shots.
 You know, he`s not, ah, you know, he doesn`t punch with his whole body, Tyson, like certain fighters.
 He likes to flick and jab and poke his jab and leave it there and that`s why he doesn`t like, he likes mobility in his wrist rather than having it locked, yeah.", name:"Ricky Hatton", description:"Interview", },

{ title:"`ABSOLUTE INSULT!` RICKY HATTON RESPONDS TO DEONTAY WILDER`S CLAIMS HE HELPED TYSON FURY CHEAT!",

date:"11/06/2020 12:00 AM",

url:"https://www.youtube.com/embed/5HWFJxTy4R0", tag:"youtubeVideo",

startTime:"00:07:19", endTime:"00:07:30",

words:"Timecode
 00:07:19 to 00:07:30

 Ricky Hatton Says:
 Yeah,
and it's like, I think um, he was, you know
Deontay was you know having a a dig at me,
you know indirectly, cause I was the individual
pulling the glove up, but I mean- ",

name:"Ricky Hatton",

description:"Interview", },

{ title:"`ABSOLUTE INSULT!` RICKY
HATTON RESPONDS TO DEONTAY WILDER'S
CLAIMS HE HELPED TYSON FURY CHEAT!",

date:"11/06/2020 12:00 AM",

url:"https://www.youtube.com/embed/5HWFJxTy4R0", tag:"youtubeVideo",

startTime:"00:24:45", endTime:"00:25:14",

words:"Timecode
 24:45 to 25:14

 Ricky Hatton says:
 I know I keep
repeating myself about the style of Tyson, you
know, he flicks his jab, so if you flick your jab,

what happens to your glove?
 It falls down
 your hand.
 Flicking the jab, flicking the
 jab, flicking the, flicking the.
 Every round I
 had to nearly pull it up.
 Pull the gloves up
 Rick.
 Pull the gloves a bit man.

 Because he`s, that`s the way his style is, he
 flicks it.
 Come on, puttin a, puttin a thing
 in his glove, honest to God.", name:"Ricky
 Hatton", description:"Interview", }
];

//--//

// Dedicated to God the Father
 // All Rights Reserved Christopher Andrew
 Topalian Copyright 2000-2024
 // <https://github.com/ChristopherTopalian>
 //
<https://github.com/ChristopherAndrewTopalian>
 // sciencelimages.js

let sciencelimages =
 [
 { title:"Electricity is the flow of electrons
 around a circuit", date:"10/07/2020 12:00 AM",

url:"http://ffden-2.phys.uaf.edu/webproj/211_fall_2016/Sterling_Stasak/Sterling_Stasak/Physics%20of%20Electricity.html",

image:"https://cdn4.explainthatstuff.com/electronflow.png", info:"website", tag:"electricity", description:"electron flow"},

{ title:"What is Electric Current? Definition & Direction of flow of current", date:"08/01/2018 12:00 AM", url:"https://www.youtube.com/watch?v=kAL17fHlv4U",

image:"https://i.ytimg.com/vi/kAL17fHlv4U/maxresdefault.jpg", info:"youtubeVideo", tag:"electricity", description:"electron flow"},

{ title:"Current Electricity and Electrical Potential", date:"09/08/2010 12:00 AM",

url:"http://paulzhang-sph3u.blogspot.com/2010/09/current-electricity-and-electrical.html",

image:"https://1.bp.blogspot.com/_kJYnXmViR

mg/TII-IIFDpsI/AAAAAAAAAAQ/fWFfGCZBiXI/s1600/electric+current.jpg", info:"website", tag:"electricity", description:"electron flow"},

{ title:"Electricity Simplified", date:"01/01/2001 12:00 AM",

url:"https://docireport.org/2013/08/17/electricity-simplified/",

image:"https://docireport.files.wordpress.com/2013/08/electron-flow.jpg", info:"website", tag:"electricity", description:"electron flow"},

{ title:"DIY - How to Build a mini electric motor", date:"07/27/2017 12:00 AM", url:"https://www.youtube.com/watch?v=Xnc4oLHZ438",

image:"https://i.ytimg.com/vi/Xnc4oLHZ438/maxresdefault.jpg", info:"youtubeVideo", tag:"electricity", description:"electric motor"},

```
{  title:"Rotation electric motor, easy
homemade",  date:"02/04/2013 12:00 AM",
  url:"https://www.youtube.com/watch?
v=WKklyuzghQg",

image:"https://i.ytimg.com/vi/WKklyuzghQg/maxresdefault.jpg",  info:"youtubeVideo",
  tag:"electricity",  description:"electric
motor"},

{  title:"How to make an electric motor
homemade",  date:"08/10/2014 12:00 AM",
  url:"https://www.youtube.com/watch?
v=kkXUmex-fy4",
  image:"https://i.ytimg.com/vi/kkXUmex-
fy4/maxresdefault.jpg",  info:"youtubeVideo",
  tag:"electricity",  description:"electric
motor"}
];
```

//--//

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


```
//  
https://github.com/ChristopherAndrewTopalian  
// socialMediaEvidence.js  
  
let socialMediaEvidence =  
[  
  { title:"Jeffrey Epstein hosted Bill Clinton on  
his private island, documents reveal",  
    date:"07/31/2020 03:05 AM",  
    url:"https://nypost.com/2020/07/31/jeffrey-  
epstein-hosted-bill-clinton-on-private-island-  
court-docs/", info:"Witness Testimony",  
  
    image:"https://nypost.com/wp-content/uploads  
/sites/2/2020/07/clinton-epstein.jpg?  
quality=90&strip=all&w=1236&h=820&crop=1",  
    tag:"twitter", description:"epstein",  
    probability:"98"},  
  
  { title:"Bill Clinton visited Jeffrey Epstein's  
private island, unsealed court documents  
suggest", date:"07/31/2020 12:00 AM",  
    url:"https://www.foxnews.com/us/bill-  
clinton-visited-jeffrey-epsteins-private-island-  
unsealed-court-documents-suggest",  
    info:"testimony",
```

image:"https://a57.foxnews.com/cf-images.us-east-1.prod.boltdns.net/v1/static/694940094001/a178ccca-ad16-4650-9e23-b520d8f99642/c0ebbece-df07-4bcf-852b-2e84318ee934/1280x720/match/931/524/image.jpg?ve=1&tl=1", tag:"twitter",
 description:"epstein", probability:"97"},
 { title:"MIT Technology Review @techreview",
 date:"07/31/2020 02:55 PM",
 url:"https://twitter.com/techreview/status/1289273594761994241", info:"The pandemic has rapidly grown the presence of QAnon and the spread of its conspiracy theories. If we stand any hope of stopping it, online platforms must do much more than fact checks and account bans.",
 image:"https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/mit_orig.jpg",
 tag:"twitter", description:"censorship",
 probability:"99"},
 { title:"Save the Children",date:"08/01/2020 12:04 AM",

url:"https://twitter.com/i/status/1289411729265393664", info:"HUMANITY IS ALIVE",

**image:"https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/save_orig.jpg",
tag:"twitter", description:"epstein",
probability:"100"},**

**{ title:"Thousands of protesters against
German coronavirus restrictions converge in
Berlin", date:"08/01/2020 08:31 AM",**

**url:"https://twitter.com/AP/status/1289539152258662402", info:"German People unite for
freedom",**

**image:"https://collegeofscripting.weebly.com/uploads/6/4/4/8/64482293/
germanprotest_orig.jpg", tag:"twitter",
description:"freedom", probability:"100"}
];**

//--//

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
// techVideos.js
```

```
let techVideos =  
[  
{ title:"How To Make a Water Pump From DC  
Motor at Home | DC Motor Ideas",  
date:"03/31/2020 12:00 AM",  
url:"https://www.youtube.com/embed/qnhReMd  
7dAk", tag:"youtubeVideo", description:"water  
pump"},  
  
{ title:"Simple Homemade DIY Water PUMP",  
date:"09/22/2020 12:00 AM",  
url:"https://www.youtube.com/embed/9fwo_VE  
MKos", tag:"youtubeVideo", description:"water  
pump"},  
  
{ title:"How To Make Water Pump 12V At  
Home/Centrifugal/WaterProof/PROPELLER BIG  
60mm/775/288W/V14", date:"08/11/2020 12:00  
AM",
```

url:"https://www.youtube.com/embed/JS828t-63HI", tag:"youtubeVideo", description:"water pump"},

{ title:"Powerful Homemade Electric Motor (Physics Explained)", date:"04/10/2016 12:00 AM",

url:"https://www.youtube.com/embed/nw1JDYY3zfA", tag:"youtubeVideo", description:"electric motor"},

{ title:"The First Electric Motor",

date:"02/07/2015 12:00 AM",

url:"https://www.youtube.com/embed/8LTP49d7KF8", tag:"youtubeVideo", description:"electric motor"},

{ title:"How to make a air pump at home",

date:"07/12/2019 12:00 AM",

url:"https://www.youtube.com/embed/KgG-GjF0c8o", tag:"youtubeVideo", description:"air pump"},

{ title:"How To Make a Mini Air Pump using DC Motor", date:"11/18/2019 12:00 AM",

url:"https://www.youtube.com/embed/7f-

**FrBc6j38", tag:"youtubeVideo",
description:"air pump"},**

**{ title:"How to Make a Matchbox Microphone",
date:"02/03/2015 12:00 AM",
url:"https://www.youtube.com/embed/yj-
wkw98j7Q", tag:"youtubeVideo",
description:"microphone"},**

**{ title:"How to Make Microphone using Pencil
Nobs and Match Box", date:"04/22/2017 12:00
AM",
url:"https://www.youtube.com/embed/3weyFhg
9jsM", tag:"youtubeVideo",
description:"microphone"},
];**

//--//

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

**https://github.com/ChristopherAndrewTopalian
// theNews.js**

let theNews =

[

{ headline:`Bill Gates Met With Jeffrey Epstein Many Times, Despite His Past`, date:`10/12/2019 12:00 AM`, url:`https://www.nytimes.com/2019/10/12/business/jeffrey-epstein-bill-gates.html`, tag:`news`, description:`epstein`},

{ headline:`Ghislaine Maxwell was a faithful lapdog to Jeffrey Epstein, wanted to marry him, royal cousin claims`, date:`07/06/2020 12:00 AM`, url:`https://www.foxnews.com/entertainment/ghislaine-maxwell-jeffrey-epstein-christina-oxenberg-prince-andrew`, tag:`news`, description:`epstein`},

{ headline:`Massachusetts Legislature Overrides Governors Veto, Allowing Minors to Obtain Abortions without Parental Consent`, date:`01/18/2021 02:23 PM`, url:`https://www.nationalreview.com/corner/massachusetts-legislature-overrides-governors-veto-allowing-minors-to-obtain-abortions-

**without-parental-consent/`, tag:`news`,
description:`crimes against humanity`},**

**{ headline:`New York Democrats bill allowing
governor to detain individuals dangerous to
public health sparks backlash`,
date:`01/03/2021 12:00 AM`,
url:`https://www.foxnews.com/politics/new-
york-bill-a416-detainment-assembly-ny`,
tag:`news`, description:`constitution
violation`},**

**{ headline:`Bidens executive order unlevels
the playing field for girls`, date:`01/21/2021
08:12 PM`,
url:`https://nypost.com/2021/01/21/bidens-
exec-order-unlevels-the-playing-field-for-girls/`,
tag:`news`, description:`executive order`},**

**{ headline:`More hand sanitizers identified
for methanol risk and added to FDA list of ones
to avoid`, date:`08/04/2020 12:11 PM`,
url:`https://www.usatoday.com/story/money/20
20/08/04/hand-sanitizer-recalls-methanol-risk-
fda-warning-coronavirus/5581182002/`,
tag:`news`, description:`hand soap`},**

{ headline:`Covid-19 mask safety: Two or three layers best to protect against virus, study finds`, date:`07/24/2020 07:39 AM`, url:`https://www.cnn.com/2020/07/23/health/covid-mask-layers-wellness/index.html`, tag:`news`, description:`mask`},

{ headline:`CDC post says to wear a mask with 2 or more layers of fabric over the nose and mouth.`, date:`01/16/2021 04:29 PM`, url:`https://twitter.com/CDCgov/status/1350555688683134985`, tag:`news`, description:`mask`},

{ headline:`Heres how you can be forced to get the Covid vaccine in the U.S.`, date:`04/01/2021 05:08 PM`, url:`https://www.cnbc.com/2021/04/01/can-i-be-forced-to-get-the-covid-vaccine.html`, tag:`news`, description:`vaccines`},

{ headline:`Smallpox virus squads and the mandatory vaccinations upheld by the Supreme Court`, date:`04/01/2021 04:36 PM`, url:`https://www.washingtonpost.com/history/2

```
021/04/01/vaccine-supreme-court-smallpox-  
covid/`, tag:`news`, description:`vaccines`  
];
```

```
//--//
```

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

```
let theNotes =  
[  
  {  note:"Here is a simple Note.",  
    date:"08/22/1998 12:00 AM",number:"1",  
    tag:"note"},  
  
  {  note:"Here is the second note.",  
    date:"08/29/2018 12:00 AM",number:"2",  
    tag:"note"},
```

```
{  note:"Here is the third note.",
  date:"09/07/2019 12:00 AM",number:"3",
  tag:"note"},
```

```
{  note:"Here is the fourth note.",
  date:"10/16/2020 10:19 AM",number:"4",
  tag:"note"}
```

```
];
```

```
//--//
```

```
// Dedicated to God the Father
```

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024
```

```
// https://github.com/ChristopherTopalian
```

```
//
```

```
https://github.com/ChristopherAndrewTopalian
```

```
// videoEvidence.js
```

```
let videoEvidence =
```

```
[
```

```
  { title:`Event 201 Pandemic Exercise:  
Highlights Reel`, date:`11/04/2019 12:00 AM`,  
url:`https://www.youtube.com/embed/AoLw-  
Q8X174`, tag:`youtubeVideo`,  
description:`exercise`},
```

{ title:`Video footage shows massive explosion in Tianjin, China`, date:`08/12/2015 12:00 AM`, url:`https://www.youtube.com/embed/qARRLog38k`, tag:`youtubeVideo`, description:`important`},

{ title:`Najaf , Iraq explosion video`, date:`08/06/2020 12:00 AM`, url:`https://www.youtube.com/embed/YARLD34YiAY`, tag:`youtubeVideo`, description:`important`},

{ title:`HD footage shows new angle of blast that rocked Beirut, Lebanon`, date:`08/09/2020 12:00 AM`, url:`https://www.youtube.com/embed/F-vGklKxGAg`, tag:`youtubeVideo`, description:`important`}
];

//--//

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
// videoInvestigator.js
```

```
let bdaBoxingComments =  
[  
{ title:"Lomachenko fans making excuses like  
the LDBC/Wilder fanboys?", date:"10/21/2020  
12:00 AM",  
url:"https://www.youtube.com/embed/JZpvDUe  
UU24", tag:"youtubeVideo",  
startTime:"00:04:20", endTime:"00:05:20",  
description:"shoulder excuse", words:"BDA  
cites the shoulder injury excuse. BDA attempts  
to excuse the excuse by citing the source of  
the report.", name:"BDA", },
```

```
{ title:"Lomachenko fans making excuses like  
the LDBC/Wilder fanboys?", date:"10/21/2020  
12:00 AM",  
url:"https://www.youtube.com/embed/JZpvDUe  
UU24", tag:"youtubeVideo",  
startTime:"00:04:44", endTime:"00:05:44",
```

description:"funny", words:"Bucho claims that he found NO channels making excuses for Lomachenko. Bucho name calls any person making excuses for Lomachenko, as PSYCHOS. Bucho says there are a lot of PSYCHOS and redefines channels to mean: MAJOR Boxing Channels.", name:"Bucho", },

**{ title:"Lomachenko fans making excuses like the LDBC/Wilder fanboys?", date:"10/21/2020 12:00 AM",
url:"https://www.youtube.com/embed/JZpvDUeUU24", tag:"youtubeVideo",
startTime:"00:06:48", endTime:"00:07:48",
description:"shoulder excuse", words:"BDA cites the shoulder injury excuse. BDA attempts to excuse the excuse by citing the source of the report. BDA also attempts to excuse the excuse by claiming that the excuse had to be announced or else Lomachenko HATERS, would call Lomachenko a LIAR, for not citing the excuse.", name:"BDA", }
];**

//--//

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

Topalian Spreadsheet to Javascript 025


```
<!-- Dedicated to God the Father -->
<!-- All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024 -->
<!-- https://github.com/ChristopherTopalian -->
<!--
https://github.com/ChristopherAndrewTopalian
-->
<!--
Topalian_Spreadsheet_To_Javascript_025.html
-->
<!-- Version 025 - (2024-08-10) -->
```

```
<html>
```

```
<head>
```

```
<title> Topalian Spreadsheet To Javascript
```

```
</title>
```

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

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

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

```
<script src =  
'js/4processInput/processSpreadsheetData.js'>  
</script>
```

```
<script src = 'js/5copy/copyIt.js'></script>
```

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

```
</head>
```

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

```
</body>
```

```
</html>
```

```
/* style001.css */
```

```
body
```

```
{  
    background: rgb(30, 30, 30);  
}
```

```
#pasteDataHere
```

```
{  
    height: 100px;  
    width: 600px;  
    box-sizing: border-box;  
    background-color: rgb(0, 0, 0);  
    border-radius: 6px;  
    border: 3px solid rgb(0,0,0);  
    border-color: rgb(255, 255, 255);  
    cursor:pointer;  
    color:rgb(255, 255, 255);  
    font-size: 14px;  
    padding: 9px 9px;  
    text-decoration: none;  
}
```

```
#displayData
```

```
{  
    height: 100px;
```

```
width: 600px;
padding: 9px 9px;
background-color: rgb(0, 0, 0);
border-radius: 6px;
border: 3px solid rgb(0,0,0);
border-color: black;
cursor: pointer;
color:rgb(255, 255, 255);
font-size: 14px;
text-decoration: none;
box-sizing: border-box;
}
```

#displayContainer

```
{
  display: flex;
  flex-direction: column;
}
```

#copyButton

```
{
  width: 100px;
  background-color: rgb(0, 77, 129);
  font-family: exo black;
  font-size: 26px;
  color: white;
}
```

```
border-radius: 7px;  
}
```

```
#copyButton:hover  
{  
background-color: rgb(36, 135, 201);  
}
```

```
#appTitle  
{  
font-family: Arial;  
font-size: 30px;  
font-weight: bold;  
color: white;  
}
```

```
#instructions  
{  
font-family: tahoma;  
font-size: 26px;  
color: white;  
}
```

```
/* Dedicated to God the Father */  
/* All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024 */
```

`/* https://github.com/ChristopherTopalian */`

`/*`

`https://github.com/ChristopherAndrewTopalian`

`*/`

// 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 textClear(theId)  
{  
    ge(theId).innerHTML = "";  
}
```

```
//--//
```

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


// makeInterface.js

```
function makeInterface()
{
    // mainContainer
    let mainContainer = ce('div');
    mainContainer.id = 'mainContainer';
    mainContainer.style.display = 'flex';
    mainContainer.style.flexDirection = 'column';
    ba(mainContainer)

    //-//

    // appTitle
    let appTitle = ce('div');
    appTitle.id = 'appTitle';
    appTitle.textContent = 'Topalian Spreadsheet
To Javascript Converter';
    mainContainer.append(appTitle);

    //-//

    let instructionsDiv = ce('div');
    instructionsDiv.id = 'instructions';
    instructionsDiv.textContent = 'Paste Data
Into the Text Area Below';
```

```
instructionsDiv.style.marginBottom = '5px';  
mainContainer.append(instructionsDiv);
```

```
//-//
```

```
let hrOverInput = ce('hr');  
hrOverInput.style.width = '100%';  
mainContainer.append(hrOverInput);
```

```
//-//
```

```
let inputDiv = ce('textarea');  
inputDiv.id = 'pasteDataHere';  
inputDiv.placeholder = 'Paste Here';  
inputDiv.oninput = function()  
{  
    processSpreadsheetData();  
};  
mainContainer.append(inputDiv);
```

```
//-//
```

```
let hrUnderInput = ce('hr');  
hrUnderInput.style.width = '100%';  
mainContainer.append(hrUnderInput);
```

//-//

```
let outputDiv = ce('textarea');  
outputDiv.id = 'displayData';  
outputDiv.style.marginBottom = '7px';  
mainContainer.append(outputDiv);
```

//-//

```
let copyButton = ce('button');  
copyButton.id = 'copyButton';  
copyButton.onclick = function()  
{  
    copyIt();  
};  
copyButton.textContent = 'Copy';  
mainContainer.append(copyButton);  
}
```

//--//

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

//

<https://github.com/ChristopherAndrewTopalian>

// processSpreadsheetData.js

```
function processSpreadsheetData()
{
    let ourData = ge("pasteDataHere").value;
    ourData.trim();

    for (let j = 0; j < ourData.length; j++)
    {
        // /g means do a global replacement,
        meaning all occurrences

        // /s means any spaces found

        // Trim all beginning white space when it
        finds "space
        ourData = ourData.replace(/^\s/g, '');

        // Trim all ending white space when it finds
        space"
        ourData = ourData.replace(/\s$/, '');
    }

    let test =
    ge("displayData").innerHTML=ourData.split(/},/
    g).join('}', '+ \n');
```

}

//--//

// Dedicated to God the Father

// All Rights Reserved Christopher Andrew

Topalian Copyright 2000-2024

// <https://github.com/ChristopherTopalian>

//

<https://github.com/ChristopherAndrewTopalian>

```
// copylt.js
```

```
function copylt()  
{  
    let copyText = ge("displayData");  
    copyText.select();  
    copyText.setSelectionRange(0, 1000000);  
    document.execCommand("copy");  
  
    alert("Successfully COPIED. \nNow PASTE it  
in TopalianData.js");  
}
```

```
//--//
```

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

```
// whenLoaded.js
```

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

```
//--//
```

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


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.

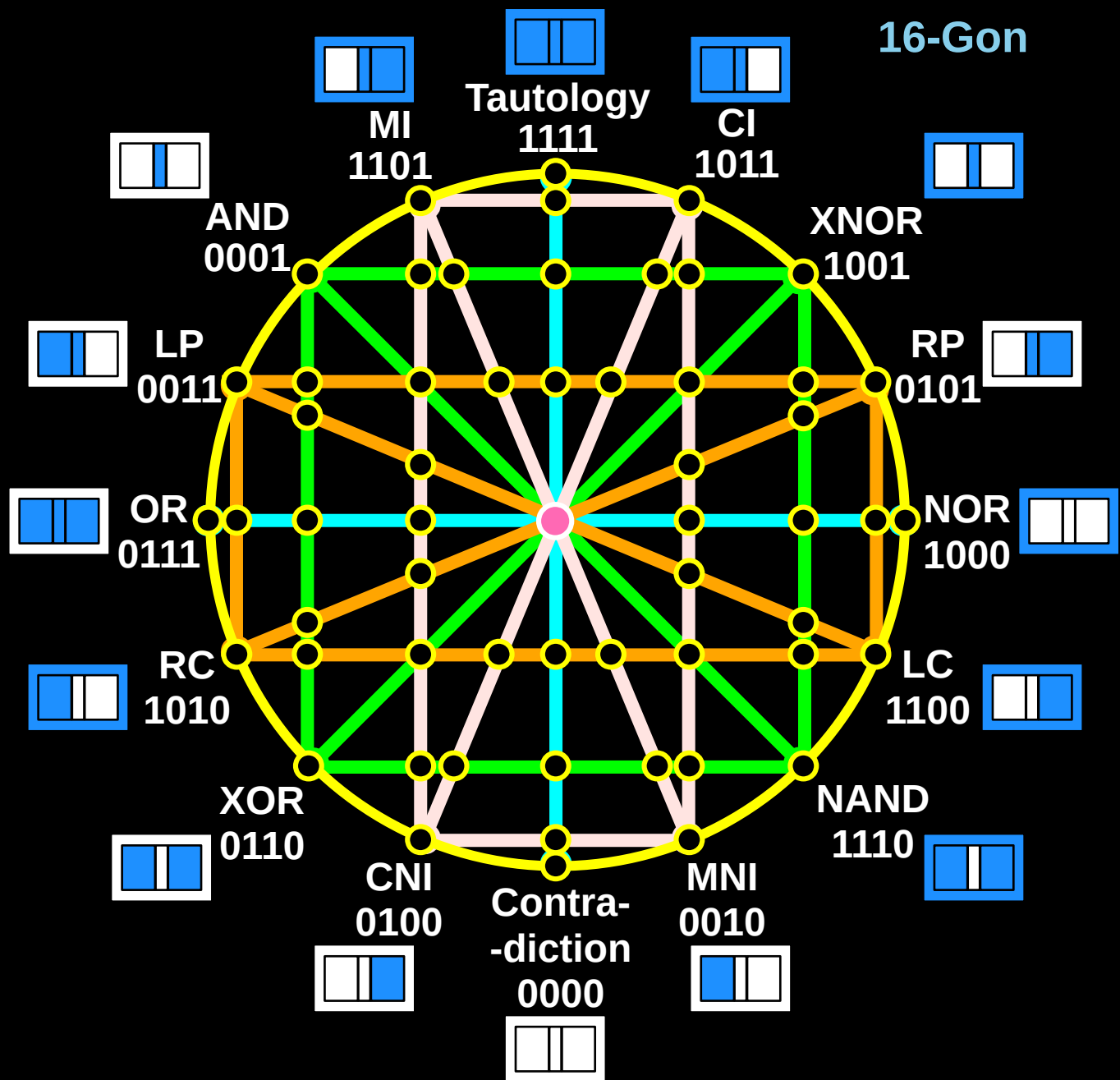
We can do the same thing for .html files, where we change the file type it will be combining to .html and it will combine all .html files into one main.html file.

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**](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