

Topalian JavaScript Genetics Navigator

by
Christopher Andrew Topalian

All Rights Reserved
Copyright 2000-2024

Dedicated to God the Father

Topalian JavaScript Genetics Navigator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Genetics_Navigator/Topalian

Names Only

Name
Reverse

BRCA1 CFTR
EGFR TP53

BRCA1

Names All
Reverse All

Search

Name: BRCA1
Sequence: AGCTTGAC...
Function: DNA repair
Mutations: 185delAG, 5382insC
Chromosome: 17

A G
C T
T G
A C
.
.

185delAG 5382insC

Topalian JavaScript Genetics Navigator

6:48 AM
11/6/2024

Topalian JavaScript Genetics Navigator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Genetics_Navigator/Topalian

Names Only

Name
Reverse

BRCA1 CFTR
EGFR TP53

Names All
Reverse All

Search

Name: CFTR
Sequence: TTGTCATG...
Function: Regulates chloride channels
Mutations: Δ F508, G551D
Chromosome: 7

TTGTCATG... Δ F508 G551D

Topalian JavaScript Genetics Navigator

6:49 AM
11/6/2024

Topalian JavaScript Genetics Navigator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Genetics_Navigator/Topalian

Names Only

Name
Reverse

BRCA1 CFTR
EGFR TP53

EGFR

Names All
Reverse All

Search

Name: EGFR
Sequence: GGCTGTAA...
Function: Cell growth and division
Mutations: L858R, T790M
Chromosome: 7

GGCTGTAA... L858R T790M

Topalian JavaScript Genetics Navigator

G G C T G T A A . . . L858R T790M

G G
C T
G T
A A
. .
. .

6:49 AM
11/6/2024

Topalian JavaScript Genetics Navigator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Genetics_Navigator/Topalian

Names Only

Name
Reverse

BRCA1 CFTR
EGFR TP53

TP53

Names All
Reverse All

Search

Name: TP53
Sequence: AGGTGGAG...
Function: Tumor suppressor
Mutations: R175H, R248Q, R273H
Chromosome: 17

A G
G T
G G
A G
. .
. .

A G G T G G A G . . . R175H R248Q R273H

Topalian JavaScript Genetics Navigator

6:50 AM
11/6/2024

Topalian JavaScript Genetics Navigator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Genetics_Navigator/Topalian ★ 100%

Names Only
Name
Reverse

BRCA1 CFTR
EGFR TP53

Topalian JavaScript Genetics Navigator

Gene Name...
Search

Names All
Reverse All

Name: **BRCA1**
Function: DNA repair
Chromosome: 17
Mutations: 185delAG, 5382insC
Sequence: AGCTTGAC.....

Name: **CFTR**
Function: Regulates chloride channels
Chromosome: 7
Mutations: ΔF508, G551D
Sequence: TTCTGATC

6:50 AM
11/6/2024

Topalian JavaScript Genetics Navigator
PLAYING

file:///D:/_1Code/0_JS_Publish/Topalian_JavaScript_Genetics_Navigator/Topalian

Names Only

Name
Reverse

BRCA1 CFTR
EGFR TP53

BRCA1

Names All
Reverse All

Search

Name: **BRCA1**
Function: DNA repair
Chromosome: 17
Mutations: 185delAG, 5382insC
Sequence: AGCTTGAC.....

A G
C T
T G
A C
.
.

AGCTTGAC... 185delAG 5382insC

Topalian JavaScript Genetics Navigator

6:51 AM
11/6/2024


```
<!-- Dedicated to God the Father -->
<!-- All Rights Reserved Christopher Andrew
Topalian Copyright 2000-2024 -->
<!-- https://github.com/ChristopherTopalian -->
<!--
https://github.com/ChristopherAndrewTopalian
-->
<!--
Topalian_JavaScript_Genetics_Navigator_(Onli
ne_Version).html -->
<!-- Version 001 - (2024-11-06) -->
```

```
<html>
```

```
<head>
```

```
<title> Topalian JavaScript Genetics Navigator
```

```
</title>
```

```
<style>
```

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

```
body
```

```
{
```

```
    background-color: rgb(30, 30, 30);
```

```
    font-family: Arial;
```

```
    color: rgb(200, 200, 200);
```

```
}
```

```
a
```

```
{
```

```
  color: rgb(100, 100, 100);
```

```
}
```

```
hr
```

```
{
```

```
  margin-top: 5px;
```

```
}
```

```
button
```

```
{
```

```
  background-color: rgb(0, 77, 129);
```

```
  margin-left: 2px;
```

```
  margin-right: 2px;
```

```
  margin-top: 1px;
```

```
  margin-bottom: 1px;
```

```
  border-style: solid;
```

```
  border-width: 1px;
```

```
  border-color: rgb(100, 100, 255);
```

```
  border-radius: 4px;
```

```
  padding-left: 7px;
```

```
  padding-right: 7px;
```

```
  padding-top: 1px;
```

```
padding-bottom: 1px;
opacity: 1.0;
font-size: 14px;
font-weight: bold;
color: rgb(255, 255, 255);
text-align: center;
cursor: pointer;
}

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

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

.result
{
    background-color: rgb(0, 0, 0);
    margin-top: 10px;
    padding: 10px;
```

```
border: 1px solid rgb(200, 200, 200);  
color: rgb(255, 255, 255);  
}
```

```
.panel  
{  
    background-color: rgb(0, 0, 0);  
    border: solid 1px rgb(100, 100, 100);  
    overflow-y: scroll;  
}
```

```
.nucleotideSpan  
{  
    margin-right: 2px;  
    border: solid 1px rgb(100, 100, 100);  
    border-radius: 8px;  
    padding-left: 4px;  
    padding-right: 4px;  
    padding-top: 0.5px;  
    padding-bottom: 0.5px;  
    font-weight: bold;  
    color: rgb(255, 255, 255);  
    text-align: center;  
    line-height: 30px;  
    text-shadow: 1px 1px 2px rgb(0, 0, 0);  
    cursor: pointer;  
}
```

```
}
```

```
.nucleotideSpan:hover
```

```
{
```

```
    position: relative;
```

```
    top: 1px;
```

```
}
```

```
.nucleotideDiv1
```

```
{
```

```
    padding: 5px;
```

```
    border: 1px solid rgb(255, 255, 255);
```

```
    border-radius: 8px;
```

```
    font-weight: bold;
```

```
    text-align: center;
```

```
    text-shadow: 1px 1px 2px rgb(0, 0, 0);
```

```
    cursor: pointer;
```

```
}
```

```
.nucleotideDiv1:hover
```

```
{
```

```
    position: relative;
```

```
    top: 1px;
```

```
}
```

```
.nucleotideDiv2
```

```
{  
  border: 1px solid rgb(255, 255, 255);  
  border-radius: 8px;  
  padding: 5px;  
  font-weight: bold;  
  text-align: center;  
  text-shadow: 1px 1px 2px rgb(0, 0, 0);  
  cursor: pointer;  
}
```

.nucleotideDiv2: hover

```
{  
  position: relative;  
  top: 1px;  
}
```

.mutationMarker

```
{  
  padding: 5px;  
  border: 1px solid rgb(100, 100, 100);  
  color: rgb(200, 200, 200);  
  text-align: center;  
  cursor: pointer;  
}
```

.mutationMarker: hover

```
{  
  position: relative;  
  top: 1px;  
}
```

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

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

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

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

```
/*
```

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

```
/*
```

```
https://sites.google.com/view/CollegeOfScripti  
ng */
```

```
</style>
```

```
<script>
```

```
// shortcuts.js
```

```
function ge(whichId)
```

```
{
```

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

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

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

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



```
    return result;  
}
```

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

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

```
// worldVariables.js
```

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

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

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

```
// genes.js
```

```
let genes = [  
  {  
    name: `BRCA1`,  
    sequence: `AGCTTGAC...`,  
    function: `DNA repair`,  
    mutations: ['185delAG', '5382insC'],  
    chromosome: 17  
  },  
  
  {  
    name: `TP53`,  
    sequence: `AGGTGGAG...`,  
    function: `Tumor suppressor`,  
    mutations: ['R175H', 'R248Q', 'R273H'],  
    chromosome: 17  
  },  
  
  {  
    name: `CFTR`,  
    sequence: `TTGTCATG...`,
```

```
function: `Regulates chloride channels`,
mutations: ['ΔF508', 'G551D'],
chromosome: 7
},

{
  name: `EGFR`,
  sequence: `GGCTGTAA...`,
  function: `Cell growth and division`,
  mutations: ['L858R', 'T790M'],
  chromosome: 7
}
];

//--//
```

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

```
// makeDNAHorizontalDiagram.js

// make and display gene sequence
function
makeDNAHorizontalDiagram(geneName)
{
    // find gene data by name
    let gene = null;

    for (let i = 0; i < genes.length; i++)
    {
        if (genes[i].name === geneName)
        {
            gene = genes[i];
            break;
        }
    }

    if (!gene)
    {
        console.error('Gene not found!');
        return;
    }

    // get panel element
    let panelBottom = ge('panelBottom');
```

```
// clear any previous content
panelBottom.innerHTML = "";

// make a container for gene sequence
let geneContainer = ce('div');
geneContainer.style.overflowX = 'scroll';
geneContainer.style.whiteSpace = 'nowrap';
geneContainer.style.padding = '10px';
geneContainer.style.border = '1px solid
rgb(204, 204, 204)';

// make nucleotide sequence display
for (let i = 0; i < gene.sequence.length; i++)
{
    // current nucleotide
    let nucleotide = gene.sequence[i];

    // make a span for each nucleotide
    let nucleotideSpan = ce('span');
    nucleotideSpan.textContent = nucleotide;
    nucleotideSpan.className =
'nucleotideSpan';
    nucleotideSpan.style.width = '15px';
    nucleotideSpan.style.height = '30px';
```

```
nucleotideSpan.onmouseover = function()
{
    hoverSound2();
};

// color code nucleotides
if (nucleotide === 'A')
{
    nucleotideSpan.style.backgroundColor
= 'green';
}
else if (nucleotide === 'T')
{
    nucleotideSpan.style.backgroundColor
= 'red';
}
else if (nucleotide === 'G')
{
    nucleotideSpan.style.backgroundColor
= 'blue';
}
else if (nucleotide === 'C')
{
    nucleotideSpan.style.backgroundColor
= 'yellow';
}
```

```
    else
    {
        nucleotideSpan.style.backgroundColor
= 'rgb(100, 100, 100)';
        // for unknown characters
    }

    geneContainer.append(nucleotideSpan);
}

//--//

// make mutation markers
for (let j = 0; j < gene.mutations.length; j++)
{
    let mutationMarker = ce('span');
    mutationMarker.textContent =
gene.mutations[j];
    mutationMarker.className =
'mutationMarker';
    mutationMarker.style.backgroundColor =
'rgb(0, 50, 50)';
    mutationMarker.style.padding = '2px 5px';
    mutationMarker.style.marginLeft = '10px';
    mutationMarker.style.border = 'solid 1px
rgb(200, 200, 200)';
```

```
mutationMarker.style.borderRadius =  
'8px';  
mutationMarker.style.cursor = 'pointer';  
mutationMarker.title = 'Mutation: ' +  
gene.mutations[j];  
  
mutationMarker.onmouseover = function()  
{  
    hoverSound2();  
};  
  
geneContainer.append(mutationMarker);  
}  
  
panelBottom.append(geneContainer);  
}  
  
/--//
```

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



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

```
// makeDNAVerticalDiagram.js
```

```
function makeDNAVerticalDiagram(sequence)  
{  
    let panelRight = ge('panelRight');  
    // clear previous content  
    panelRight.innerHTML = "";  
  
    let container =  
document.createElement('div');  
    container.style.display = 'grid';  
    // two equal-width columns  
    container.style.gridTemplateColumns = '1fr  
1fr';  
    container.style.gridGap = '5px';  
    container.style.marginLeft = 4 + 'px';  
    container.style.marginRight = 4 + 'px';  
    container.style.color = 'rgb(255, 255, 255)';  
  
    panelRight.appendChild(container);  
  
    // color mapping
```

```
let nucleotideColors = {  
  A: { color: 'green' },  
  T: { color: 'red' },  
  C: { color: 'yellow' },  
  G: { color: 'blue' }  
};  
  
// loop through sequence in steps of 2  
for (let i = 0; i < sequence.length; i += 2)  
{  
  // get current pair of nucleotides  
  let nucleotide1 = sequence[i];  
  
  // handle odd-length sequences  
  let nucleotide2 = sequence[i + 1] || "";  
  
  // make and style divs for nucleotides  
  let nucleotideDiv1 = ce('div');  
  nucleotideDiv1.className =  
'nucleotideDiv1';  
  nucleotideDiv1.style.backgroundColor =  
nucleotideColors[nucleotide1] ?  
nucleotideColors[nucleotide1].color : 'rgb(100,  
100, 100)'; // set bg color  
  nucleotideDiv1.textContent = nucleotide1;  
  nucleotideDiv1.onmouseover = function()
```

```

{
    hoverSound2();
};
container.append(nucleotideDiv1);

```

```
//-//
```

```

let nucleotideDiv2 = ce('div');
nucleotideDiv2.className =
'nucleotideDiv2';
    nucleotideDiv2.style.backgroundColor =
nucleotideColors[nucleotide2] ?
nucleotideColors[nucleotide2].color : 'rgb(100,
100, 100)'; // set bg color
    nucleotideDiv2.textContent = nucleotide2;
nucleotideDiv2.onmouseover = function()
{
    hoverSound2();
};
container.append(nucleotideDiv2);
}
}

```

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

// Dedicated to God the Father

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
//  
https://sites.google.com/view/CollegeOfScripti  
ng
```

```
// makeInterface.js
```

```
function makeInterface()  
{  
    let panelWidth = '150px';  
    let panelHeight = '60px';  
  
    //-//  
  
    let mainDiv = ce('div');  
    mainDiv.id = 'mainDiv';  
    ba(mainDiv);  
  
    //-//  
  
    // left panel  
    let panelLeft = ce('div');
```

```
panelLeft.id = 'panelLeft';  
panelLeft.className = 'panel';  
panelLeft.style.position = 'absolute';  
panelLeft.style.left = '0px';  
panelLeft.style.top = '0px';  
panelLeft.style.bottom = '0px';  
panelLeft.style.width = panelWidth;  
mainDiv.append(panelLeft);
```

```
//-//
```

```
// right panel
```

```
let panelRight = ce('div');  
panelRight.id = 'panelRight';  
panelRight.className = 'panel';  
panelRight.style.position = 'absolute';  
panelRight.style.right = '0px';  
panelRight.style.top = '0px';  
panelRight.style.bottom = '0px';  
panelRight.style.width = panelWidth;  
mainDiv.append(panelRight);
```

```
//-//
```

```
// top panel
```

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

```
panelTop.id = 'panelTop';
panelTop.className = 'panel';
panelTop.style.position = 'absolute';
// starts after left panel
panelTop.style.left = panelWidth;
// ends before right panel
panelTop.style.right = panelWidth;
panelTop.style.top = '0px';
panelTop.style.height = panelHeight;
panelTop.style.display = 'flex';
panelTop.style.flexDirection = 'row';
mainDiv.append(panelTop);
```

```
//-//
```

```
// bottom panel
let panelBottom = ce('div');
panelBottom.id = 'panelBottom';
panelBottom.className = 'panel';
panelBottom.style.position = 'absolute';
// starts after left panel
panelBottom.style.left = panelWidth;
// ends before right panel
panelBottom.style.right = panelWidth;
panelBottom.style.bottom = '0px';
panelBottom.style.height = panelHeight;
```

```
mainDiv.append(panelBottom);
```

```
//-//
```

```
// center panel
```

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

```
panelCenter.id = 'panelCenter';
```

```
panelCenter.className = 'panel';
```

```
panelCenter.style.position = 'absolute';
```

```
// starts after left panel
```

```
panelCenter.style.left = panelWidth;
```

```
// ends before right panel
```

```
panelCenter.style.right = panelWidth;
```

```
// starts after top panel
```

```
panelCenter.style.top = panelHeight;
```

```
// ends before bottom panel
```

```
panelCenter.style.bottom = panelHeight;
```

```
panelCenter.style.backgroundColor =
```

```
'rgb(100, 100, 100)';
```

```
mainDiv.append(panelCenter);
```

```
}
```

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

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

```
// All Rights Reserved Christopher Andrew  
Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian  
//  
https://sites.google.com/view/CollegeOfScripti  
ng
```

```
// makeLabel.js
```

```
function makeLabel(whichElement, whichText)  
{  
    let theId = ge(whichElement);  
  
    let theLabel = ce('label');  
    theLabel.setAttribute('for', theId);  
    theLabel.textContent = whichText;  
    theLabel.style.fontSize = '15px';  
    theLabel.style.lineHeight = '14px';  
    theLabel.style.textAlign = 'center';  
    theLabel.style.display = 'block';  
    theLabel.style.marginTop = '4px';  
    theLabel.style.marginBottom = '4px';  
    return theLabel;  
}
```



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

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

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

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

```
//
```

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

```
//
```

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

```
// makeNameButtons.js
```

```
function makeNameButtons()
```

```
{
```

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

```
    subDiv.id = 'subDivLeft';
```

```
    subDiv.title = 'subDiv';
```

```
    ge('panelLeft').append(subDiv);
```

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

```
// label Names Only
```

```
subDiv.append(makeLabel(subDiv, 'Names  
Only'));
```

```
//-//
```

```
let namesContainer = ce('div');  
namesContainer.id = 'namesContainer';  
namesContainer.title = 'namesContainer';  
namesContainer.style.overflowY = 'scroll';  
namesContainer.style.border = 'solid 1px  
rgb(100, 100, 100)';  
namesContainer.style.borderRadius = '8px';  
namesContainer.style.display = 'flex';  
namesContainer.style.flexDirection =  
'column';  
subDiv.append(namesContainer);
```

```
//-//
```

```
let nameButton = ce('button');  
nameButton.id = 'nameButton';  
nameButton.title = 'nameButton';  
nameButton.textContent = 'Name';  
nameButton.onmouseover = function()  
{  
    hoverSound();
```

```
};  
nameButton.onclick = function()  
{  
    clickSound();  
  
    // sort by name in ascending order  
    genes.sort(function(a, b)  
    {  
        return a.name.localeCompare(b.name);  
    });  
  
    if (ge('tempDiv'))  
    {  
        ge('tempDiv').remove();  
    }  
  
    let tempDiv = ce('div');  
    tempDiv.id = 'tempDiv';  
    tempDiv.title = 'tempDiv';  
    tempDiv.style.overflowY = 'scroll';  
    tempDiv.style.height = '200px';  
    ge('panelLeft').append(tempDiv);  
  
    //-//  
  
    for (let x = 0; x < genes.length; x++)
```

```

{
    let buttonHere = ce('button');
    buttonHere.textContent =
genes[x].name;
    buttonHere.id = genes[x].name;
    buttonHere.title = genes[x].name;
    buttonHere.onmouseover = function()
    {
        hoverSound();
    };
    buttonHere.onclick = function()
    {
        clickSound();
    }

    makeDNAHorizontalDiagram(genes[x].name);

    makeDNAVerticalDiagram(genes[x].sequence);

    ge('results').innerHTML =
    'Name: ' + genes[x].name + '<br>' +
    'Sequence: ' + genes[x].sequence +
    '<br>' +
    'Function: ' + genes[x].function +
    '<br>' +
    'Mutations: ' +
    genes[x].mutations.join(', ') + '<br>' +

```

```

        'Chromosome: ' +
genes[x].chromosome + '<br>';

        // search box value update
        ge('searchGene').value =
genes[x].name;
    };
    tempDiv.append(buttonHere);
}
};
ge('namesContainer').append(nameButton);

// -//

```

```

let reverseButton = ce('button');
reverseButton.id = 'reverseButton';
reverseButton.title = 'reverseButton';
reverseButton.textContent = 'Reverse';
reverseButton.onmouseover = function()
{
    hoverSound();
};
reverseButton.onclick = function()
{
    clickSound();
}

```

```
genes.reverse();

if (ge('tempDiv'))
{
    ge('tempDiv').remove();
}

let tempDiv = ce('div');
tempDiv.id = 'tempDiv';
tempDiv.title = 'tempDiv';
ge('panelLeft').append(tempDiv);

for(let x = 0; x < genes.length; x++)
{
    let buttonHere = ce('button');
    buttonHere.textContent =
genes[x].name;
    buttonHere.title = genes[x].name;
    buttonHere.onmouseover = function()
    {
        hoverSound();
    };
    buttonHere.onclick = function()
    {
        clickSound();
    }
}
```

```
makeDNAHorizontalDiagram(genes[x].name);
```

```
makeDNAVerticalDiagram(genes[x].sequence);
```

```

    ge('results').innerHTML =
    'Name: ' + genes[x].name + '<br>' +
    'Sequence: ' + genes[x].sequence +
    '<br>' +
    'Function: ' + genes[x].function +
    '<br>' +
    'Mutations: ' + genes[x].mutations +
    '<br>' +
    'Chromosome: ' +
    genes[x].chromosome + '<br>';

```

```

    // search box value update
    ge('searchGene').value =
    genes[x].name;
    };
    tempDiv.append(buttonHere);
  }
};

```

```
ge('namesContainer').append(reverseButton);
```

```
    ge('panelLeft').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  
//  
// https://sites.google.com/view/CollegeOfScripti  
// ng
```

```
// makeNamesAllButtons.js
```

```
function makeNamesAllButtons()  
{  
    let subDiv = ce('div');  
    ge('panelTop').append(subDiv);  
  
    //--//
```

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



```
namesAllContainer.id = 'namesAllContainer';
namesAllContainer.style.overflowY = 'scroll';
namesAllContainer.style.border = 'solid 1px
rgb(100, 100, 100)';
namesAllContainer.style.borderRadius =
'8px';
namesAllContainer.style.display = 'flex';
namesAllContainer.style.flexDirection =
'column';
subDiv.append(namesAllContainer);

// - //
```

```
let nameAllButton = ce('button');
nameAllButton.id = 'nameAllButton';
nameAllButton.textContent = 'Names All';
nameAllButton.onmouseover = function()
{
    hoverSound();
};
nameAllButton.onclick = function()
{
    clickSound();

    // sort by name in ascending order
    genes.sort(function(a, b)
```

```
{  
    return a.name.localeCompare(b.name);  
});  
  
displayResults(genes);  
};  
namesAllContainer.append(nameAllButton);  
  
//-//
```

```
let reverseAllButton = ce('button');  
reverseAllButton.id = 'reverseAllButton';  
reverseAllButton.textContent = 'Reverse All';  
reverseAllButton.onmouseover = function()  
{  
    hoverSound();  
};  
reverseAllButton.onclick = function()  
{  
    clickSound();  
  
    genes.reverse();  
  
    displayResults(genes);  
};
```

```
namesAllContainer.append(reverseAllButton);  
}
```

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

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

```
// makeNamesTextbox.js
```

```
function makeNamesTextbox()  
{  
    let theTextarea = ce('div');  
    theTextarea.id = 'namesTextbox';  
    theTextarea.style.position = 'relative';  
    theTextarea.style.left = cr('mainDiv').right +  
'500px';
```

```
theTextarea.style.top = cr('mainDiv').bottom  
+ '1500px';  
ba(theTextarea);  
}
```

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

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

```
// makeResultsDiv.js
```

```
function makeResultsDiv()  
{  
    let subDiv = ce('div');  
    ge('panelCenter').append(subDiv);  
  
    //-//
```

```
let resultsDiv = ce('div');
resultsDiv.id = 'results';
resultsDiv.className = 'result';
subDiv.append(resultsDiv);
}
```

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

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

```
// makeSearchBox.js
```

```
function makeSearchBox()
{
    let subDiv = ce('div');
    ge('panelTop').append(subDiv);
```

```
//-//
```

```
let inputField = ce('input');
inputField.type = 'text';
inputField.id = 'searchGene';
inputField.style.paddingLeft = '5px';
inputField.style.paddingRight = '5px';
inputField.style.backgroundColor = 'rgb(0, 0, 0)';
inputField.style.color = 'rgb(255, 255, 255)';
inputField.style.marginBottom = '2px';
inputField.style.borderRadius = '8px';
inputField.style.width = '100%';
inputField.placeholder = 'Gene Name...';
inputField.onmouseover = function()
{
    hoverSound();
};
inputField.onclick = function()
{
    clickSound();
};
subDiv.append(inputField);

//-
```

```
subDiv.append(ce('label').textContent = '');
```

```
//-//
```

```
let searchButton = ce('button');  
searchButton.textContent = 'Search';  
searchButton.onmouseover = function()  
{  
    hoverSound();  
};  
searchButton.onclick = function()  
{  
    clickSound();  
  
    filterGenes(genes);
```

```
makeDNAHorizontalDiagram(inputField.value);
```

```
makeDNAVerticalDiagram(genes[findIndexByName(genes, inputField.value)].sequence);  
};  
subDiv.append(searchButton);  
}
```

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

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

// makeTitleOfApp.js
```

```
function makeTitleOfApp()
{
    let theTitle = ce('a');
    theTitle.id = 'theTitle';
    theTitle.href =
'https://github.com/ChristopherTopalian';
    theTitle.target = '_blank';
    theTitle.textContent = 'Topalian JavaScript
Genetics Navigator';
    theTitle.style.position = 'absolute';
    theTitle.style.right = 0 + 'px';
    theTitle.style.bottom = -5 + 'px';
    theTitle.style.margin = 10 + 'px';
```



```
theTitle.style.fontSize = '15px';  
theTitle.style.fontWeight = 'bold';  
theTitle.style.textAlign = 'right';  
theTitle.style.lineHeight = 15 + 'px';  
theTitle.style.textDecoration = 'none';  
panelLeft.append(theTitle);  
}
```

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

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

```
// filterGenes.js
```

```
function filterGenes(whichArray)  
{  
    let searchValue =  
    ge('searchGene').value.toLowerCase();
```

```
let filteredGenes = [];  
  
for (let i = 0; i < whichArray.length; i++)  
{  
    if  
(whichArray[i].name.toLowerCase().includes(s  
earchValue))  
    {  
        filteredGenes.push(whichArray[i]);  
    }  
}  
  
displayResults(filteredGenes);  
}
```

//--//

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

```
//  
https://sites.google.com/view/CollegeOfScripting  
  
// findIndexByName.js  
  
function findIndexByName(whichArray,  
whichName)  
{  
    let index = whichArray.findIndex(function(a)  
    {  
        return a.name === whichName  
    });  
  
    return index;  
}  
  
/--/  
  
// Dedicated to God the Father  
// All Rights Reserved Christopher Andrew  
// Topalian Copyright 2000-2024  
// https://github.com/ChristopherTopalian  
//  
https://github.com/ChristopherAndrewTopalian
```

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

```
// displayResults.js
```

```
function displayResults(filteredGenes)
```

```
{
```

```
    let resultsDiv = ge('results');
```

```
    resultsDiv.innerHTML = '';
```

```
    if (filteredGenes.length === 0)
```

```
    {
```

```
        resultsDiv.innerHTML = 'No results  
found.';
```

```
        return;
```

```
    }
```

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

```
    {
```

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

```
        geneDiv.className = 'result';
```

```
        resultsDiv.append(geneDiv);
```

```
//-//
```

```
let geneNameDiv = ce('div');  
geneNameDiv.innerHTML = 'Name: ' +  
'<b>' + filteredGenes[i].name + '</b>';  
geneDiv.append(geneNameDiv);
```

```
//-//
```

```
let geneFunctionDiv = ce('div');  
geneFunctionDiv.textContent = 'Function: '  
+ filteredGenes[i].function;  
geneDiv.append(geneFunctionDiv);
```

```
//-//
```

```
let geneChromosomeDiv = ce('div');  
geneChromosomeDiv.textContent =  
'Chromosome: ' +  
filteredGenes[i].chromosome;  
geneDiv.append(geneChromosomeDiv);
```

```
//-//
```

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

```
geneMutationsDiv.textContent =  
'Mutations: ' + filteredGenes[i].mutations.join(',  
'');
```

```
geneDiv.append(geneMutationsDiv);
```

```
//-//
```

```
let geneSequenceDiv = ce('div');  
geneSequenceDiv.textContent =  
'Sequence: ' +  
filteredGenes[i].sequence.substring(0, 10) +  
'...';
```

```
geneDiv.append(geneSequenceDiv);
```

```
}  
}
```

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

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

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

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

```
//
```

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

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

```
// sound.js
```

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

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

```
  oscillator.frequency.setValueAtTime(frequency,  
audioCtx.currentTime);  
  oscillator.detune.setValueAtTime(detune,  
audioCtx.currentTime);
```

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

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

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

// attack

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

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

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

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

```
function hoverSound()
```



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

```
function hoverSound2()  
{  
    playTone(  
        300,    // frequency  
        300,    // duration  
        'triangle', // waveType  
        0.08,   // volume  
        -200,   // detune  
        0.1,    // attackTime  
        0.1,    // releaseTime  
        0.0     // panValue -0.5 is left  
    );  
}
```

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

//--//

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

```
// whenLoaded.js
```

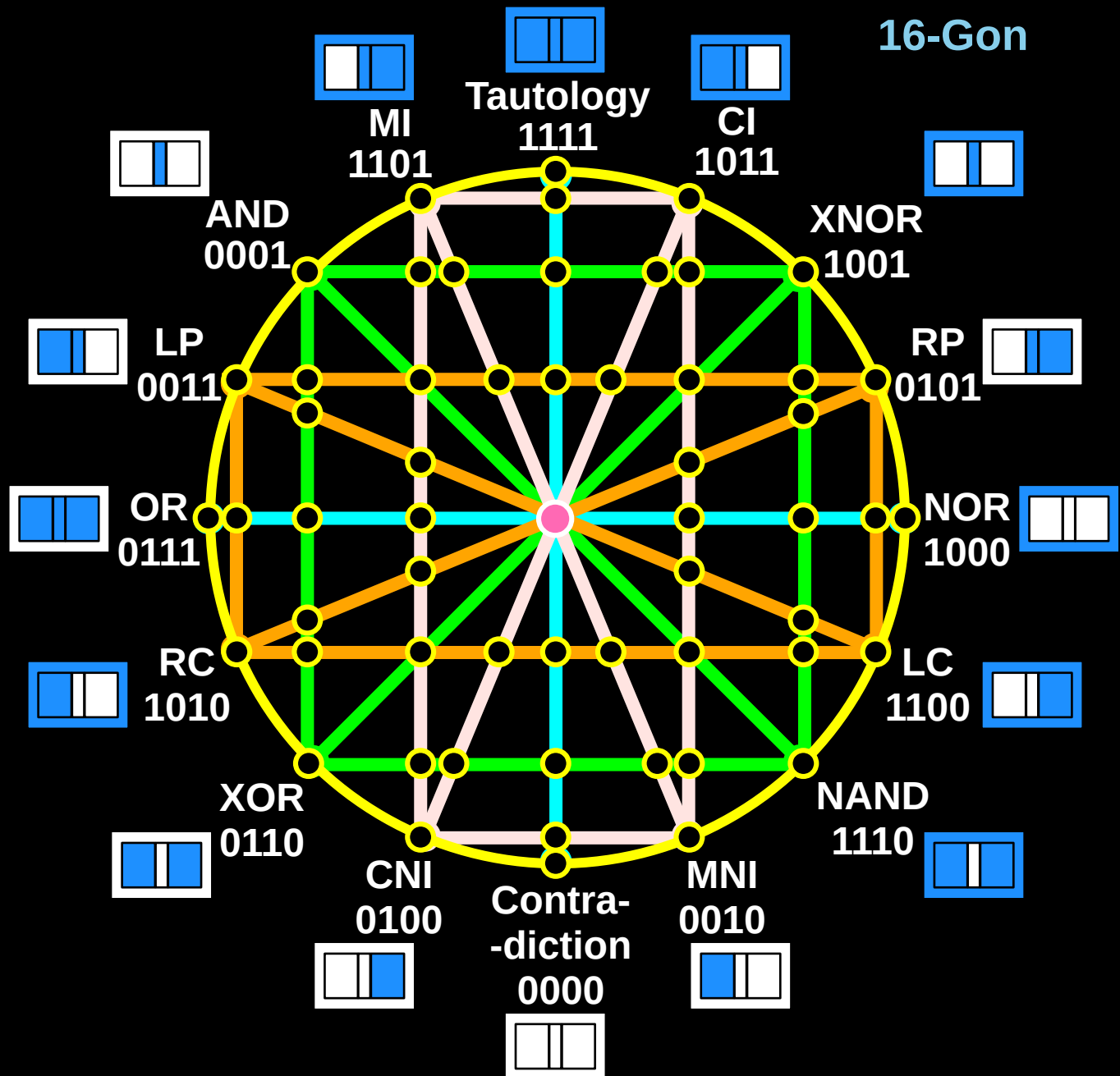
```
function whenLoaded()  
{  
    makeInterface();  
  
    makeSearchBox();  
  
    makeNameButtons();  
  
    makeNamesAllButtons();  
  
    makeResultsDiv();  
  
    makeTitleOfApp();  
}
```

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

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

```
//  
https://sites.google.com/view/CollegeOfScripti  
ng  
  
</script>  
  
</head>  
  
<body onload = "whenLoaded();">  
  
</body>  
  
</html>
```

True Artificial Intelligence System



For More Tutorials:

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

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

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

CollegeOfScripting.weebly.com

CollegeOfScripting.wordpress.com

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

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

[Rumble.com/user/CollegeOfScripting](https://rumble.com/user/CollegeOfScripting)

Dedicated to God the Father

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

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

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

CollegeOfScripting.weebly.com