

Wynnnter Ong
CS 370
Software Engineering
Project 3

Summary

This is a basic demo converting normal text into highway sign style text. This demo can convert all the letters (a-zA-Z), all the numbers (0-9), and some special characters (&-/'!?). This project was coded using CSS/HTML/JS.

This can be run by double clicking the html file, which should open it in the web browser.

Showcase

Highway GUI Demo

Enter text in the input box below, and click the "Submit Text" button to convert the text into highway sign style text.
To clear the highway sign style text, click the "Clear Text" button.



Highway GUI Demo

Enter text in the input box below, and click the "Submit Text" button to convert the text into highway sign style text.
To clear the highway sign style text, click the "Clear Text" button.



Highway GUI Demo

Enter text in the input box below, and click the "Submit Text" button to convert the text into highway sign style text.
To clear the highway sign style text, click the "Clear Text" button.



Highway GUI Demo

Enter text in the input box below, and click the "Submit Text" button to convert the text into highway sign style text.
To clear the highway sign style text, click the "Clear Text" button.



Documentation

The code is found in the same zip file as this, named HighwayGUI.html

The code at the very beginning is style code, which designates where the CSS starts and ends. It centers the text, changes the background to black, and changes the color to orange to simulate a highway sign.

The next part of the code is where the HTML starts, with a text explanation. Next there is a input text field, followed by two buttons which are run the functions `parse()` and `remove()` when clicked. After that there is a div section with `id="content"`, which lets the CSS know exactly where to implement itself and where the functions will work.

The third part of the code is the JS, which contains 3 methods and a table called `characters` which contains all the possible characters converted into an array form.

- First is the `remove()` function, which simply deletes everything in the div `id="content"` as long as there is a "child" in it.
- Second is the `parse()` function, which is the main part. The first line creates a new div element, which will contain the single line of parsed text. The second line creates a text variable which gets the text in the input box, while also lowercasing all the letters for easier parsing. The third line creates `characterstring`, which will actually hold all the raw

HTML text to be displayed. Because the characters are represented with 7 rows, 7 separate arrays are created for each row.

Onto the for loops, the first one goes through the input text and gets each word and puts it into a variable currword. The second for loop is to repeat it 7 times, one for each row. The third for loop parses through each character in currword, and will push a value into that row's array depending on the value. If it is a character, it will parse through the character's current row and push a 0 where a dot would be and a 1 where whitespace would be. Afterwards, if the character parsed was a special symbol, it will push a 1,2,3 depending on how wide the character is and how much whitespace is needed. Otherwise if the character parsed was a whitespace character, it will push a 3 to represent a large whitespace. After the for loops, all the row arrays will be filled up with the data needed to display the text.

To convert the arrays into usable html, an iterator is created that is the row array's values, and a for loop goes through them and passes the current value at each index into an Output() function. After passing through the function, it gets returned a string and appends that string onto the characterstring. After the for loop is completed, a
 is appended onto the end of the characterstring to start a new row, and this process is repeated for all the row arrays.

The second to last line just links the characterstring to the div element as HTML, and the last line adds the div element to the div id="content".

- The third function is Output(), which converts the 0,1,2, or 3 into actual HTML. The 0 represents ●, which is a black circle [documented here](#). The 1,2,3 represents , which is whitespace. The higher the number the more whitespace is added.