

Text To Speech Webpage Tutorial

For the text to speech webpage, I first used a free tutorial resource from Codepen titled, Text to Speech with Voice Customization. I will post the link at the end of this tutorial document. The tutorial supplied the HTML, CSS, and JavaScript code. We will be using the Visual Studio Text Editor, so you will be able to view your work using the Google Chrome Browser. Listed below are screen shots of the three files and their code.

Open the Visual Studio text editor and click “New” and type in the code supplied below, then click “Save As”, and type in index.html.

```
20
21 <div class="wrapper">
22   <label for="text">AI Text to Speech</label>
23   <textarea id="text">What are the advantages of text to speech technology?</textarea>
24   <div class="properties">
25     <label for="voice">Voice:</label>
26     <select id="voice"></select>
27   </div>
28
29   <label for="pitch">Pitch:</label>
30   <input id="pitch" type="range" min="0.1" max="2" step="0.1" value="1">
31   <output for="pitch">1</output>
32
33   <label for="rate">Rate:</label>
34   <input id="rate" type="range" min="0.1" max="2" step="0.1" value="1">
35   <output for="rate">1</output>
36
37   <label for="volume">Volume:</label>
38   <input id="volume" type="range" min="0" max="1" step="0.1" value="1">
39   <output for="volume">1</output>
40 </div>
41 <button id="speak">Speak</button>
42
```

Again, in the Visual Studio text editor and click “New” and type in the code supplied below, then click “Save As”, and type in style.css. Note within the body I have used a background image, you will supply your own image, or you can choose a web gradient background color. This is the complete CSS code for this tutorial.

```
1  *, ::before, ::after {
2  |   box-sizing: border-box;
3  | }
4
5  body {
6  |   display: grid;
7  |   justify-content: center;
8  |   align-items: center;
9  |   min-height: 100vh;
10 |   margin: 0;
11 |   padding: 20px;
12 |   color: ■ linen;
13 |   background-image: url("pattern.gif");
14 |   opacity: 90%;
15 |   font-size: 30px;
16 | }
17
18  p{
19 |   color:■ antiquewhite;
20 |   font-family: "Lucida Console", "Courier New", monospace ;
21 |   font-size: 200px;
22 |
23 | }
24
25  .wrapper {
26 |   display: grid;
27 |   gap: 20px;
28 |   width: 800px;
29 |   max-width: calc(100vw - 40px);
30 |   padding: 30px;
31 |   border-radius: 10px;
32 |   background-color: ■ teal;
33 |   background-image: linear-gradient(280deg, ■ #4b3a48 25%, ■ teal 100%, ■ #FFAC1C 25%);
34 |
35 | }
```

```

35     }
36
37     < #text {
38         display: inline-block;
39         height: 100px;
40         padding: 20px;
41         background-color: #aliceblue;
42         border: none;
43         font-size: 40px;
44         font-family: "Lucida Console", "Courier New", monospace;
45         resize: vertical;
46     }
47
48     < .properties {
49         display: grid;
50         grid-template-columns: max-content minmax(0, auto) 40px;
51         gap: 20px;
52         padding: 20px;
53         background-color: #0003;
54     }
55
56     < #voice {
57         font-size: 0.875rem;
58         font-family: "Lucida Console", "Courier New", monospace;
59     }
60
61     < #speak {
62         padding: 10px;
63         border: 1px solid #antiquewhite;
64         color: #linen;
65         background-color: #333;
66         font-size: inherit;
67         font-family: "Lucida Console", "Courier New", monospace;
68         cursor: pointer;
69         appearance: none;
70     }

```

```

71
72
73     < h1{
74
75         font-family: "Lucida Console", "Courier New", monospace;
76         font-size: 65px;
77         text-align:center;
78     }
79
80
81     < ul.myUL {
82         font-size:55px;
83         font-family: "Lucida Console", "Courier New", monospace;
84         display: inline-block;
85         text-align: left;
86     }
87
88
89     < hr{
90         margin-top: inherit;
91         border-top: 1px double #FFAC1C;
92     }
93
94

```

In order for our webpage to be responsive we will need to add Media Queries to the style.css file. Type in the code just below the hr tag, and then click save.

```
94
95
96 ⑴ @media only screen and (min-width: 280px), (max-width:350px) {
97 ⑴  body, .wrapper {
98     width: 100%;
99     padding: 10px;
100    margin-left: 12px;
101
102  }
103 }
104 ⑴ @media only screen and (min-width: 280px), (max-width:350px) {
105 ⑴  img{
106     width: 100%;
107     margin-left: 40px;
108
109  }
110 }
111
112 ⑴ @media only screen and (min-width: 280px), (max-width:350px) {
113 ⑴  #text, #speak, hr{
114     font-size:18px;
115
116  }
117 }
118 ⑴ @media only screen and (min-width: 280px), (max-width:350px) {
119 ⑴  .properties{
120
121     font-size: 18px;
122
123  }
124 }
```

```

124 }
125 ∨ @media only screen and (min-width: 280px ),(max-width:350px) {
126 ∨   h1 {
127     |
128     |   font-size: 24px;
129     |
130   }
131 ∨ @media only screen and (min-width: 280px), (max-width:350px) {
132 ∨   ul.myUL{
133     |
134     |   font-size: 22px;
135     |
136   }
137 }
138
139 ∨ @media only screen and (min-width: 280px), (max-width:350px) {
140 ∨   p{
141     |
142     |   font-size: 90px;
143     |
144   }
145 }
146
147 }
148 ∨ @media only screen and (max-width: 2000px) {
149 ∨   body, .wrapper, #text, .properties, #voice, #speak, h1, p, hr, header p {
150     |   width: 100%;
151   }
152 }

```

Lastly, in the Visual Studio text editor click “New” and type in the code supplied below, then click “Save As”, and type in script.js. This is the complete JavaScript code for the tutorial.

```

1  // grab the UI elements to work with
2  const textEl = document.getElementById('text');
3  const voiceInEl = document.getElementById('voice');
4  const pitchInEl = document.getElementById('pitch');
5  const rateInEl = document.getElementById('rate');
6  const volumeInEl = document.getElementById('volume');
7  const pitchOutEl = document.querySelector('output[for="pitch"]');
8  const rateOutEl = document.querySelector('output[for="rate"]');
9  const volumeOutEl = document.querySelector('output[for="volume"]');
10 const speakEl = document.getElementById('speak');
11
12 // add UI event handlers
13 pitchInEl.addEventListener('change', updateOutputs);
14 rateInEl.addEventListener('change', updateOutputs);
15 volumeInEl.addEventListener('change', updateOutputs);
16 speakEl.addEventListener('click', speakText);
17
18 // update voices immediately and whenever they are loaded
19 updateVoices();
20 window.speechSynthesis.onvoiceschanged = updateVoices;
21
22 function updateOutputs() {
23   // display current values of all range inputs
24   pitchOutEl.textContent = pitchInEl.value;
25   rateOutEl.textContent = rateInEl.value;
26   volumeOutEl.textContent = volumeInEl.value;
27 }
28
29 function updateVoices() {
30   // add an option for each available voice that isn't already added
31   window.speechSynthesis.getVoices().forEach(voice => {
32     const isAlreadyAdded = [...voiceInEl.options].some(option => option.value === voice.voiceURI);
33     if (!isAlreadyAdded) {
34       const option = new Option(voice.name, voice.voiceURI, voice.default, voice.default);
35       voiceInEl.add(option);
36     }
37   });
38 }

```

```

38 }
39
40 function speakText() {
41   // stop any speaking in progress
42   window.speechSynthesis.cancel();
43
44   // create new utterance with all the properties
45   const text = textEl.value;
46   const utterance = new SpeechSynthesisUtterance(text);
47   utterance.voice = window.speechSynthesis.getVoices().find(voice => voice.voiceURI === voiceInEl.value);
48   utterance.pitch = pitchInEl.value;
49   utterance.rate = rateInEl.value;
50   utterance.volume = volumeInEl.value;
51
52   // speak that utterance
53   window.speechSynthesis.speak(utterance);
54 }
55

```

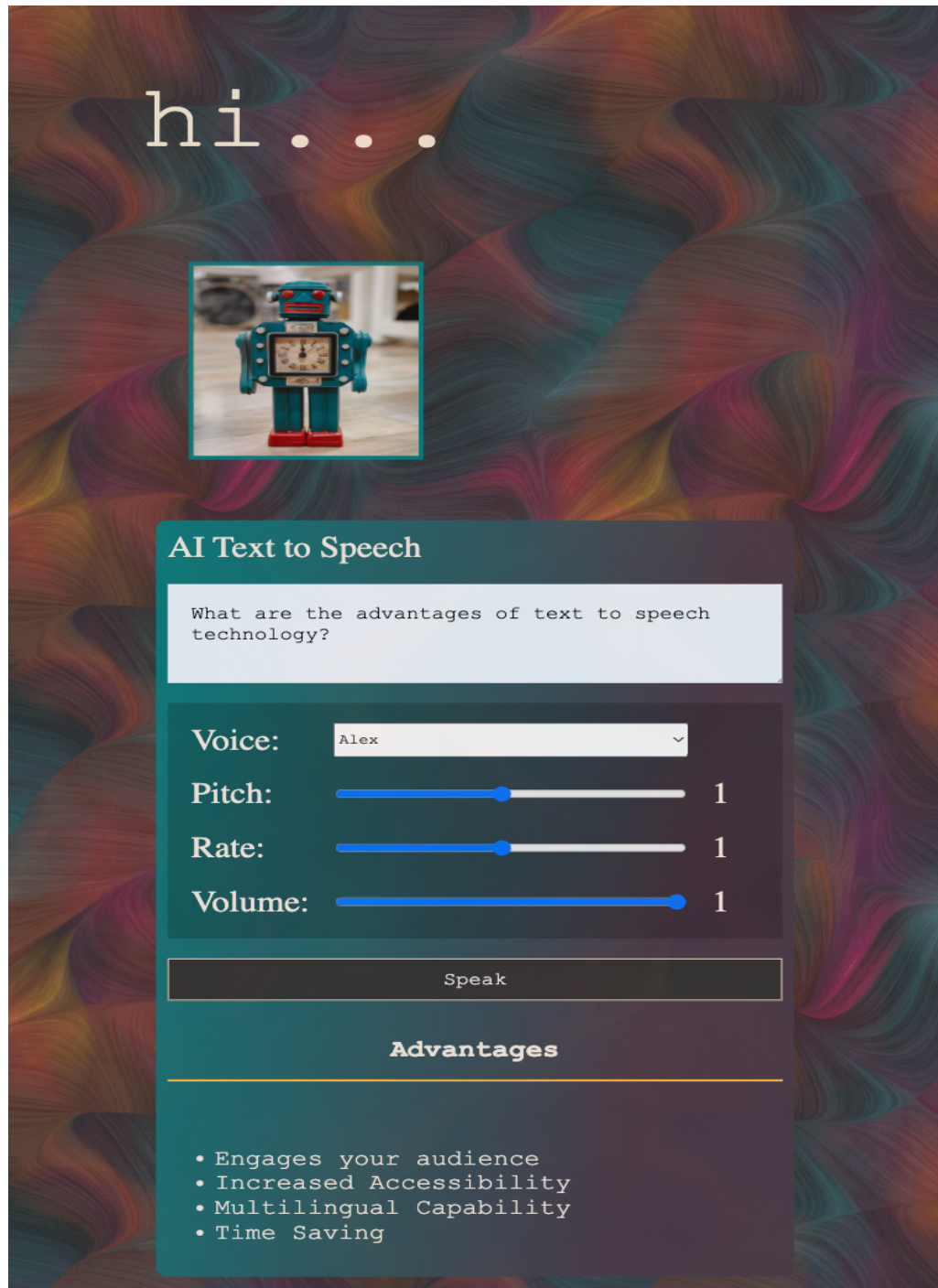
Next, we need to add a `<!DOCTYPE html>` tag, a head tag, and meta tags to the top of our index.html file. We will also be adding a header tag, a paragraph tag, and a div tag with an image and border around the image (you will supply your own image). Type in the code below to the top of your index.html file and then click save.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,initial-scale=1.0">
7   <meta name="description" content="A text to speech webpage example using AI.">
8   <meta name="keywords" content="Text, Speech, Webpage, Example, Trend, AI">
9   <title>Text to Speech</title>
10  <link rel="stylesheet" href="style.css">
11  <script src="script.js" type="module"></script>
12 </head>
13 <body>
14   <p>hi...</p>
15   <div>
16     
17   </div>
18 </body>
19 </html>
```

Lastly, we will be adding a div class tag, a heading one tag, and unordered list tags in the index.html file. We will place these tags just below the button tag. Type in the code below in your index.html file and then click save.

```
42
43   <div class="detail">
44     <h1>Advantages</h1><hr><br>
45     <ul class="myUL">
46       <li>Engages your audience</li>
47       <li>Increased Accessibility</li>
48       <li>Multilingual Capability</li>
49       <li>Time Saving</li>
50     </ul></div>
51
52   </div>
53
54 </body>
55 </html>
```

Congratulations you completed the text to speech tutorial! If you did it correctly you will get the following output below.



Codepen Tutorial. URL: <https://codepen.io/lonekorean/pen/PozMjgO>

