Talking to your browser
(doesn't make you crazy if your browser can respond)

Chen Hui Jing / @hj_chen

</RK>
SingaporeCSS

Kena cheat onto the pirate ship liab... (०﹏०)
What is CSS? 😐

Nah… I'm just kidding. I'm going make a bold assumption that even if you don't write CSS, you at least heard of it before and know what it does. Hopefully.
Evolution of CSS Specifications

CSS1
Recommendation: 17 Dec 1996

CSS2
Recommendation: 12 May 1998

CSS2.1
Recommendation: 7 Jun 2011

CSS2.2
Working draft: 12 Apr 2016

CSS3
Decision to modularise: 14 Apr 2000
(26 modules)

CSS Snapshot 2018
(99 modules)

Completed
CSS Snapshot 2018
CSS Snapshot 2017
CSS Snapshot 2015
CSS Snapshot 2010
CSS Snapshot 2007
CSS Color Level 3
CSS Namespaces
Selectors Level 3
CSS Level 2 Revision 1
Media Queries
CSS Style Attributes
CSS Fonts Level 3
CSS Basic User Interface Level 3

Stable
CSS Backgrounds and Borders Level 3
CSS Conditional Rules Level 3
CSS Multi-column Layout Level 1
CSS Values and Units Level 3
CSS Cascading and Inheritance Level 3
CSS Fonts Level 3
CSS Writing Modes Level 3
CSS Counter Styles Level 3
CSS Containment Level 1

Rewriting
CSS Generated Content Level 3

Revising
CSS Paged Media Level 3
CSSOM View
CSS Intrinsic & Extrinsic Sizing Level 3
CSS Ruby Level 1
CSS Overflow Level 3
CSS Box Model Level 3
CSS Scrollbars Level 1

Refining
CSS Animations Level 1
Web Animations
CSS Text Level 3
CSS Transforms Level 1
CSS Transitions
CSS Box Alignment Level 3
Selectors Level 4
CSS Lists Level 2
Motion Path Level 1
Preview of CSS Level 2
CSS Fonts Level 4
CSS Easing Functions Level 1
CSS Logical Properties and Values Level 1

Testing
CSS Images Level 3
CSS Speech
CSS Text Decoration Level 3
CSS Shapes Level 1
CSS Blending Level 1
CSS Fragments Level 3
CSS Grid Layout Level 1
CSS Display Level 3
CSS Will Change Level 1
CSS Logical Properties and Values Level 4

Exploring
CSS Backgrounds and Borders Level 4
CSS Device Adaptation
CSS Exclusions
Filter Effects Level 1
CSS Generated Content for Paged Media
CSS Page Flows
CSS Template Layout
CSS Line Grid
CSS Positioned Layout Level 3
CSS Regions
CSS Table Level 3
CSS Object Model
CSS Font Loading
CSS Scroll Level 1
CSS Inline Layout Level 3
CSS Scrollbar Level 1
CSS Basic User Interface Level 4
CSS Text Level 4
CSS Properties and Values API Level 1

CSS Typing OOM Level 1
CSS Worklets Level 1
CSS Color Level 4
CSS Intrinsic Sizing Level 1
CSS Image Values and Replaced Content Level 4
CSS Fill and Stroke Level 3
CSS Overflow Level 4
CSS Grid Layout Level 2
CSS Text Decoration Level 4
CSS Layout API Level 1
CSS Values and Units Level 4
CSS Shadow Parts
CSS Fragmentation Level 4
CSS Spatial Navigation Level 1
CSS Color Adjustment Level 1
CSS Overflow Behavior Level 1
CSS Animation Worklet API
CSS Containment Level 2
CSS Snapshot 2018
(99 modules)

Completed
CSS Snapshot 2018
CSS Snapshot 2017
CSS Snapshot 2015
CSS Snapshot 2010
CSS Snapshot 2007
CSS Color Level 3
CSS Namespaces
Selectors Level 3
CSS Level 2 Revision 1
Media Queries
CSS Style Attributes
CSS Fonts Level 3
CSS Basic User Interface Level 3

Stable
CSS Backgrounds and Borders Level 3
CSS Conditional Rules Level 3
CSS Multi-column Layout Level 1
CSS Values and Units Level 3
CSS Cascading and Inheritance Level 3
CSS Fonts Level 3
CSS Writing Modes Level 3
CSS Counter Styling Level 3
CSS Containment Level 1

Exploring
CSS Backgrounds and Borders Level 4
CSS Device Adaptation
CSS Exclusions
Filter Effects Level 1
CSS Generated Content for Paged Media
CSS Page Floats
CSS Template Layout
CSS Line Grid
CSS Positioned Layout Level 3
CSS Regions
CSS Table Level 3
CSS Object Model
CSS Font Loading
CSS Scoping Level 1
CSS Inline Layout Level 3
CSS Round Display Level 1
CSS Basic User Interface Level 4
CSS Text Level 4
CSS Properties and Values API Level 1

CSS Color Level 4

CSS Typed OM Level 1
Worklets Level 1
CSS Rhythmic Sizing Level 1
CSS Image Values and Replaced Content Level 4
CSS Fill and Stroke Level 3
CSS Overflow Level 4
CSS Grid Layout Level 2
CSS Text Decoration Level 4
CSS Layout API Level 1
CSS Values and Units Level 4
CSS Shadow Parts
CSS Fragmentation Level 4
CSS Spatial Navigation Level 1
CSS Color Adjustment Level 1
CSS Overscroll Behavior Level 1
CSS Animation Worklet API
CSS Containment Level 2

Revising
CSS Paged Media Level 3
CSSOM View
CSS Intrinsic & Extrinsic Sizing Level 3
CSS Ruby Level 1
CSS Overflow Level 3
CSS Box Model Level 3
CSS Pseudo-Elements Level 4
CSS Scrollbars Level 1

CSS Animations Level 1
Web Animations
CSS Text Level 3
CSS Transforms Level 1
CSS Transitions
CSS Box Alignment Level 3
Selectors Level 4
CSS Lists Level 3
Motion Path Level 1
Preview of CSS Level 2
CSS Fonts Level 4
CSS Easing Functions Level 1
CSS Logical Properties and Values Level 1

Refining
CSS Images Level 3
CSS Speech
CSS Text Decoration Level 3
CSS Shapes Level 1
CSS Masking Level 1
CSS Fragmentation Level 3
CSS Cascading Variables
Compositing and Blending Level 1
CSS Syntax Level 3

Testing
CSS Grid Layout Level 1
CSS Display Level 3
CSS Will Change Level 1
Media Queries Level 4
Geometry Interfaces Level 1
CSS Cascading and Inheritance Level 4
CSS Scroll Snap Level 1
CSS Painting API Level 1
CSS Writing Modes Level 4
5.1. Named Colors

CSS defines a large set of **named colors**, so that common colors can be written and read more easily. A `<named-colors>` is written as an `<ident>`, accepted anywhere a `<color>` is. As usual for CSS-defined `<ident>`s, all of these keywords are case-insensitive.

The names resolve to colors in sRGB.

16 of CSS’s named colors come from HTML originally: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. Most of the rest come from one version of the X11 color system, used in Unix-derived systems to specify colors for the console. (Two special color values, `transparent` and `currentcolor`, are specially defined in their own sections.)

The following table defines all of the opaque named colors, by giving equivalent numeric specifications in the other color syntaxes.

<table>
<thead>
<tr>
<th>Named</th>
<th>Numeric</th>
<th>Color name</th>
<th>Hex</th>
<th>rgb</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aliceblue</code></td>
<td><code>#F0F8FF</code></td>
<td>240 248 255</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>antiquewhite</code></td>
<td><code>#FAEBD7</code></td>
<td>250 235 215</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>aqua</code></td>
<td><code>#00FFFF</code></td>
<td>0 255 255</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>aquamarine</code></td>
<td><code>#7FFFD4</code></td>
<td>127 255 212</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>azure</code></td>
<td><code>#F0FFFF</code></td>
<td>240 255 255</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>beige</code></td>
<td><code>#F5F5DC</code></td>
<td>245 245 220</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>bisque</code></td>
<td><code>#FFE4C4</code></td>
<td>255 228 196</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>black</code></td>
<td><code>#000000</code></td>
<td>0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>blanched almond</code></td>
<td><code>#FFEBCD</code></td>
<td>255 235 205</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>blue</code></td>
<td><code>#0000FF</code></td>
<td>0 0 255</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>blueviolet</code></td>
<td><code>#8A2BE2</code></td>
<td>138 43 226</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>brown</code></td>
<td><code>#A52A2A</code></td>
<td>165 42 42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>burlywood</code></td>
<td><code>#DEB887</code></td>
<td>222 184 135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>cadetblue</code></td>
<td><code>#5F9EA0</code></td>
<td>95 158 160</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>chartreuse</code></td>
<td><code>#7FFF00</code></td>
<td>127 255 0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If you also like colours...

*insert random caption here*

Where did CSS named colours come from?
The list of 256 extended color keywords from the [IC-CIELAB] color space and their corresponding color codes in Hex and RGB format.
Your browser can speak English

For the most part.
Cue bo liao idea...

Singlish lesson #1: Bo liao

Hokkien for “nothing better to do” . Dangerously idle.
In Mandarin, it's “无聊 (wú liáo) ”
“What for he go and do that sort of thing? Must be damn bo liao.”

Source: The Oxford Singlish Dictionary
Change background colour with your voice!

Ah-maz-ing
Web Speech API

Speech Recognition
provides the ability to recognise voice context from an audio input and respond appropriately
- SpeechRecognition
- SpeechGrammar

Speech Synthesis
a text-to-speech component that allows programs to read out their text content
- SpeechSynthesis
- SpeechSynthesisVoice
- SpeechSynthesisUtterance
Make your browser listen to you 🎧
Bo liao also must plan

甲、Have web page
乙、Set background colour with CSS custom property
丙、Have button to trigger microphone
丁、Capture voice and process with SpeechRecognition
戊、Use result to update background colour
<!doctype html>
<html class="no-js" lang="en">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>Let's talk CSS colours</title>
  <meta name="description" content="Playing around with the WebSpeech API, CSS custom properties and CSS naming conventions."
  <meta name="author" content="Chen Hui Jing">

  <link rel="stylesheet" href="styles.css">

  <!--[if lt IE 9]>
  <![endif]-->
</head>

<body>
  <main></main>
  <script src="scripts.js"></script>
</body>
</html>
Set background colour with CSS custom property

```css
:root {
  --bg-colour: transparent;
}

main {
  /* Of course got other styles la... */
  /* You think magic meh... */
  background-color: var(--bg-colour);
}

/* Moar styles not shown here */
```
What is a CSS custom property? 🤔

Defined in CSS Custom Properties for Cascading Variables Module Level 1

Introduces cascading variables as a new primitive value type that is accepted by all CSS properties, and custom properties for defining them

```
var( <custom-property-name> , <declaration-value>? )
```
Have button to trigger microphone

<body>
<main>
  <h1>CSS Colours</h1>
  <p>How well do you know CSS named colours? Test both your knowledge as well as your browser's ability to recognise your accent when you speak English <span class="kaomoji">^-_(-ツ)_/-</span></p>

  <button type="button" id="activateMic" class="btn-speak">Speak</button>

  <pre><code id="consoleLog">Click the button then say a colour…</code></pre>
</main>
<script src="scripts.js"></script>
</body>
Capture voice and process with SpeechRecognition

But first...
Feature detection.
# Browser support for SpeechRecognition

<table>
<thead>
<tr>
<th>Speech Recognition API</th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Chrome for Android</th>
<th>Android Browser</th>
<th>Samsung Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9</td>
<td>16</td>
<td>68</td>
<td>76</td>
<td>12</td>
<td>13.1</td>
<td>4.4</td>
<td>9.2</td>
<td>10</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>17</td>
<td>69</td>
<td>77</td>
<td>12.1</td>
<td>12.4</td>
<td>all</td>
<td></td>
<td>9.2</td>
<td>4.4.4</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>18</td>
<td>70</td>
<td>78</td>
<td>13</td>
<td>13.2</td>
<td>all</td>
<td>78</td>
<td>76</td>
<td>10.1</td>
</tr>
</tbody>
</table>

Data from caniuse.com | Ember from caniuse.org
((window, undefined) => {
    const document = window.document;
    const docElement = document.documentElement;

    const speechRecognition = window.webkitSpeechRecognition || window.mozSpeechRecognition || window.speechRecognition;
    const speechGrammarList = window.webkitSpeechGrammarList || window.mozSpeechGrammarList || window.speechGrammarList;

    function addClass(className) {
        docElement.className = `\${docElement.className} ${className}\`;
    }

    docElement.className = docElement.className.replace(/(^|\s)no-js(\s|$)/, '\$1js$2');
    if (speechRecognition !== undefined) {
        addClass('speech');
    } else {
        addClass('no-speech');
    }
})(window);

Feature detection by Cătălin Mariș
To activate in Firefox

Must be newer than 72.0a1 (2019-10-22)

- type about:config in your address bar
- search for the media.webspeech.recognition.enable and media.webspeech.recognition.force_enable preferences
- set them to true
How come browser can speak English?

Record utterance

Receive transcript

Processing by speech recognition engine
Who's doing the processing?

Google.

Google Cloud Speech-to-Text, with speech recognition in 120 languages. Mozilla is currently developing their own service called Deep Speech, hopefully can be validated in 2020 as a replacement for Google, at least in English.
const colours = ['maroon', 'darkred', 'brown', 'firebrick', 'rosybrown', 'indianred', 'lightcoral', 'red', 'snow', 'mistyrose']; /* the rest of the 148 named CSS colours */
const grammar = '#JSGF V1.0; grammar colours; public <colour> = ' + colours.join(' | ') + ';';

JSpeech Grammar Format (JSGF)

#JSGF V1.0; states the format and version used. Must be included first.

grammar colours; public <colour> indicates the type of term we want recognised, followed by list of items separated by pipe character.
/* Define speech recognition instance */
const recognition = new speechRecognition();
/* Create new speech grammar list */
const speechRecognitionList = new speechGrammarList();
/* Add grammar to the list */
speechRecognitionList.addFromString(grammar, 1);
/* Add speech grammar list to speech recognition instance */
recognition.grammars = speechRecognitionList;
/* Set language of the recognition */
recognition.lang = 'en-US';
/* Can choose to return interim results or final results */
recognition.interimResults = false;
/* Set number of alternative potential matches */
recognition.maxAlternatives = 1;
const micBtn = document.getElementById('activateMic')
const consoleLog = document.getElementById('consoleLog')

micBtn.addEventListener('click', function() {
  recognition.start(); /* Start speech recognition service */
  consoleLog.innerHTML = 'Ready to receive a colour command.'
}, false)
recognition.onresult = function(event) {
  const last = event.results.length - 1;
  const colour = event.results[last][0].transcript;
  const sanitiseColour = colour.replace(/\s/g, ' ');
  consoleLog.innerHTML = 'You probably said: ' + sanitiseColour +
  docBody.style.setProperty('--bg-colour', sanitiseColour);
}

**Returns** `SpeechRecognitionResultList` object with `SpeechRecognitionResult` objects, which can be accessed like an array

`[last]` returns the `SpeechRecognitionResult` at the last position
SpeechRecognitionResultList(1)
  0: SpeechRecognitionResult(1)
    0: SpeechRecognitionAlternative
      confidence: 0.7521789073944092
      transcript: "green"
    <prototype>: SpeechRecognitionAlternativePrototype { transcript: Getter, confidence: Getter, ... }
    isFinal: true
    length: 1
  <prototype>: SpeechRecognitionResultPrototype { item: item(), length: Getter, isFinal: Getter, ... }
  length: 1
  <prototype>: SpeechRecognitionResultListPrototype { item: item(), length: Getter, ... }
Don't need an extra HTML element for the warning message

```css
.no-speech body::before {
  content: 'Tragically, your browser does not support the Speech API.';
  font-family: sans-serif;
  line-height: 1.3;
  font-size: 85%;
  padding: 0.5em;
  background-color: #ab3c3c;
  color: white;
  text-align: center;
}
```
Make your browser talk back 🎤
# Browser support for SpeechSynthesis 😞

**Speech Synthesis API**

A web API for controlling text-to-speech output.

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Chrome for Android</th>
<th>Android Browser</th>
<th>Samsung Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;9</code></td>
<td>16</td>
<td>68</td>
<td>76</td>
<td>12</td>
<td>12.1</td>
<td>12.4</td>
<td>all</td>
<td>78</td>
<td>78</td>
<td>10.1</td>
</tr>
<tr>
<td><code>10</code></td>
<td>17</td>
<td>69</td>
<td>77</td>
<td>12.1</td>
<td>12.4</td>
<td>all</td>
<td>all</td>
<td>78</td>
<td>78</td>
<td>10.1</td>
</tr>
<tr>
<td><code>11</code></td>
<td>18</td>
<td>70</td>
<td>78</td>
<td>13</td>
<td>13.2</td>
<td>all</td>
<td>all</td>
<td>78</td>
<td>78</td>
<td>10.1</td>
</tr>
<tr>
<td></td>
<td>76</td>
<td>71</td>
<td>76</td>
<td>TP</td>
<td>TP</td>
<td>TP</td>
<td>TP</td>
<td>TP</td>
<td>TP</td>
<td>TP</td>
</tr>
</tbody>
</table>

Data from caniuse.com | Timed from caniuse.devstack
Moar planning...

子、Add select dropdown and play button
丑、Populate select with device voice options
寅、Change voice based on selected option
卯、Play response when button clicked
Add requisite elements

```html
<form id="hearResponse" class="response">
  <select id="pickVoice"></select>
  <button id="playResponse" class="btn-response">Hear response</button>
</form>
```
const select = document.getElementById('pickVoice');
voices = speechSynthesis.getVoices();
voices.forEach(function(voice) {
    const option = document.createElement('option');
    option.textContent = voice.name + ' (' + voice.lang + ');
    if(voice.default) {
        option.textContent += ' -- DEFAULT';
    }
    option.setAttribute('data-lang', voice.lang);
    option.setAttribute('data-name', voice.name);
    select.appendChild(option);
});
Firefox voice list

Array(67)

const responseForm = document.getElementById('hearResponse');
responseForm.addEventListener('submit', function(event) {
  event.preventDefault();
  const select = document.getElementById('pickVoice');
  speechSynthesis.cancel(); /* Needed to clear the previous result */
  /* create a new SpeechSynthesisUtterance() instance */
  const utterStuff = new SpeechSynthesisUtterance(result);
  const selectedVoice = select.selectedOptions[0].getAttribute('data-voice');
  voices.forEach(function(voice) {
    if (voice.name === selectedVoice) {
      utterStuff.voice = voice;
    }
  });
  speechSynthesis.speak(utterStuff); /* Start the utterance being */
}, false)
Do the live demo thingy
Links and stuff

- Web Speech API (Draft Community Group Report)
- Using the Web Speech API
- Web Speech API - Speech Recognition
- Web Speech API Demonstration by Google
- Let's talk CSS Colours (the bo liao app)
- Source code for the bo liao app
Thank you!

https://www.chenhuijing.com

@hj_chen

@hj_chen

@huijing

Header font is Vera Cruz BT by Ray Cruz
Body font is Morandi by Jovica Veljović