Talk to Me:
Using the Web Speech API

by Lindsay Greene

About me!

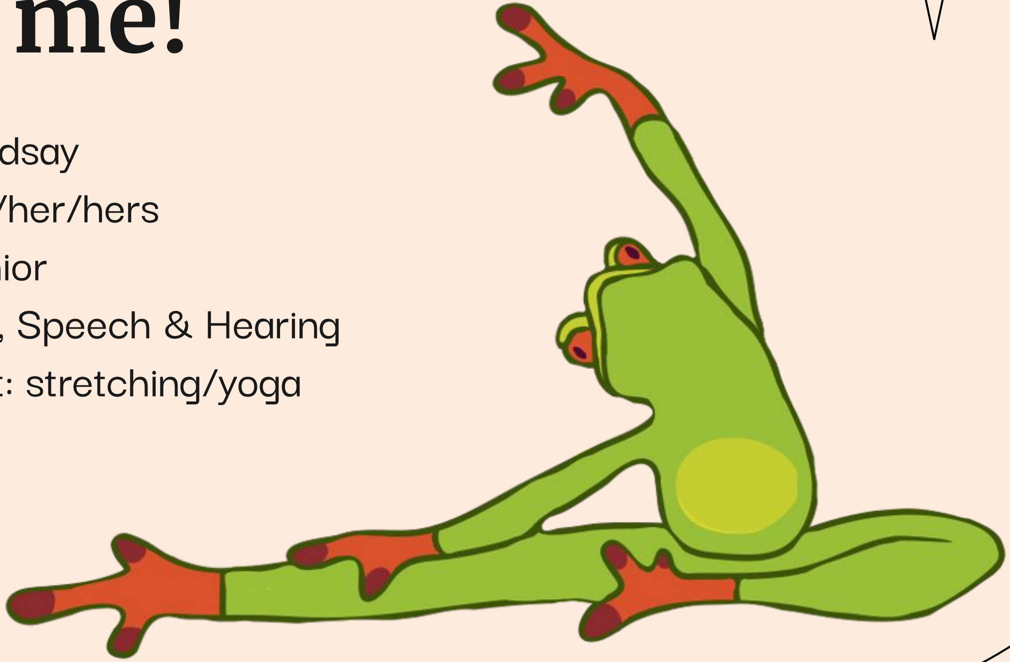
Name: Lindsay

Pronouns: she/her/hers

Year: junior

Studying: CS, Linguistics, Speech & Hearing

Favorite self care habit: stretching/yoga



Quick intros: Tell us your…

1. Preferred name
2. Pronouns (optional)
3. Favorite self-care habit

Ground rules

1.

Welcoming environment for all participants

2.

Support and teach each other

3.

Ask questions! Via chat or out loud.

4.

Leave having learned something new!

Today’s workshop:

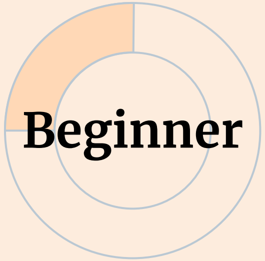
Web Speech API: used in natural language processing and generation

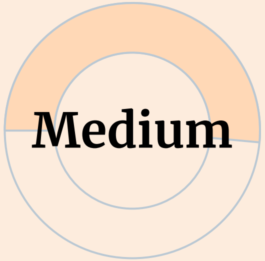
Tools you’ll need:

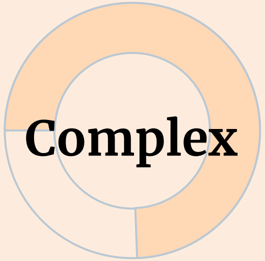
Chrome browser, VS code & live server

See download guide!

Demos







Self care
(today's example)

Talk to Me
(a personal project)

Speech articulator
(from Fizz Studio)

01.

Setting up files

An intro to the containers for our code

Tools to download

Google Chrome: https://www.google.com/chrome/

VS Code: https://code.visualstudio.com/download

VS Code Live Server extension:

- Open VS code
- On the left, click the extensions tab (looks like four blocks with one popped off)
- Search ”live server“ and install & enable the extension by Ritwick Dey
- Follow this tutorial (start at “Creating an HTML document”) to create your first file and ensure the extension works

Git: https://git-scm.com/downloads

Source code: https://github.com/lindsay-greene/hackathon-workshop

- Click the green **Code** button and select HTTPS, copy the link provided
- In the VS Code welcome window, choose **Clone Repository**
- Paste the link into the prompt that appears & choose **Clone from URL**
- Save the repository to your computer, and open it in VS Code
- You should now see five files! Open the HTML, CSS, and JS files

Understanding file types

HTML: Hypertext markup language

- Provides the content of a website
- Connects all three files
- Contains information for screen readers & the browser

CSS: Cascading style sheet

- Provides the style & cosmetic appearance of a website
- Background color, font size, text alignment, etc.

JS: JavaScript

- Provides the functionality of the website
- Makes things happen when you interact with certain elements

Vocabulary

API (application programming interface): snippets of code that allow you to borrow someone else's functionality or data. Common examples are Google Maps, sign in with Google, or the Web Speech API.

Function: reusable piece of code that performs a single, related action

NLG (natural language generation): field of CS that focuses on computers producing language

NLP (natural language processing): field of CS that focuses on computers understanding language

Repository (also called repo): a container for related code. Today's repo contains a ReadMe with a description of the repo, a Download Guide with tips and tricks, and the three files which are used to build our website

Variable: a data value that we assign to something we wish to access throughout the rest of our code

02.

Speech synthesis

How the computer can talk to us

Steps

1.

Declare text variables

2.

Declare speech variable

3.

Write the speak function

4.

Write on-click functions

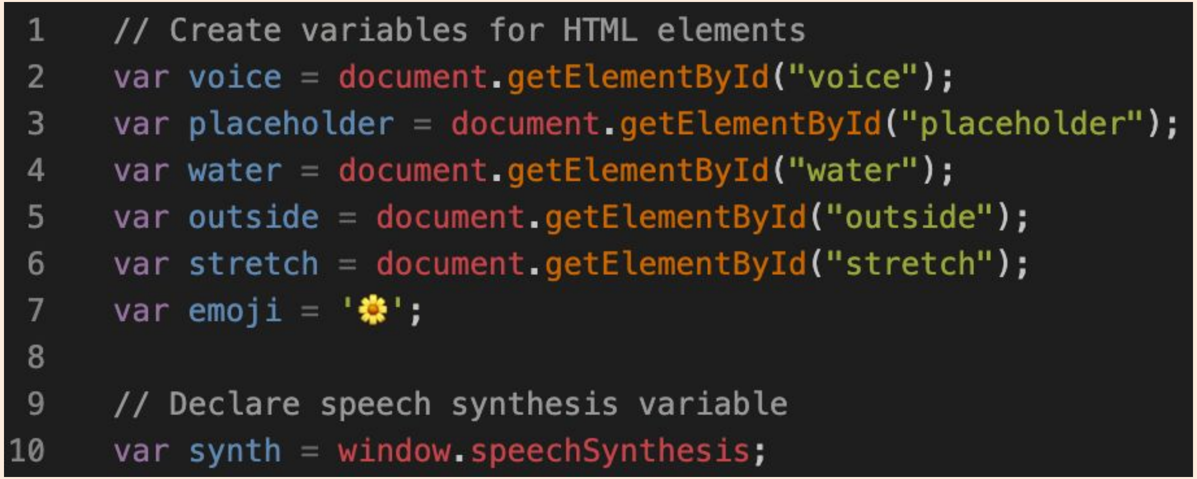
Declare variables

HTML Variables

- Lines 1 - 6
- Get access to placeholder & buttons
- Allows us to manipulate them in JS

Speech Synth Variable

- Line 10
- Give the synthesizer a name
- Allows easy access throughout the code

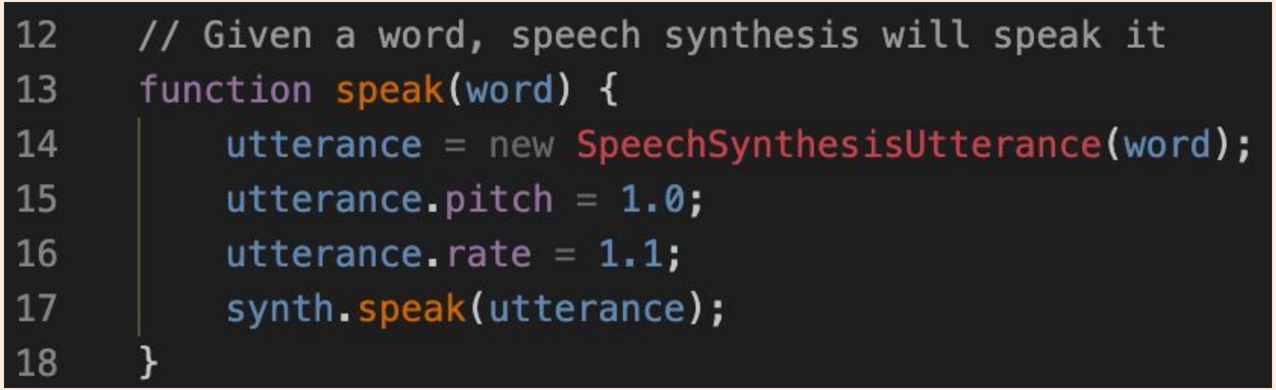


```
1    // Create variables for HTML elements
2    var voice = document.getElementById("voice");
3    var placeholder = document.getElementById("placeholder");
4    var water = document.getElementById("water");
5    var outside = document.getElementById("outside");
6    var stretch = document.getElementById("stretch");
7    var emoji = '🌼';
8
9    // Declare speech synthesis variable
10   var synth = window.speechSynthesis;
```

Speak function

Given a word, this function will speak it!

- Lines 12 - 18
- Declare new utterance with the function’s argument
- Change the pitch and rate to your liking
- Call the built-in speak method

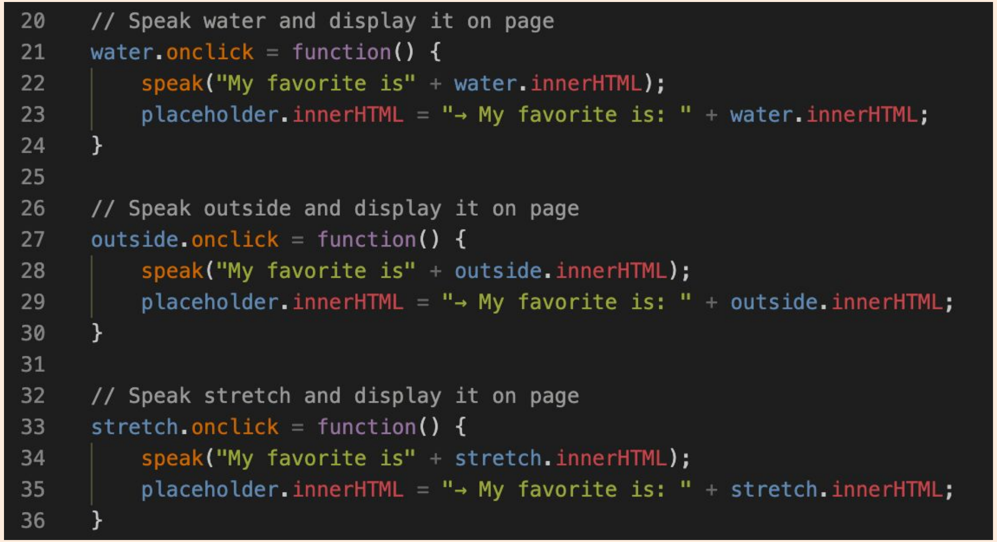


```
12  // Given a word, speech synthesis will speak it
13  function speak(word) {
14      utterance = new SpeechSynthesisUtterance(word);
15      utterance.pitch = 1.0;
16      utterance.rate = 1.1;
17      synth.speak(utterance);
18  }
```

On-click functions

These functions are executed when we click each button

- Lines 20 - 36
- Speak the option you selected
- Display your choice on the screen
- Very repetitive...



```
20  // Speak water and display it on page
21  water.onclick = function() {
22      speak("My favorite is" + water.innerHTML);
23      placeholder.innerHTML = "→ My favorite is: " + water.innerHTML;
24  }
25
26  // Speak outside and display it on page
27  outside.onclick = function() {
28      speak("My favorite is" + outside.innerHTML);
29      placeholder.innerHTML = "→ My favorite is: " + outside.innerHTML;
30  }
31
32  // Speak stretch and display it on page
33  stretch.onclick = function() {
34      speak("My favorite is" + stretch.innerHTML);
35      placeholder.innerHTML = "→ My favorite is: " + stretch.innerHTML;
36  }
```

03.

Speech recognition

How we can talk to the computer

Steps

1.

Declare speech variable

2.

Adjust recognition settings

3.

Write on-click function

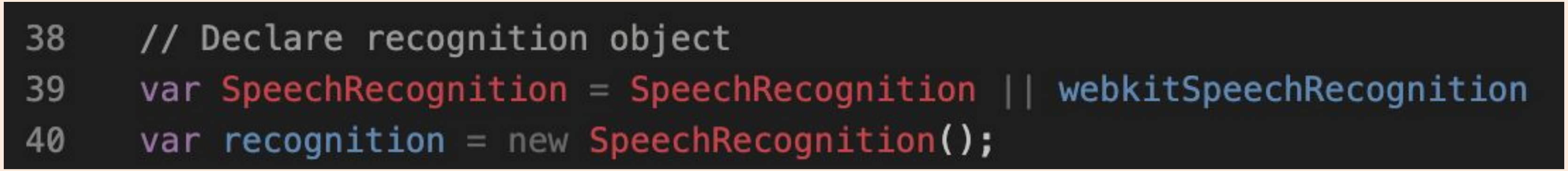
4.

Write on-recognition function

Declare variable

Create and name the speech recognizer

- Lines 38 - 40
- First ensure the browser will recognize it
- Then give it a name, for easy access throughout the code

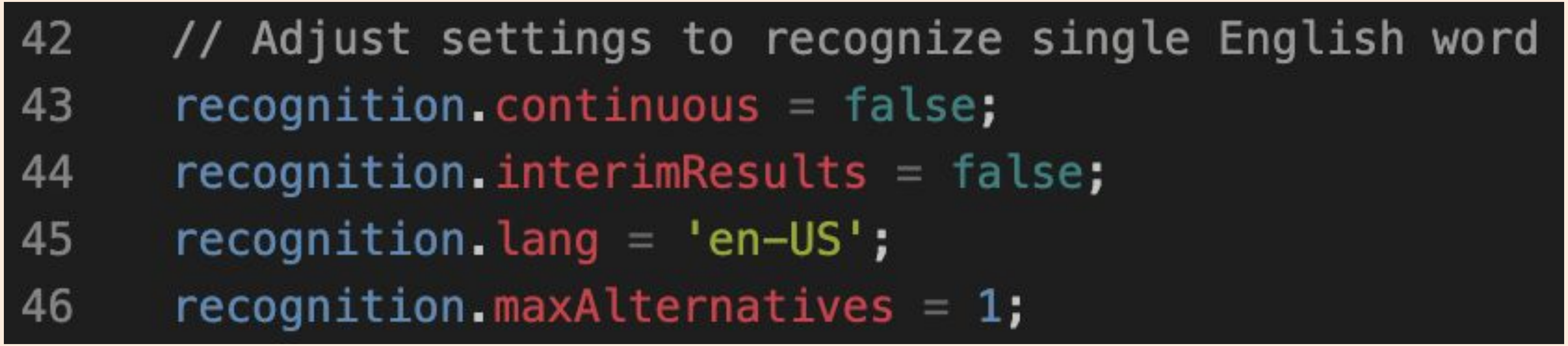


```
38    // Declare recognition object
39    var SpeechRecognition = SpeechRecognition || webkitSpeechRecognition
40    var recognition = new SpeechRecognition();
```

Adjust recognition settings

Set to recognize a single English word

- Lines 42 - 46
- Not continuous (only one word)
- No interim results (only return when the entire word is spoken)
- Set language to English
- If confidence is low, return one alternative

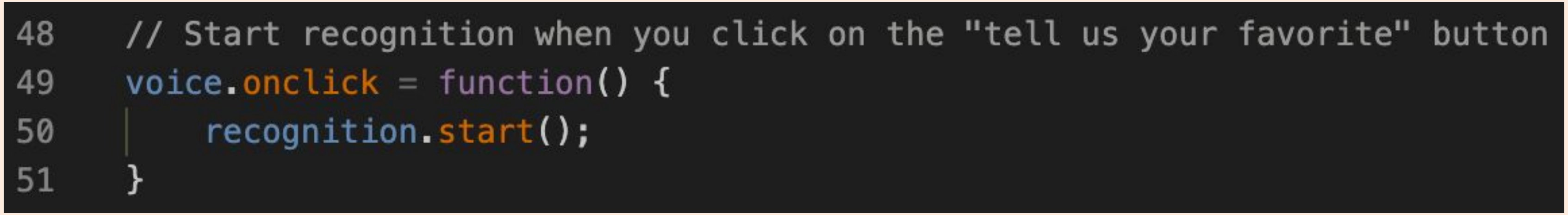


```
42  // Adjust settings to recognize single English word
43  recognition.continuous = false;
44  recognition.interimResults = false;
45  recognition.lang = 'en-US';
46  recognition.maxAlternatives = 1;
```

On-click function

This function executes when we click the “tell us your favorite” button

- Lines 48 - 51
- Call built-in speech recognition method

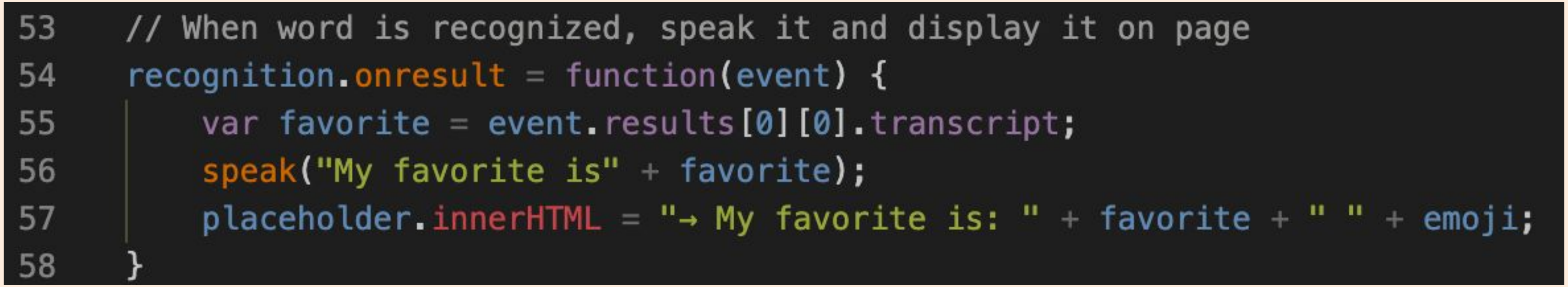


```
48  // Start recognition when you click on the "tell us your favorite" button
49  voice.onclick = function() {
50      recognition.start();
51  }
```

On-recognition function

This function executes once speech recognition returns a result

- Lines 53 - 58
- Declare variable to access result
  - First 0 gets what was recognized
  - Second 0 gets alternative in case of low confidence
- Pass result into the speak function we created earlier
- Display result on page



```
53  // When word is recognized, speak it and display it on page
54  recognition.onresult = function(event) {
55      var favorite = event.results[0][0].transcript;
56      speak("My favorite is" + favorite);
57      placeholder.innerHTML = "→ My favorite is: " + favorite + " " + emoji;
58  }
```

04.

Conclusion

Wrap up & additional resources

Web Speech API drawbacks

The API is built on “**normative**” speakers

- “Normal” = native speakers with no form of speech impediment
- Others may have trouble: the elderly, children, non-native speakers…
- Important to offer different options to users

This technology is still considered **experimental**

- Some functionalities may not work as expected
- However, it is still worth exploring and seeing uses beyond this example

Web Speech API benefits

Can add a human touch to your project

Can make you stand out, without much extra code

Makes your product more **accessible** to certain users

- Someone who has better control over their voice than their hands
- Someone who is multitasking (ex. Walking and using your application)

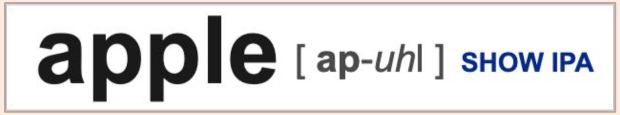
Recognition of “non-standard” speech continues to be improved

- See Google’s Project Euphonia or the startup company Voiceitt

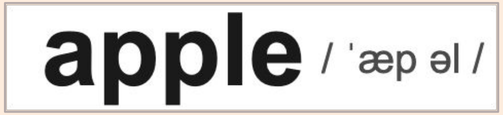
Web Speech API fun fact

If you want the API to produce a **specific sound or sequence**, rather than a full word or sentence:

- It can read **IPA** (international phonetic alphabet)
    - Standardized way to describe sounds produced across all human languages
- Call the speak function with the IPA symbols you desire
- You can find the IPA of English words on dictionary.com:



apple [ap-*uh*l] SHOW IPA



apple / ˈæp əl /

Practice exercises

1. Add more options for favorite self-care habit, or suggestions if they want to choose their own

2. Change the emoji that appears based on which button is clicked

3. Try out different CSS properties such as font styles, borders, or alignment

4. **Challenge**: The on-click functions we wrote for the drink water, go outside, and stretching buttons are very repetitive. Try writing a single function that can be used for all three rather than three repetitive ones. (*Hint: each HTML element will need to be assigned the same class, so the buttons can be accessed as a group*)

Additional resources

Source code for this workshop

Source code for medium & complex examples

Tutorial to build simple speech recognition & speech synthesis applications

Web speech API documentation

Blog post explaining speech recognition and synthesis in the medium example

Tutorial to recognize more than one word

Discussion on how to change the voice of the API

Thanks!

Stay in touch:

graylin@live.unc.edu

https://www.linkedin.com/in/lindsay-greene/

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik