

Speech recognition and text-to-speech functionalities



Vítor Ferreira

Software Developer @ Dialog



MATCH STATS

6	Shots	4
4	On Target	2
1.28	xG	1.57
5	Corners	1
13	Fouls	15
3	Yellow Cards	1
94%	Passes Completed	88%
66%	Possession	34%

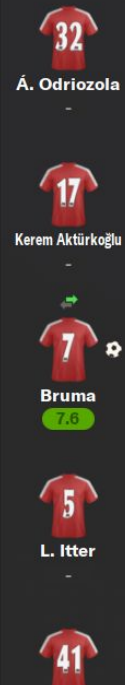
NOTABLE EVENTS

Jorginho	53'	
	54'	Orkun Kökçü (p)
K. Adeyemi	55'	
Jorginho (p)	61'	
	81'	L. Barreiro
B. White	87'	
	90'+4	Bruma
		A. Borza

Arsenal Benfica Post-Match Report



SUBS:



What are we going to
talk about?

Web Speech API

- **SpeechSynthesis (Text-to-Speech)**
- **SpeechRecognition (Asynchronous Speech Recognition)**



Why?

Why should we know more about this?

Because it can be fun!

<https://vitorstick.github.io/moving-de/>



EN to GE

GE to EN

Speak

Translate to Deutsch

text to speak*

Translate it!!

▶ Say it!!

🔄 Say it slowly!!

Because it is fun to work with! But also:

Accessibility: inclusive experiences for users with disabilities.

Chatbots: Who loves doing some vibe coding without touching your keyboard?

Different user experiences: enable app navigation with just voice commands.

And all other things you might consider.

How Web Speech API Works?

Speech Recognition:

Processes audio input from the microphone and converts it into text.

Key Objects:

- `SpeechRecognition`: Primary interface for handling speech recognition features.
- Event Listeners: `onstart`, `onresult`, `onspeechend`, `onerror`, etc.

<https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition>

Speech Synthesis:

Converts text strings into audible speech.

Providing instant auditory feedback, aiding visually impaired users as an example.

- Key Objects:
 - `SpeechSynthesis`: Manages the speech synthesis operations.
 - `SpeechSynthesisUtterance`: Represents the speech request.
 - Event Listeners: `onstart`, `onend`, `onpause`, `onresume`, `onerror`, etc.

<https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis>

Speech Synthesis: Try it out

<https://www.google.com/intl/en/chrome/demos/speech.html>

Browser Support

Limited Browser Support:

- Supported Browsers: Chrome, Firefox (limited), Edge.
- Limitations: Safari and Internet Explorer show limited or no support—consider progressive enhancement.

[https://developer.mozilla.org/en-US/docs/Web/API/Web Speech API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API)

Working in progress

Working in progress: Check it out

<https://webaudio.github.io/web-speech-api/>

[https://github.com/WebAudio/web-speech-api/i
ssues/](https://github.com/WebAudio/web-speech-api/issues/)



Demo!

First part- Listening to the user



First part- Listening to the user

```
// Check if the browser supports SpeechRecognition
const SpeechRecognition =
  (window as any).SpeechRecognition ||
  (window as any).webkitSpeechRecognition;

if (!SpeechRecognition) {
  alert('SpeechRecognition is not supported in this browser.');
```



```
  return;
}

const recognition = new SpeechRecognition();
recognition.lang = languageSelect?.value || 'en-US';
// IF TRUE, THE SPEECH RECOGNITION WILL RETURN INTERIM RESULTS
// RETURNING RESULTS AS THE USER SPEAKS
// IF FALSE, THE SPEECH RECOGNITION WILL RETURN ONLY FINAL RESULTS
recognition.interimResults = false;
// MAXIMUM NUMBER OF ALTERNATIVES TO RETURN
// IF 1, ONLY THE BEST RESULT WILL BE RETURNED
recognition.maxAlternatives = 1;
```

Second part- Synthesis



Second part- Synthesis

```
const SpeechSynthesis = window.speechSynthesis;
let textToSpeak = speechResultDiv?.textContent || 'Hello, how are you?';
const utterance = new SpeechSynthesisUtterance(textToSpeak);
utterance.lang = languageSelect?.value || 'en-US';
// Controls the speed at which the text is spoken
// Values range from 0.1 (very slow) to 10 (very fast)
utterance.rate = 1;
// Adjusts the pitch of the voice
// Changing pitch can make the voice sound deeper or higher
utterance.pitch = 1;
```


Third part- Interacting (Impress your colleagues)



Third part- Interacting (Impress your colleagues)

Let's make the color of our website
change on voice command

Accessibility and the Web Speech API

Enhancing Accessibility with WCAG and ARIA in Web Speech API

- **WCAG:** Web Content Accessibility Guidelines (WCAG) are a set of recommendations to make web content more accessible to people with disabilities.
- **ARIA:** Accessible Rich Internet Applications (ARIA) attributes improve the accessibility of web pages by providing additional semantic information for assistive technologies.

WCAG Principles Related to Web Speech

- **Perceivable:** Information and user interface components must be presentable to users in ways they can perceive.
 - Use speech synthesis to audibly read out content.
- **Operable:** Users must be able to operate the interface.
 - Enable voice commands to navigate and interact with the application.

WCAG Principles Related to Web Speech

- **aria-live:** Indicates that part of the page will be updated dynamically and that these updates need to be announced to the user.
 - Example: When using speech recognition, use polite for less critical updates and assertive for more immediate content changes.

```
</div>  
<div class="card">  
  <label for="speechResult">Speech Result:</label>  
  <div id="speechResult" aria-live="assertive"></div>  
</div>  
</section>
```

WCAG Principles Related to Web Speech

- **aria-label & aria-labelledby:** Provide labels for interactive elements without visible text labels.

```
<!-- SECOND PART: SPEAKING TO THE USER -->
<section class="card" aria-labelledby="talk-heading">
  <h2 id="talk-heading">Talk to me!</h2>
  <div class="card">
    <button id="talk" type="button" aria-label="Start speech synthesis to talk to the user">
      Start talking!
    </button>
  </div>
</section>
```

WCAG Principles Related to Web Speech

- **aria-controls:** Indicates the relationship between elements, useful for specifying which part of the page a speech control button affects.

```
<section class="card" aria-labelledby="color-change-heading">
  <h2 id="color-change-heading">Change my website color!</h2>
  <div class="card">
    <button id="colorChange" type="button" aria-controls="colorResult">
      Change color!
    </button>
  </div>
  <label for="colorResult">Color: </label>
  <div id="colorResult" aria-live="polite"></div>
</section>
```

Accessibility tools

The screenshot shows the Chrome DevTools interface with the Accessibility panel open. The left pane displays the DOM tree, and the right pane shows the Accessibility tree and ARIA attributes.

DOM Tree (Left Pane):

```
<!DOCTYPE html>
<html lang="en">
  <head>
  </head>
  <body>
    <div id="app">
      <main>
        <!-- FIRST PART: LISTENING TO THE USER -->
        <section class="card" aria-labelledby="listen-heading">
        </section>
        <!-- SECOND PART: SPEAKING TO THE USER -->
        <section class="card" aria-labelledby="talk-heading">
        </section>
        <!-- THIRD PART: CHANGE MY WEBSITE COLOR -->
        <section class="card" aria-labelledby="color-change-heading">
          <h2 id="color-change-heading">Change my website color!</h2>
          <div class="card">
            <button id="colorChange" type="button" aria-controls="colorResult">Change color!</button>
          </div>
          <label for="colorResult">Color: </label>
          <div id="colorResult" aria-live="polite"></div>
        </section>
      </main>
    </div>
    <script type="module" src="/src/main.ts?t=1746198615292"></script>
    <deep1-input-controller translate="no">
    </deep1-input-controller>
    <div id="awebChromeHelper" data-version="1.24.4">
    </div>
  </body>
  <rq-implicit-test-rule-widget class="rq-element" draggable="true" style="display: none;">
  </rq-implicit-test-rule-widget>
</html>
```

Accessibility Panel (Right Pane):

- Accessibility Tree:**
 - ☒ Enable full-page accessibility tree
 - The accessibility tree moved to the top right corner of the DOM tree.
[Send us your feedback.](#)
- ARIA Attributes:**
 - aria-live: polite
- Computed Properties:**
 - Name: ""
 - aria-labelledby: Not specified
 - aria-label: Not specified
 - Role: generic
 - Live region: polite
 - Atomic (live regions): false
 - Relevant (live regions): additions text
- Source Order Viewer:**
 - No source order information available


Windows - Narrator

Narrator


— □ ×

Welcome to Narrator


This is Narrator Home, where you can get help, access your settings, and learn about new features. Narrator is a screen reader that describes aloud what's on your screen, so you can use that information to navigate your device. To start or stop Narrator, press the **Windows logo key + Ctrl + Enter**. Explore the sections below to get started.




QuickStart
Learn the basics of Narrator.




Narrator guide
View the complete Narrator guide online.



What's new
Get an overview of new and updated features.



Settings
Customise Narrator. Press Windows logo key + Ctrl + N to access settings anytime.



Feedback
Help improve Narrator. Press Narrator key + Alt + F to give feedback anytime.

☒ Show Narrator Home when Narrator starts

[Start Narrator after sign-in](#)

Exit Narrator

Minimise



Demo!
But now in Angular

Alternatives

Alternatives

<https://aws.amazon.com/polly/>

<https://elevenlabs.io/>

Just google it, there are a lot of them



Q & A