

The background features a large, glowing circular portal with a white outline, centered in the frame. It is set against a dark, futuristic landscape with jagged, crystalline structures in shades of purple, blue, and red. The ground has a grid-like pattern, suggesting a digital or virtual environment. The overall atmosphere is mysterious and tech-oriented.

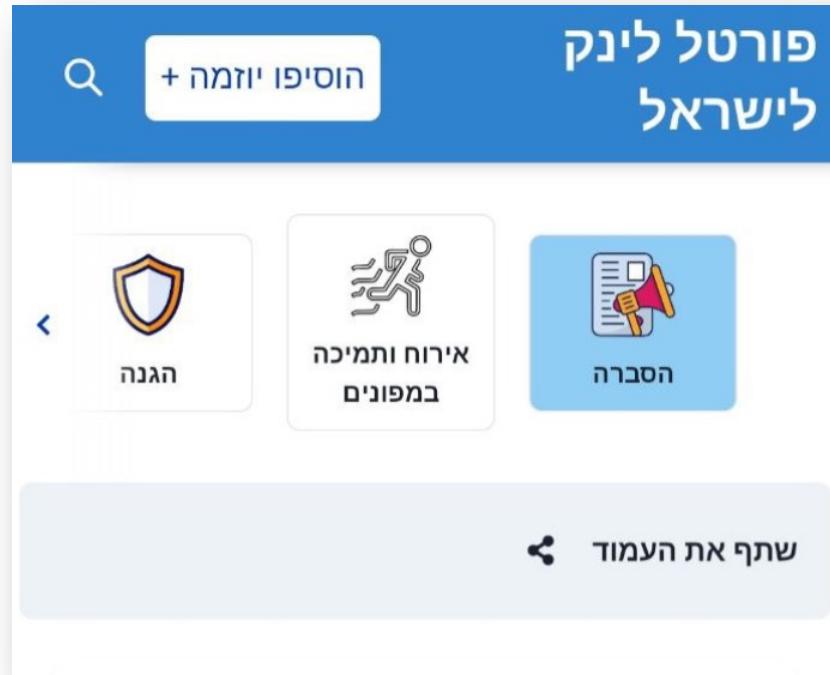
Unlocking JavaScript's Hidden Gems

About me

- 👩‍🦰👩‍🦰👩‍🦰👩‍🦰inja 🐕🦜
- **Loving Web Tech**
- **Love The Community**
- **<https://yedri.medium.com/>**
- **Developer @ CyberArk**



Once upon a time...



Once upon a time...



17:12 4G+ 48%

linksforisrael.com/

פורטל לינק
ישראל

הוסיפו יוזמה +

ארוחות ותמיכה
בمفهوم

הסברת

שתף את העמוד

ראשי

CWNCERT

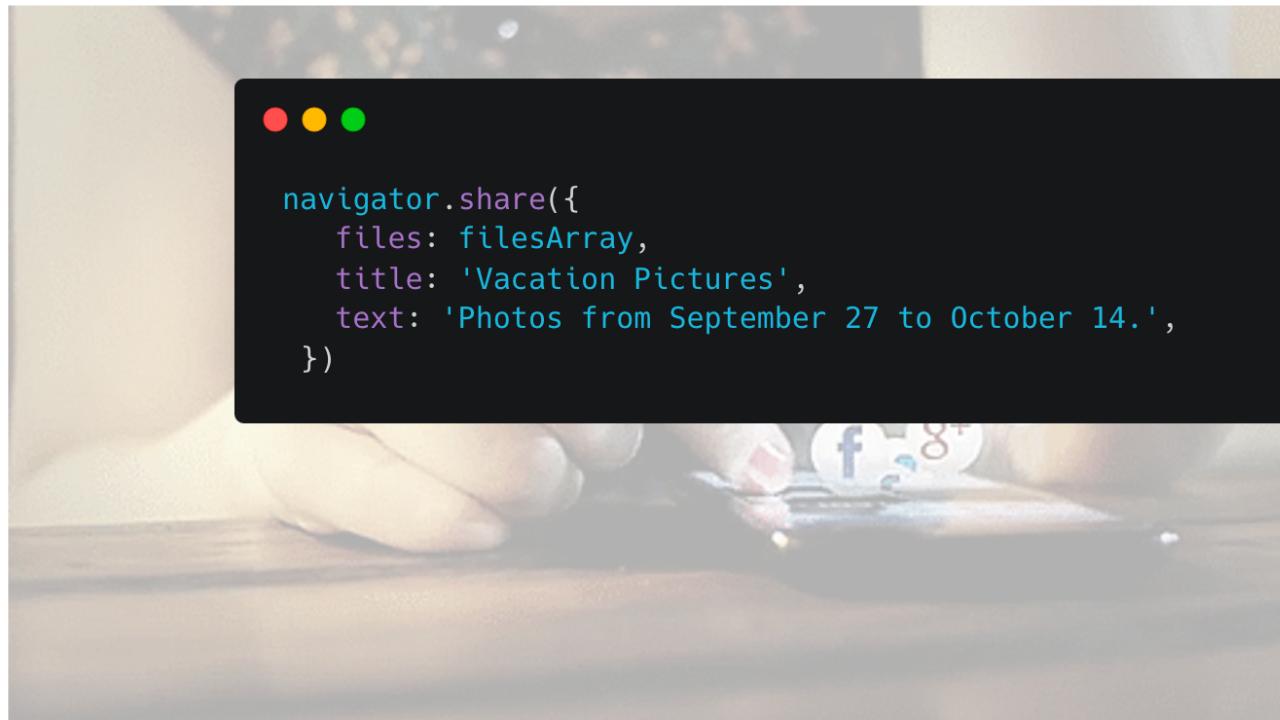
קולות ישראל -
Voice Of Israel
గוף תומך בפעילות
הסברתיות
פּוֹרִישָׁרָאֵלִית מַסְבִּי...

הופכים עליהם,
טובים אותם

in X Quora f Instagram

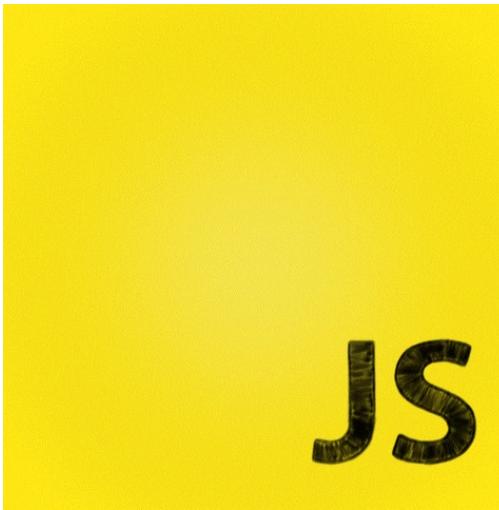


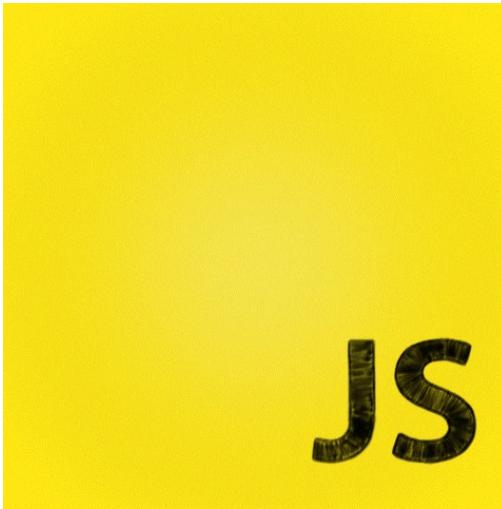
Once upon a time...



Disclaimer







Unfamiliar API in Web Development

Notifications API

Notifications API

```
● ● ●  
  
const notification = new Notification("Hi reversim 2024", {  
  badge: "revesim.com",  
  body: "reversim 2024 - notification api",  
  // vibrate: [200, 100, 200],  
  // actions: [{ action: "get", title: "Get now." }], - in SW  
  icon: "https://encrypted-tbn0.gstatic.com/images?  
q=tbn:ANd9GcTVyhreu7NLdX0FyxsJ_iZVZWp0pEUcrZjqN4hSmk1_sw8WxhDaI-  
f9Bd057vARhwX7wFA&usqp=CAU",  
});
```

Demo

(: זה דוגמא לנותיפיקציה

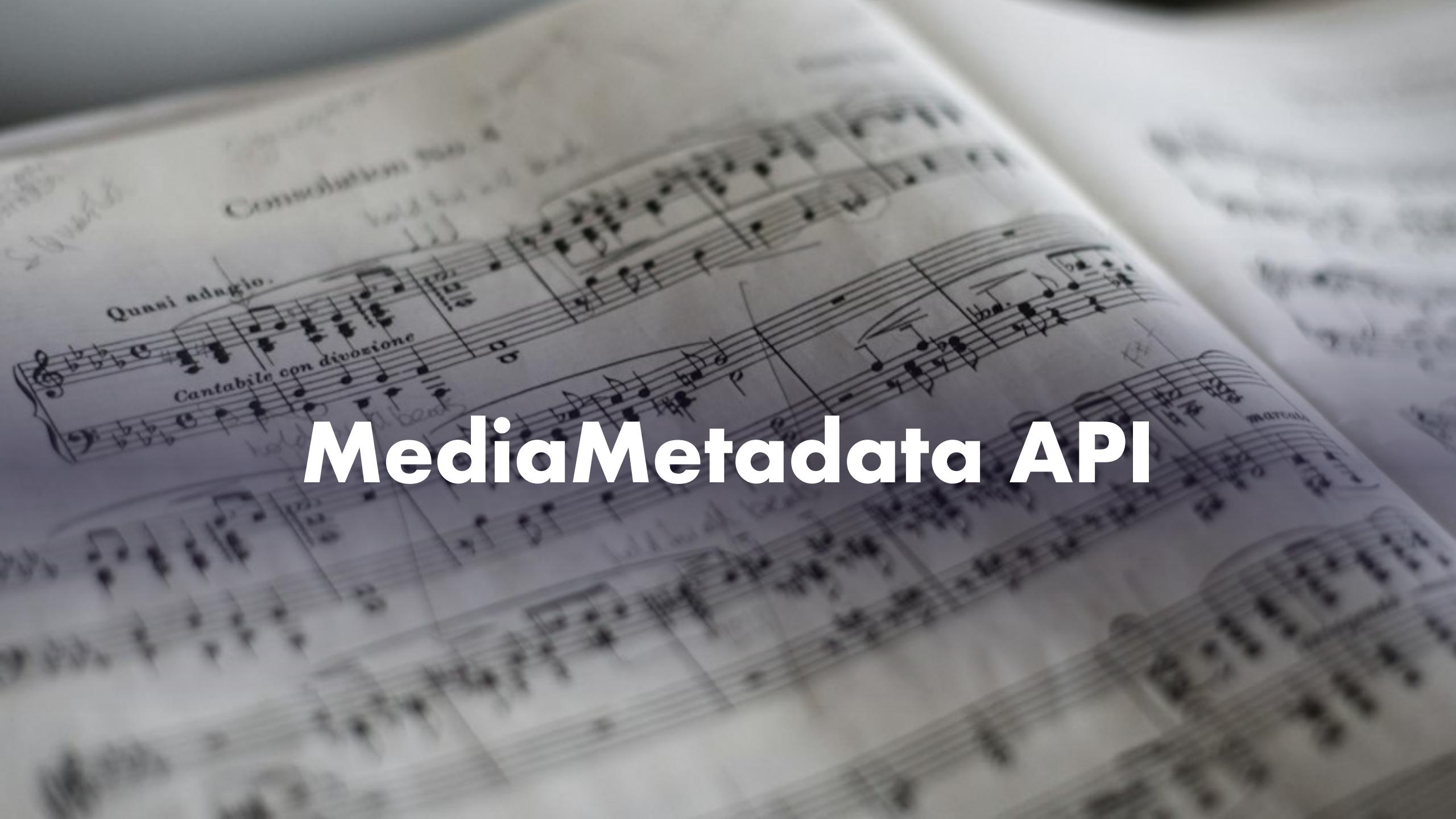
Notify me!



Notifications API

Demo

MediaMetadata API



MediaMetadata API



```
navigator.mediaSession.metadata =  
new MediaMetadata({  
  title: "Unlocking JavaScript's Hidden Gems",  
  artist: 'Yanai Edri',  
  album: 'Reversim submit 2024'  
})
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Sun 4 Feb 21:09

Rick Astley - Never Gonna Give You Up

youtube.com/watch?v=dQw4w9WgXcQ

Apps Testing Playground... Business Conference... AppGyver Lint Images CyberArk - Dynam... Torrent Finder :: To... CyberArk - Dynam... My Benefits Netlify All Bookmarks

YouTube

הענלה (k)

0:00 / 3:32

RICK ASTLEY

desianrr

Rick Astley - Never Gonna Give You Up (Official Music Video)

Elements Console Sources Performance insights Network Performance Memory Application Security Lighthouse > 30 2 4 3 hidden

Filter Default levels 51 Issues: 2 45 4 8 verbose

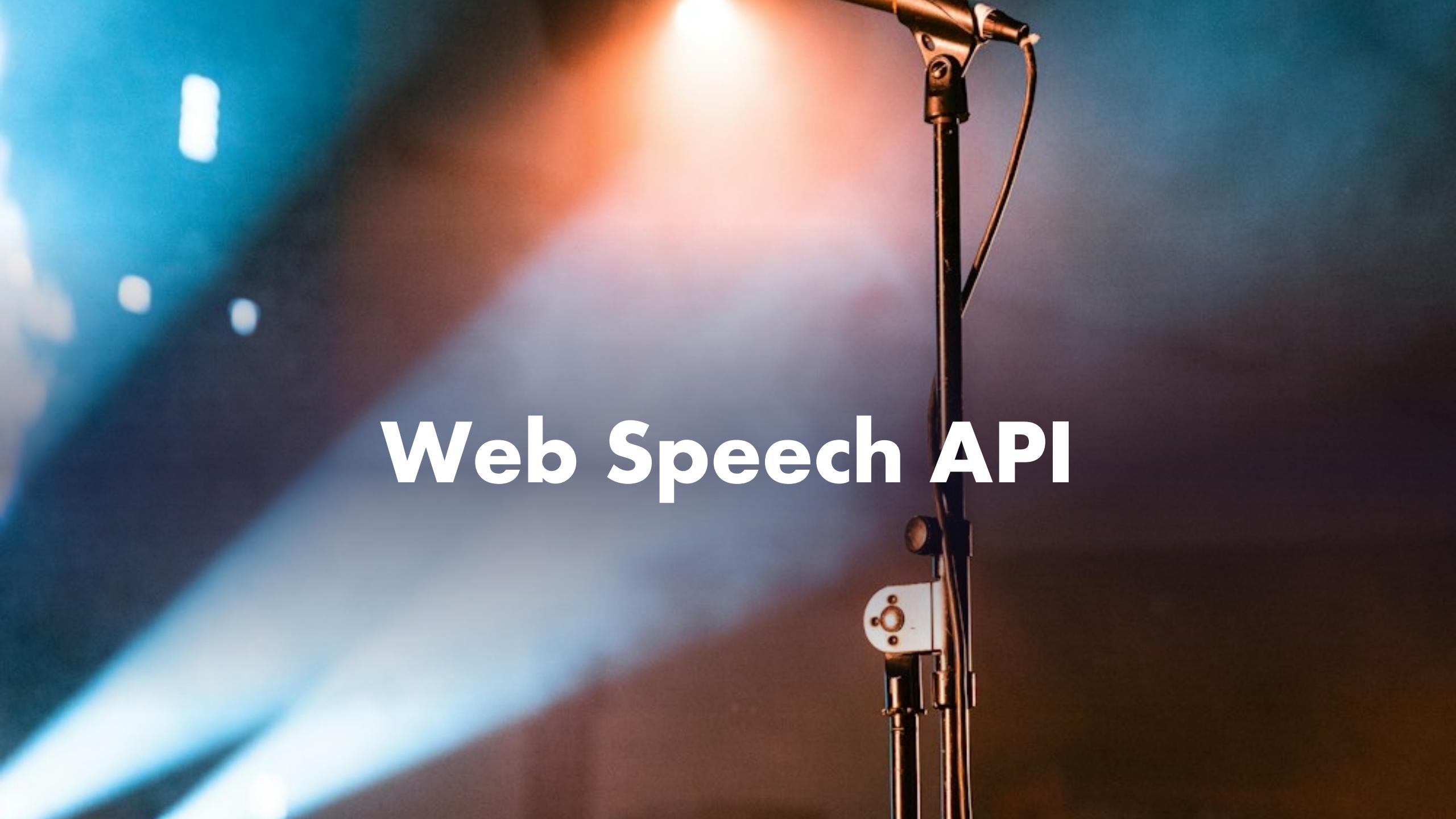
38 messages

- No user messages
- No errors
- 30 warnings
- No info

8 verbose

The screenshot shows a Chrome browser window with a YouTube video playing. The video frame shows Rick Astley singing into a microphone. Below the video is a standard YouTube control bar with play/pause, volume, and timestamp (0:00 / 3:32). The browser's address bar displays the URL youtube.com/watch?v=dQw4w9WgXcQ. The top menu bar includes options like File, Edit, View, History, Bookmarks, Profiles, Tab, Window, and Help. The system tray at the top right shows various icons for battery, signal, and system status. The developer tools console is open at the bottom, showing 51 issues with 30 warnings. The warning list includes: 38 messages, No user messages, No errors, 30 warnings, and No info. The 'Console' tab is selected in the developer tools.

Web Speech API



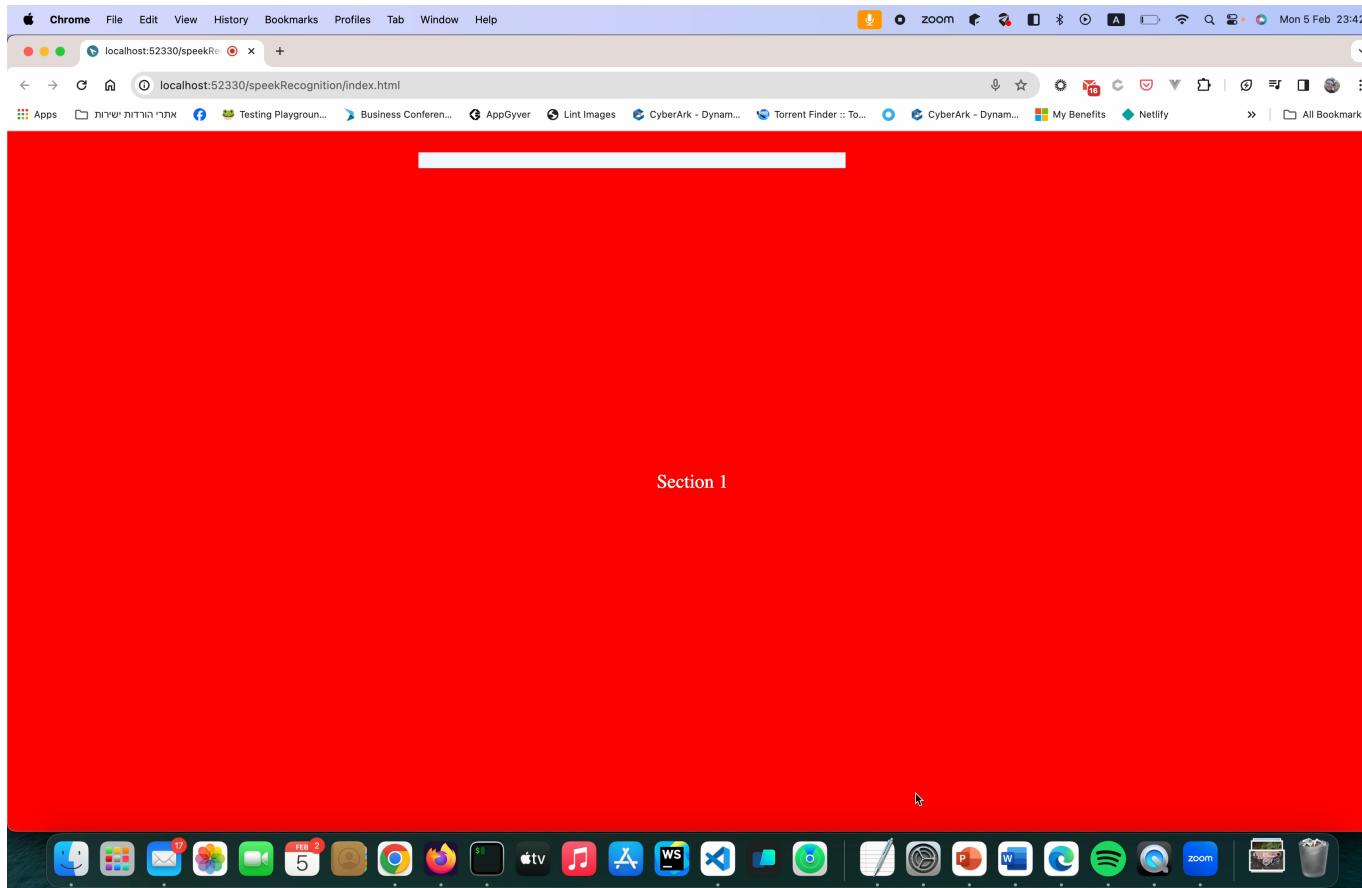
Web Speech API

Speech recognition + Speech Grammar



```
const recognition = new SpeechRecognition();
const speechRecognitionList = new SpeechGrammarList();
```

Web Speech API



[Demo](#)

Web Speech API

SpeechSynthesis



```
let utterance = new SpeechSynthesisUtterance("Hello world!");
speechSynthesis.speak(utterance);
```

Speech synthesiser

Enter some text in the input below and press return or the "play" button to hear it. change voices using the dropdown menu.

Enter text

Rate 1.5

Pitch 0.6

Carmit (Hebrew)

hi all

שלום לכל בא ריוורביס, אתם תראו מה דברים שלא חשבתם שקיימים... מהנו
הי אנשים, אתם תראו מה דברים שלא חשבתם שקיימים... מהנו

DocumentPictureInPicture API



DocumentPictureInPicture API

```
● ● ●  
  
const videoPlayer = document.getElementById("player");  
  
// Open a Picture-in-Picture window.  
const pipWindow = await window.documentPictureInPicture.requestWindow({  
  width: videoPlayer.clientWidth,  
  height: videoPlayer.clientHeight,  
});  
  
// ...
```

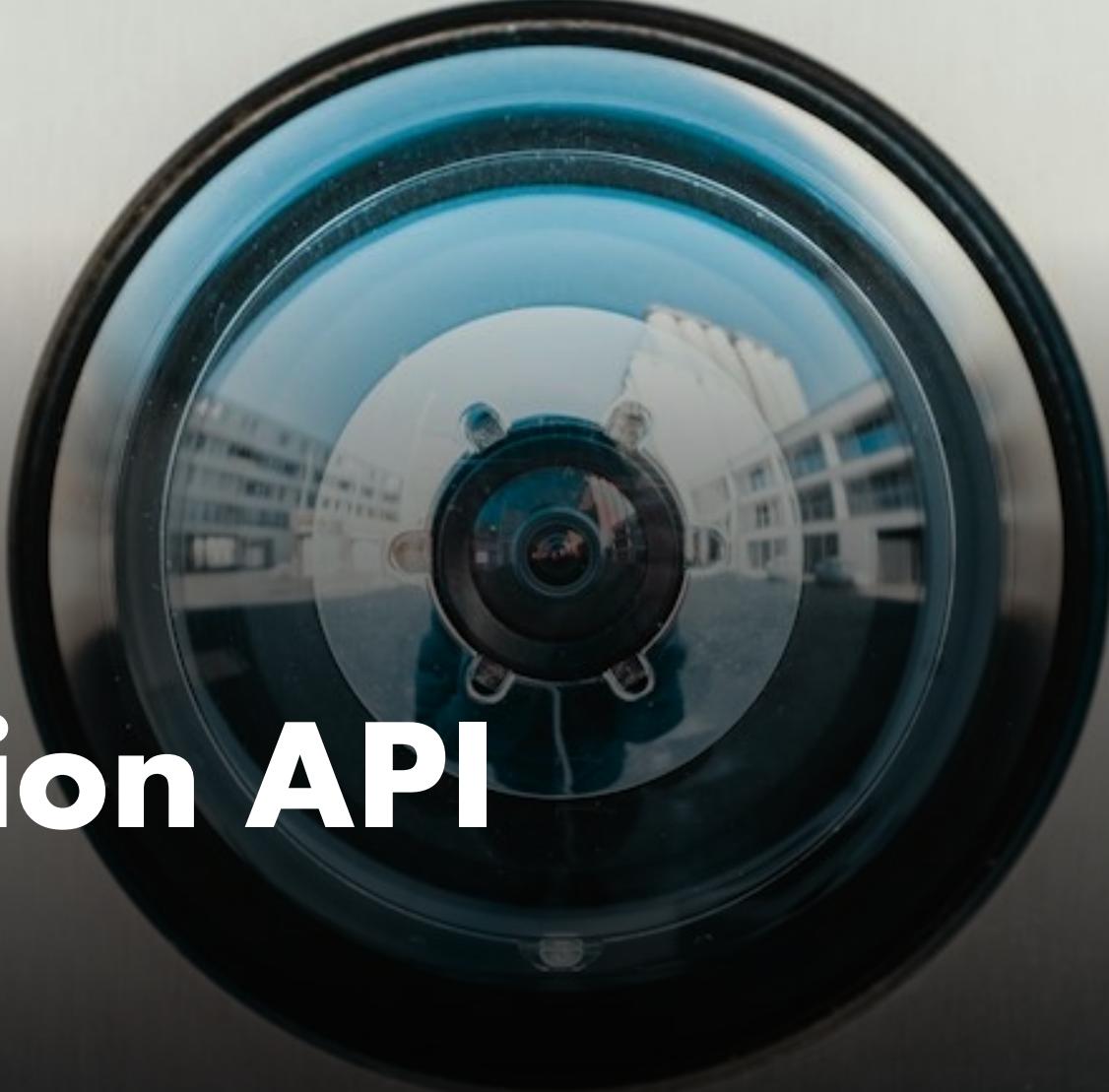
[Demo](#) [Demo](#)

*Not only for Video!!!

DocumentPictureInPicture

API



A close-up photograph of a black circular camera lens. The lens has a multi-layered blue and black anti-reflective coating. In the center of the lens, the brand name "SARIN" is visible. The lens is mounted on a silver-colored body, which is partially visible at the top and bottom. The background is a plain, light-colored wall.

Detection API

Shape & Face Detection API

<https://chromestatus.com/features#shape>

For Stream Or Static Image

Demo



Barcode Detection API

```
console.log( await BarcodeDetector.getSupportedFormats())

▼ (11) ['aztec', 'code_128', 'code_39', 'code_93', 'data_matrix', 'ean_13', 'ean_8', 'itf', 'pdf417', 'qr_code', 'upc_a', 'upc_e', 'unknown']
  0: "aztec"
  1: "code_128"
  2: "code_39"
```

Format	Description	Image
aztec	A square two-dimensional matrix following iso24778 and with a square bullseye pattern at their center, thus resembling an Aztec pyramid. Does not require a surrounding blank zone.	
code_128	A linear (one-dimensional), bidirectionally-decodable, self-checking barcode following iso15417 and able to encode all 128 characters of ASCII (hence the naming).	
code_39	A linear (one-dimensional), self-checking barcode following iso16388. It is a discrete and variable-length barcode type.	
code_93	A linear, continuous symbology with a variable length following bcs. It offers a larger information density than Code 128 and the visually similar Code 39. Code 93 is used primarily by Canada Post to encode supplementary delivery information.	
codabar	A linear barcode representing characters 0-9, A-D and symbols - . \$ / +	
data_matrix	An orientation-independent two-dimensional barcode composed of black and white modules arranged in either a square or rectangular pattern following iso16022.	
ean_13	A linear barcode based on the UPC-A standard and defined in iso15420.	
ean_8	A linear barcode defined in iso15420 and derived from EAN-13.	
itf	A continuous, self-checking, bidirectionally-decodable barcode. It will always encode 14 digits.	
pdf417	A continuous two-dimensional barcode symbology format with multiple rows and columns. It's bidirectionally decodable and uses the iso15438 standard.	
qr_code	A two-dimensional barcode that uses the iso18004 standard. The information encoded can be text, URL or other data.	
upc_a	One of the most common linear barcode types and is widely applied to retail in the United States. Defined in iso15420, it represents digits by strips of bars and spaces, each digit being associated to a unique pattern of 2 bars and 2 spaces, both of variable width. UPC-A can encode 12 digits that are uniquely assigned to each trade item, and it's technically a subset of EAN-13 (UPC-A codes are represented in EAN-13 with the first character set to 0).	
upc_e	A variation of UPC-A defined in iso15420, compressing out unnecessary zeros for a more compact barcode.	
unknown	This value is used by the platform to signify that it does not know or specify which barcode format is being detected or supported.	

Barcode Detection API

```
const barcodeDetector = new BarcodeDetector();
const image = document.querySelector("img");

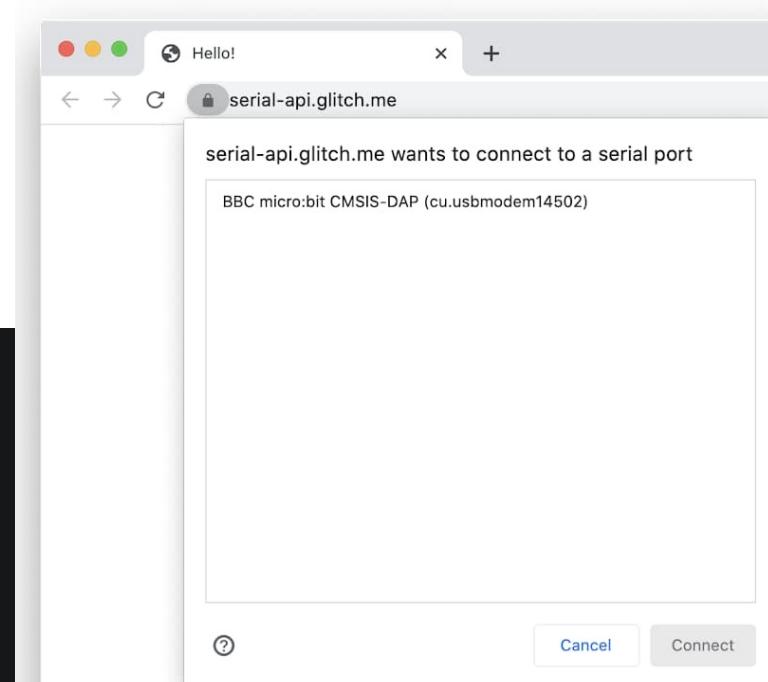
barcodeDetector
  .detect(image)
  .then((barcodes) => {
    let pre = document.createElement("pre");
    pre.innerHTML = JSON.stringify(barcodes, null, 2);
    image.after(pre);
  })
  .catch(console.error);
```

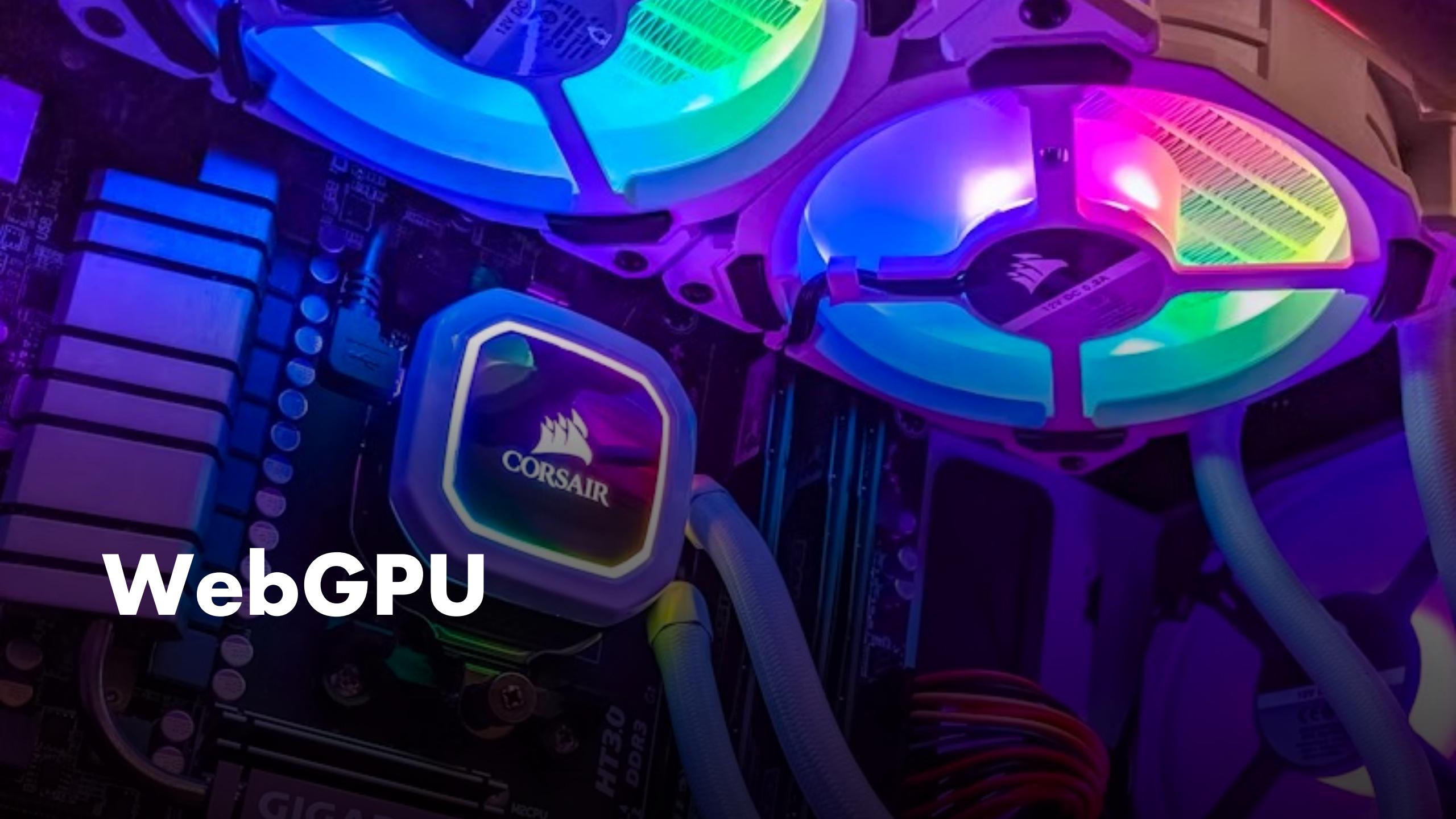


```
[  
  {  
    "boundingBox": {  
      "x": 11.5,  
      "y": 11.5,  
      "width": 175,  
      "height": 175,  
      "top": 11.5,  
      "right": 186.5,  
      "bottom": 186.5,  
      "left": 11.5  
    },  
    "cornerPoints": [  
      {  
        "x": 11.5,  
        "y": 11.5  
      },  
      {  
        "x": 186.5,  
        "y": 11.5  
      },  
      {  
        "x": 186.5,  
        "y": 186.5  
      },  
      {  
        "x": 11.5,  
        "y": 186.5  
      }  
    "format": "qr_code",  
    "rawValue": "https://daily-dev-tips.com"  
]
```

WebUSB API + Web Serial API

```
// Filter on devices with the Arduino Uno USB Vendor/Product IDs.  
const filters = [  
  { usbVendorId: 0x2341, usbProductId: 0x0043 },  
  { usbVendorId: 0x2341, usbProductId: 0x0001 }  
];  
  
// Prompt user to select an Arduino Uno device.  
const port = await navigator.serial.requestPort({ filters });  
  
const { usbProductId, usbVendorId } = port.getInfo();
```





WebGPU

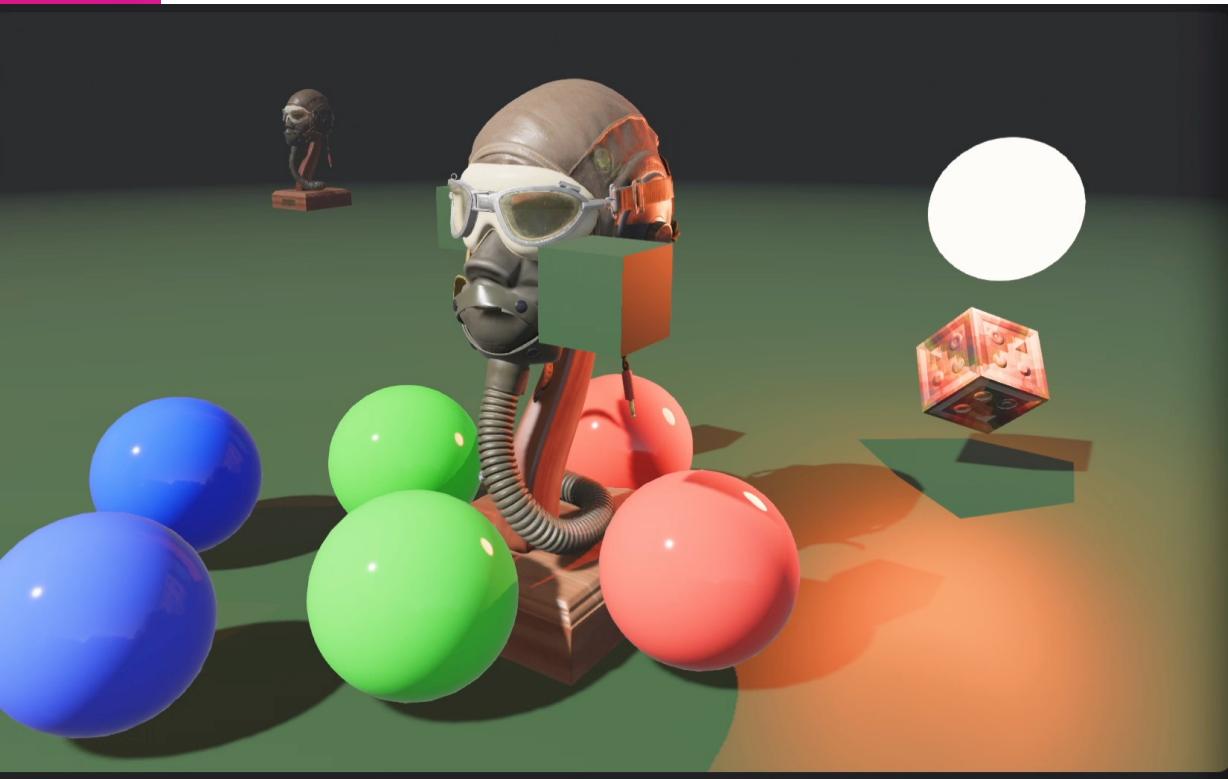
WebGPU



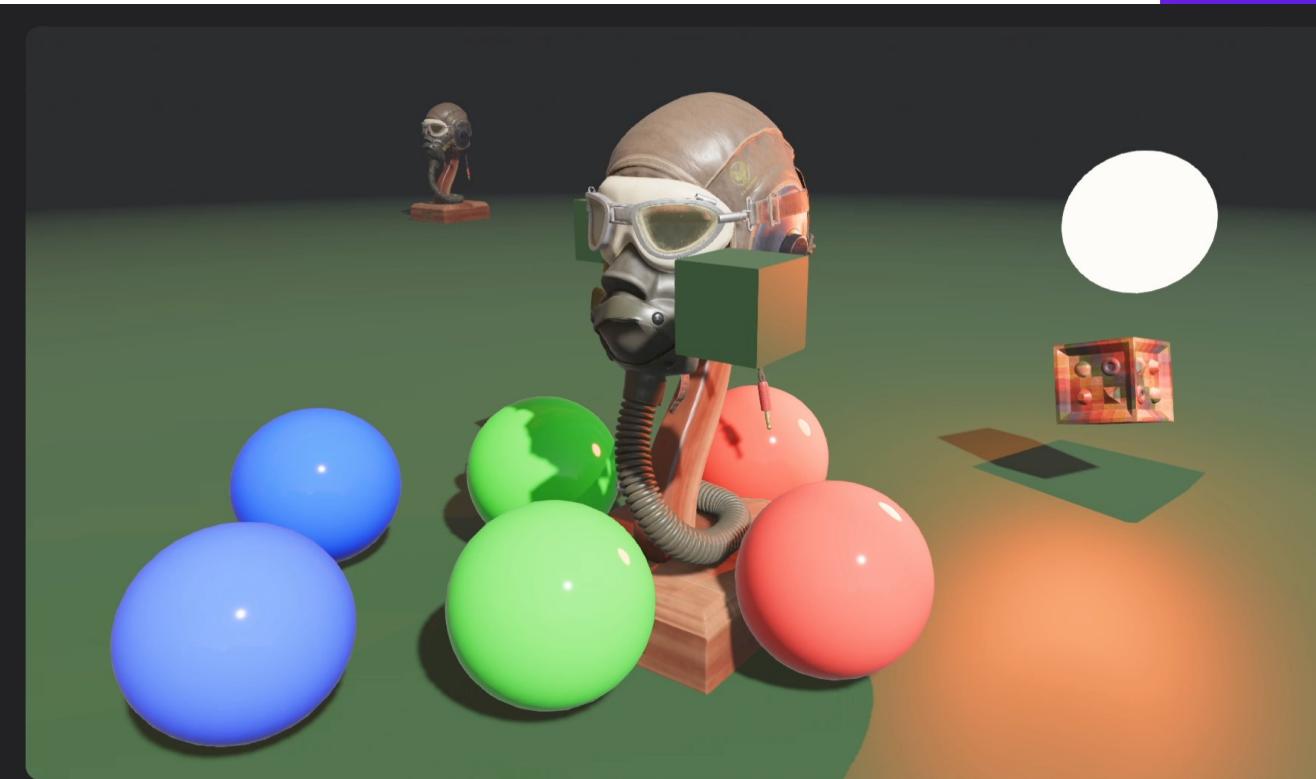
WebGPU



WebGl₂



WebGPU



A man with dark hair and a beard is wearing a white VR headset. He is pointing his right index finger forward towards the camera. He is wearing a blue zip-up hoodie. The background is blurred, showing what appears to be a city street.

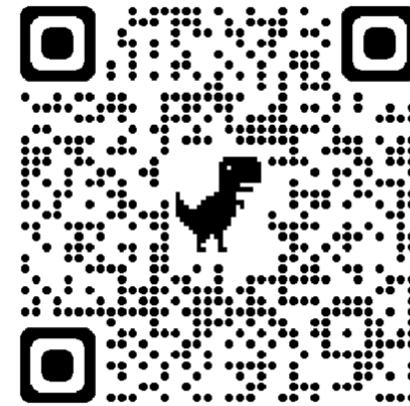
WebXR



WebXR Augmented Reality



```
// VR  
navigator.xr.requestSession("immersive-vr")  
// AR  
navigator.xr.requestSession("immersive-ar")  
// Web Device only  
navigator.xr.requestSession("inline")
```



Web demo:

<https://modelviewer.dev/examples/augmentedreality/index.html>

Mobile Demo:

<https://immersive-web.github.io/webxr-samples/>

UserActivation API



UserActivation API

```
● ● ●  
if (navigator.userActivation.isActive) {  
    // proceed to request playing media, for example  
}  
  
if (navigator.userActivation.hasBeenActive) {  
    // proceed with auto-playing an animation, for example  
}
```

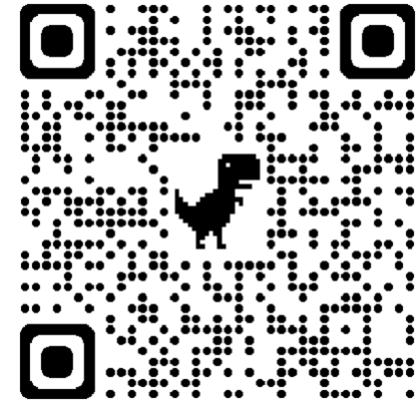


Gamepad API



Gamepad API

```
window.addEventListener( "gamepadconnected", (e) => {
  console.log(
    "Gamepad connected at index %d: %s. %d buttons, %d axes.",
    e.gamepad.index,
    e.gamepad.id,
    e.gamepad.buttons.length,
    e.gamepad.axes.length,
  );
});
```



Demo

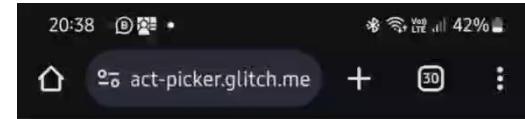
Demo

A white telephone handset is coiled on a blue surface. The handset is white with a black speaker grille and a small button. The cord is a white spiral cord.

Contact Picker API

Contact Picker API

```
● ● ●  
const props = ["name", "email", "tel", "address", "icon"];  
const opts = { multiple: true };  
...  
const contacts = await navigator.contacts.select(props, opts);
```



Contact Picker API Demo

Access to the user's contacts has been a feature of native apps since (almost) the dawn of time. The **Contact Picker API** is a new, on-demand picker that allows users to select an entry or entries from their contact list and share limited details of the selected contact(s) with a website. It allows users to share only what they want, when they want, and makes it easier for users to reach and connect with their friends and family.

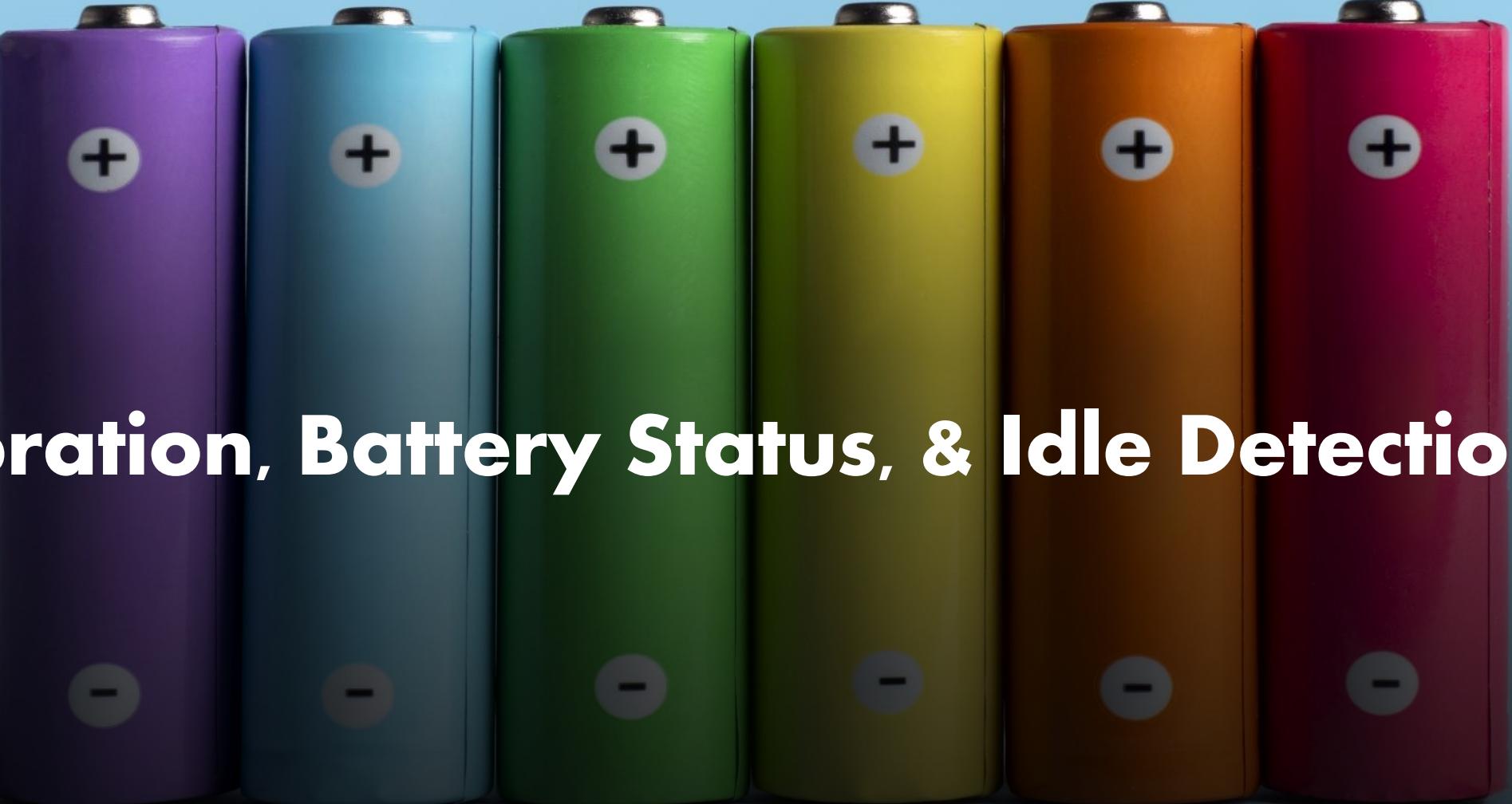
Note: This demo requires Chrome 80 or later running on Android M or later to work properly.

- Multiple results?
- Include name?
- Include email addresses?
- Include telephone numbers?
- Include addresses? (Chrome 84 and later)
- Include icons? (Chrome 84 and later)

This demo **does not** share or upload your contacts. It simply shows them on screen, and no contact data is transferred off this device.

[Open contact picker](#)



A row of six AA batteries is arranged horizontally against a blue-to-black gradient background. The batteries are colored in a spectrum: purple, teal, green, yellow, orange, and red. Each battery has a silver top with a black plus sign (+) and a grey bottom with a black minus sign (-).

Vibration, Battery Status, & Idle Detection

EyeDropper API



EyeDropper API

```
const eyeDropper = new EyeDropper();

eyeDropper
  .open()
  .then((result) => {
    resultElement.textContent = result.sRGBHex;
    resultElement.style.backgroundColor = result.sRGBHex;
  })
  .catch((e) => {
    resultElement.textContent = e;
  });

```



Web Audio & Web MID



REVERSUM SUMMIT

My browser is a musical instrument?

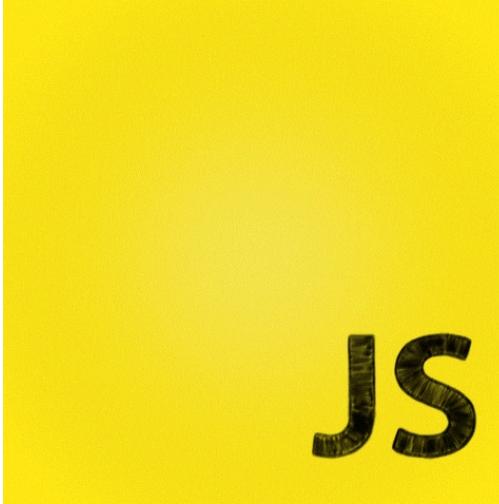
An introduction to digital sound & Web Audio API

Marik Shnitman / Mobile Developer
WIX Engineering

mariks@wix.com @Marik_Sh linkedin/mariksh github.com/marik22312



Web Audio & Web MID



JS



Mobile and PWA

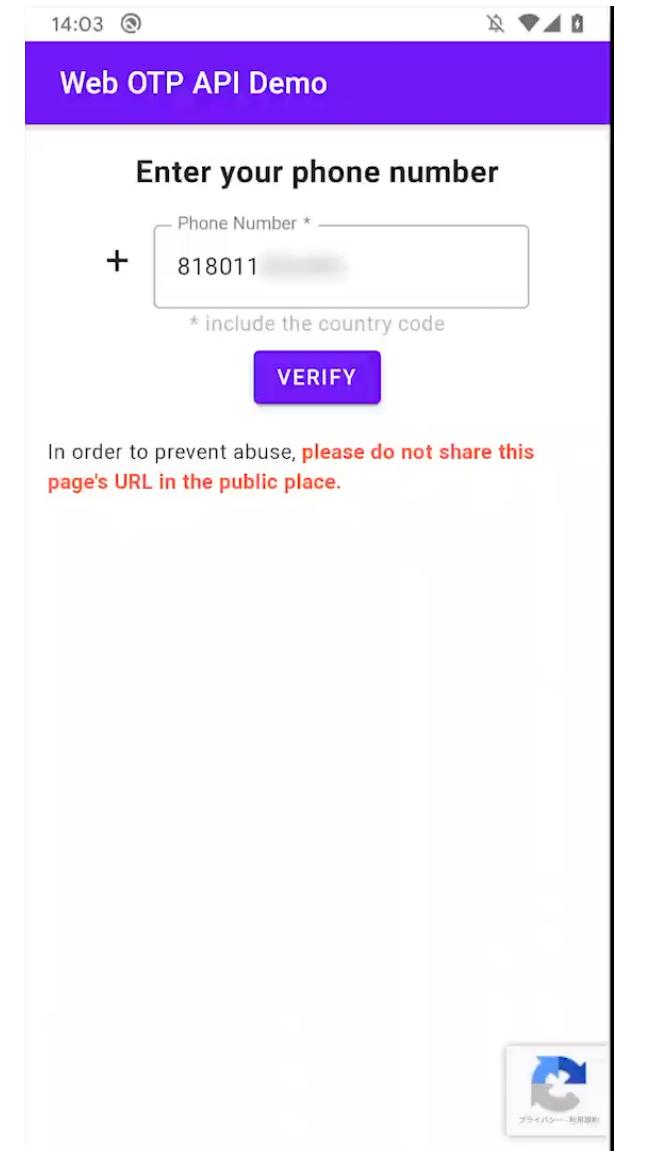
WebO(ne)T(ime)Password API

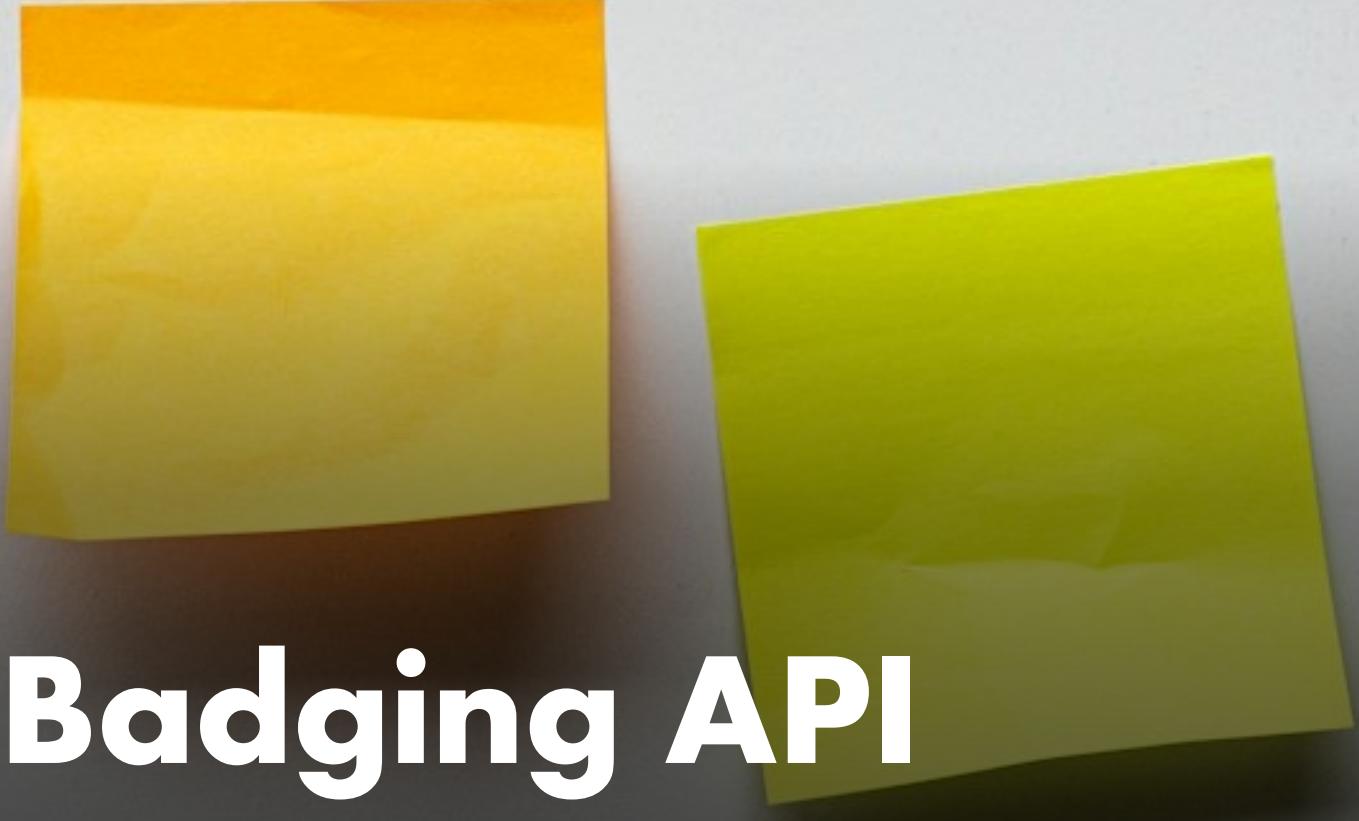


WebOTP API

```
...  
navigator.credentials.get({  
  otp: { transport:['sms'] }  
...  
}).then(otp => {  
  ...  
}
```

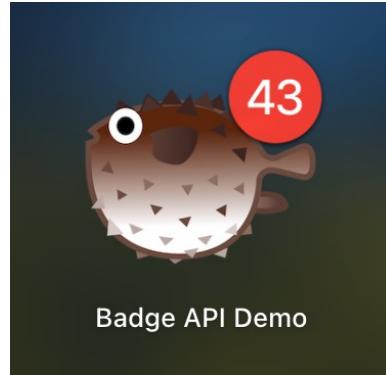
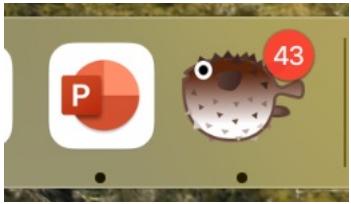
Demo



A photograph of two yellow sticky notes on a dark, textured surface. One note is positioned vertically on the left, and the other is positioned horizontally to its right, overlapping slightly.

App Badging API

App Badging API



Customize- install



Customize- install

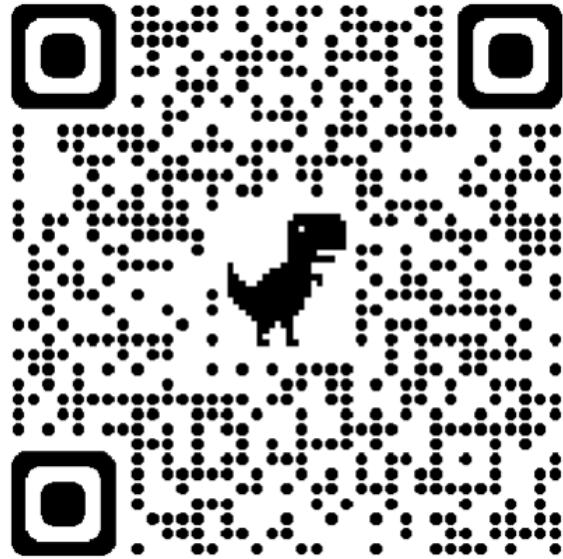


```
window.addEventListener('beforeinstallprompt', (e) => {
  // Prevent the mini-infobar from appearing on mobile
  e.preventDefault();
  // Stash the event so it can be triggered later.
  deferredPrompt = e;
  // Update UI notify the user they can install the PWA
  showInstallPromotion();
  // Optionally, send analytics event that PWA install promo was shown.
  console.log(`'beforeinstallprompt' event was fired.`);
});
```



```
window.addEventListener('appinstalled', () => {
  // Hide the app-provided install promotion
  hideInstallPromotion();
  // Clear the deferredPrompt so it can be garbage collected
  deferredPrompt = null;
  // Optionally, send analytics event to indicate successful install
  console.log('PWA was installed');
});
```

WakeLock API



<https://mdn.github.io/dom-examples/screen-wake-lock-api/>



Web Bluetooth



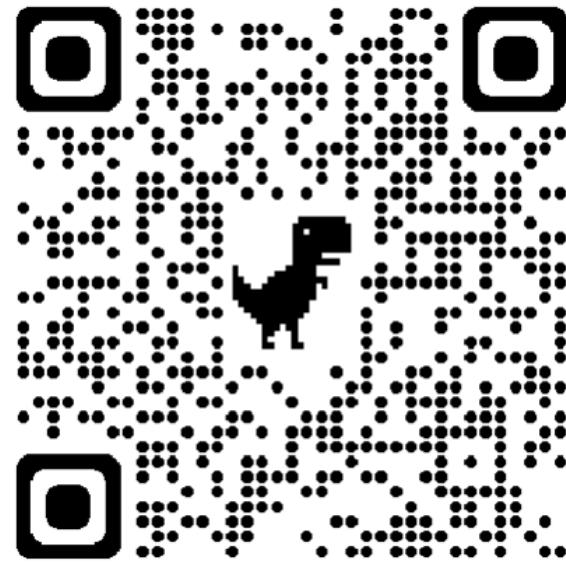
<https://googlechrome.github.io/samples/web-bluetooth/device-info.html>

Web NFC



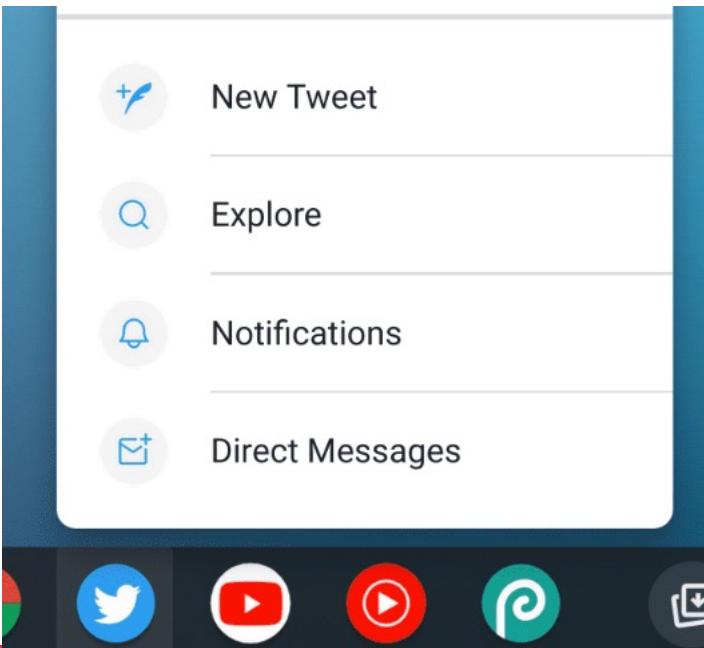
<https://labs-web-nfc.glitch.me/>

Multi-touch API

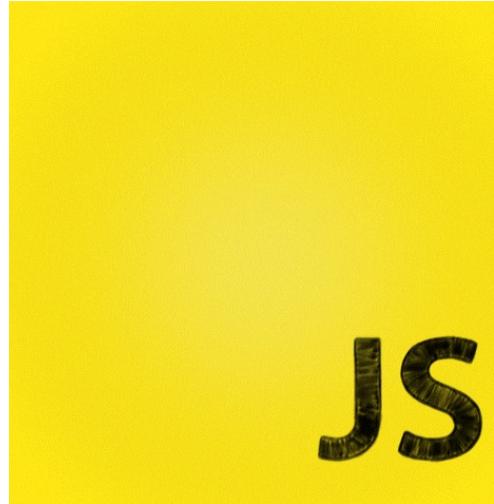


<https://codepen.io/Suniaster/pen/qBmEzOW>

App Shortcuts



```
{  
  "name": "Player FM",  
  "start_url": "https://player.fm?utm_source=homescreen",  
  ...  
  "shortcuts": [  
    {  
      "name": "Open Play Later",  
      "short_name": "Play Later",  
      "description": "View the list of podcasts you saved for later",  
      "url": "/play-later?utm_source=homescreen",  
      "icons": [{ "src": "/icons/play-later.png", "sizes": "192x192" }]  
    },  
    {  
      "name": "View Subscriptions",  
      "short_name": "Subscriptions",  
      "description": "View the list of podcasts you listen to",  
      "url": "/subscriptions?utm_source=homescreen",  
      "icons": [{ "src": "/icons/subscriptions.png", "sizes": "192x192" }]  
    }  
  ]  
}
```

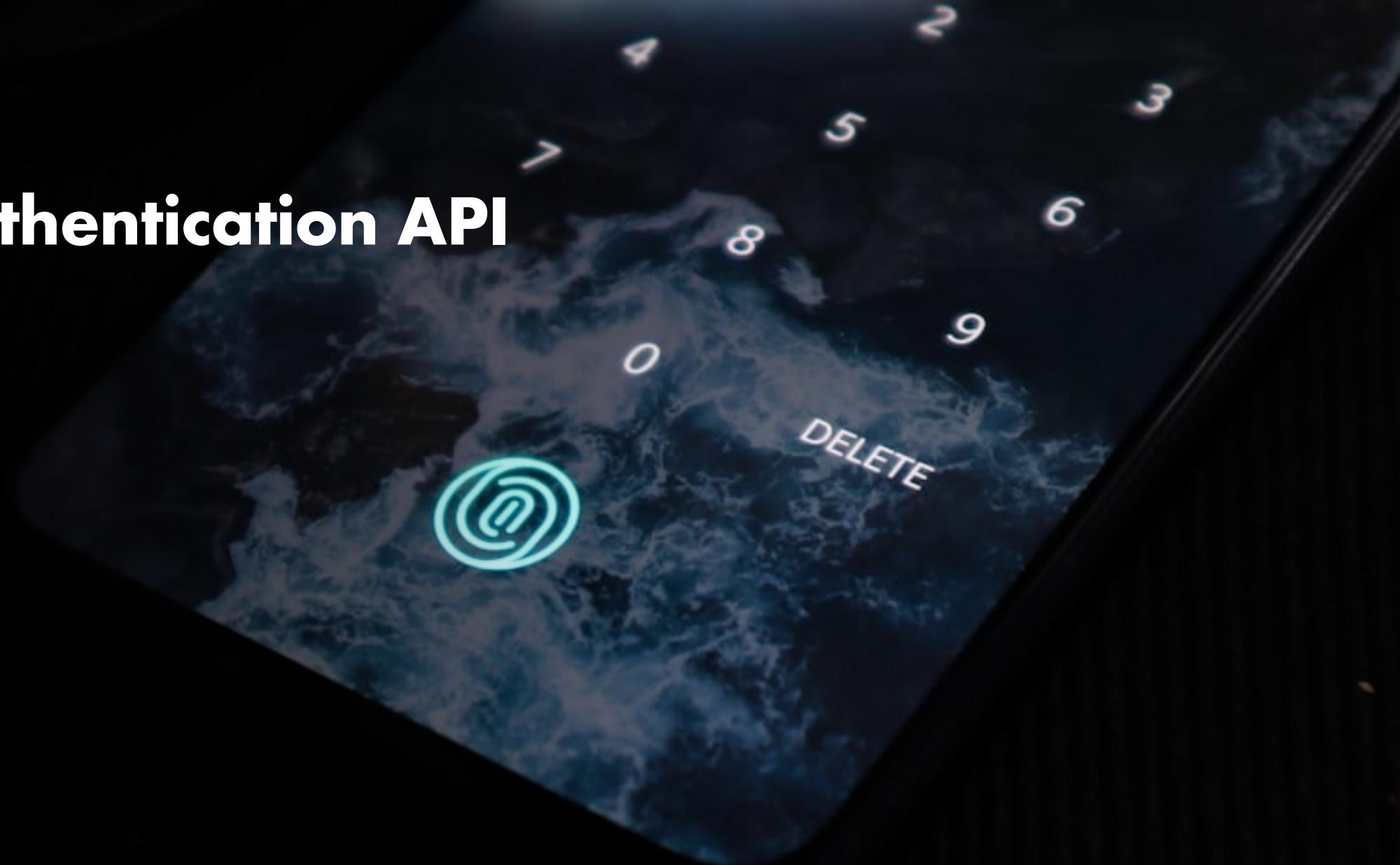


JS

Native Javascript



Web Authentication API



Web Authentication API

```
● ● ●  
let credential = await navigator.credentials.create({  
  publicKey: {  
    challenge: new Uint8Array([117, 61, 252, 231, 191, 241, ...]),  
    rp: { id: "acme.com", name: "ACME Corporation" },  
    user: {  
      id: new Uint8Array([79, 252, 83, 72, 214, 7, 89, 26]),  
      name: "jamiedoe",  
      displayName: "Jamie Doe"  
    },  
    pubKeyCredParams: [ {type: "public-key", alg: -7} ]  
  }  
});
```

Demo



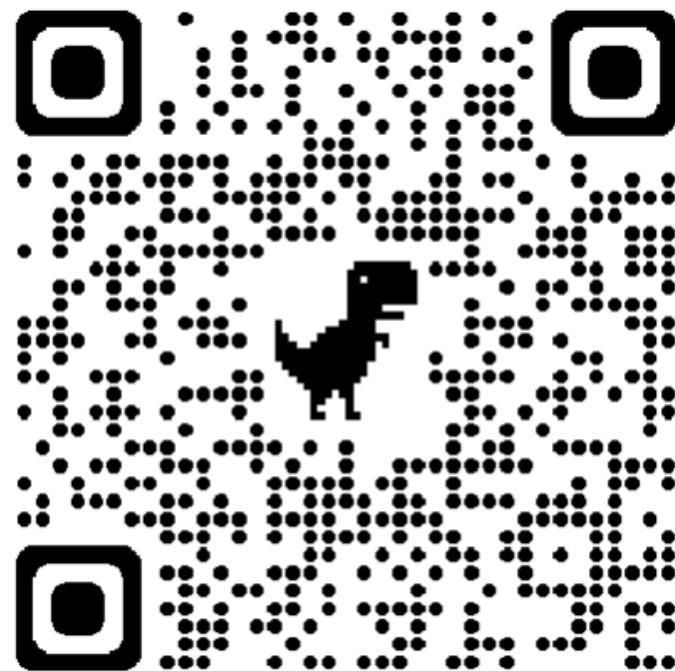
MediaStream Recording API

MediaStream Recording API

Demo

Resources

<https://yanai101.github.io/webApis.github.io/>



Additional resources

