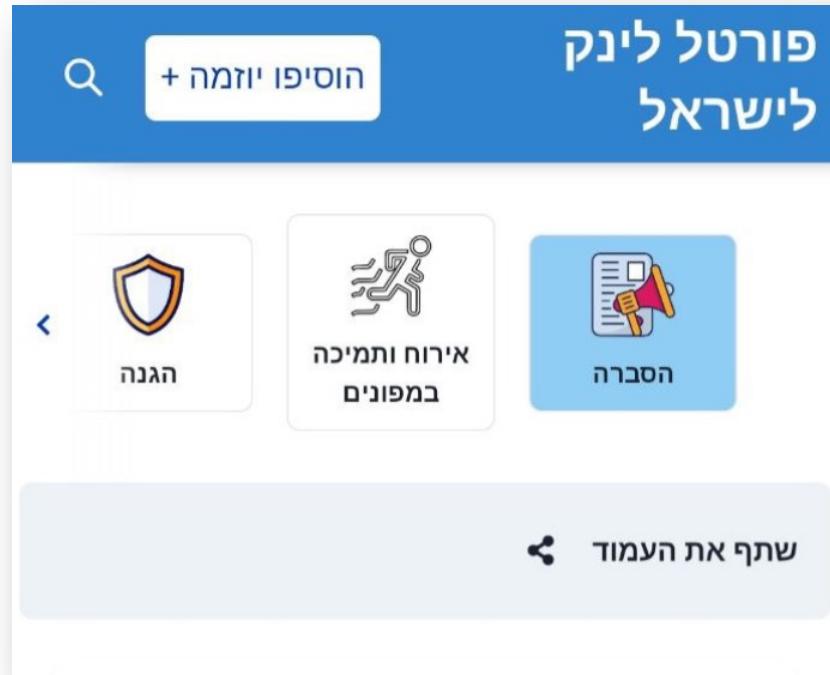


The background features a large, glowing circular portal with a white outline, centered in the frame. It is set against a dark, futuristic landscape composed of jagged, crystalline structures in shades of purple, blue, and red. The ground is a dark, reflective surface with a grid pattern. The overall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

# Unlocking JavaScript's Hidden Gems

# Once upon a time...



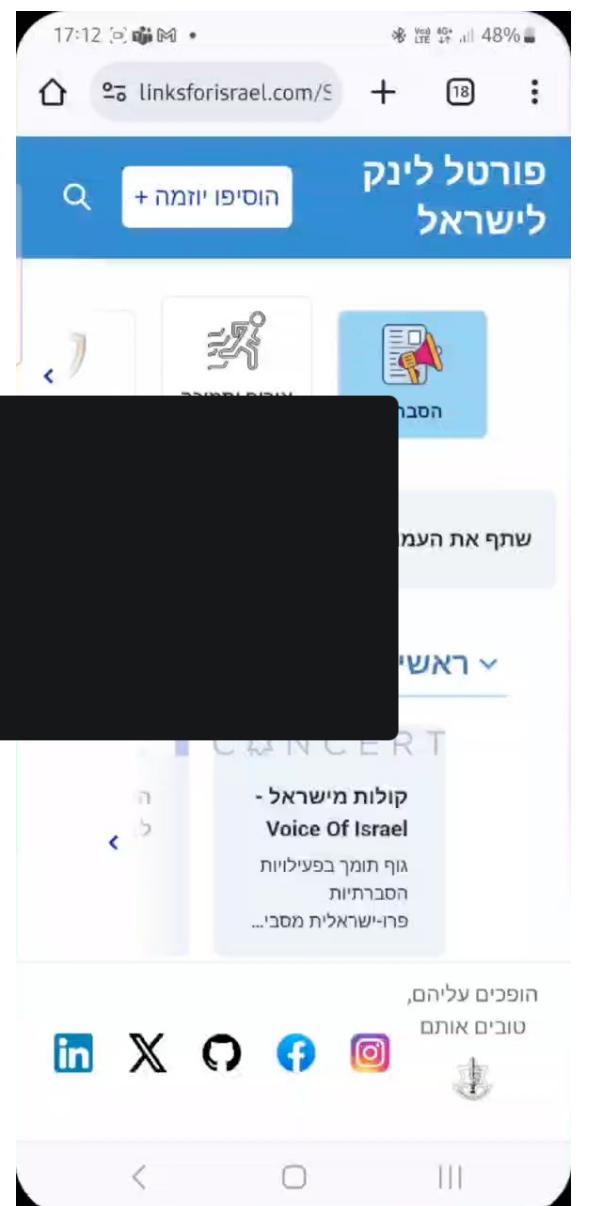
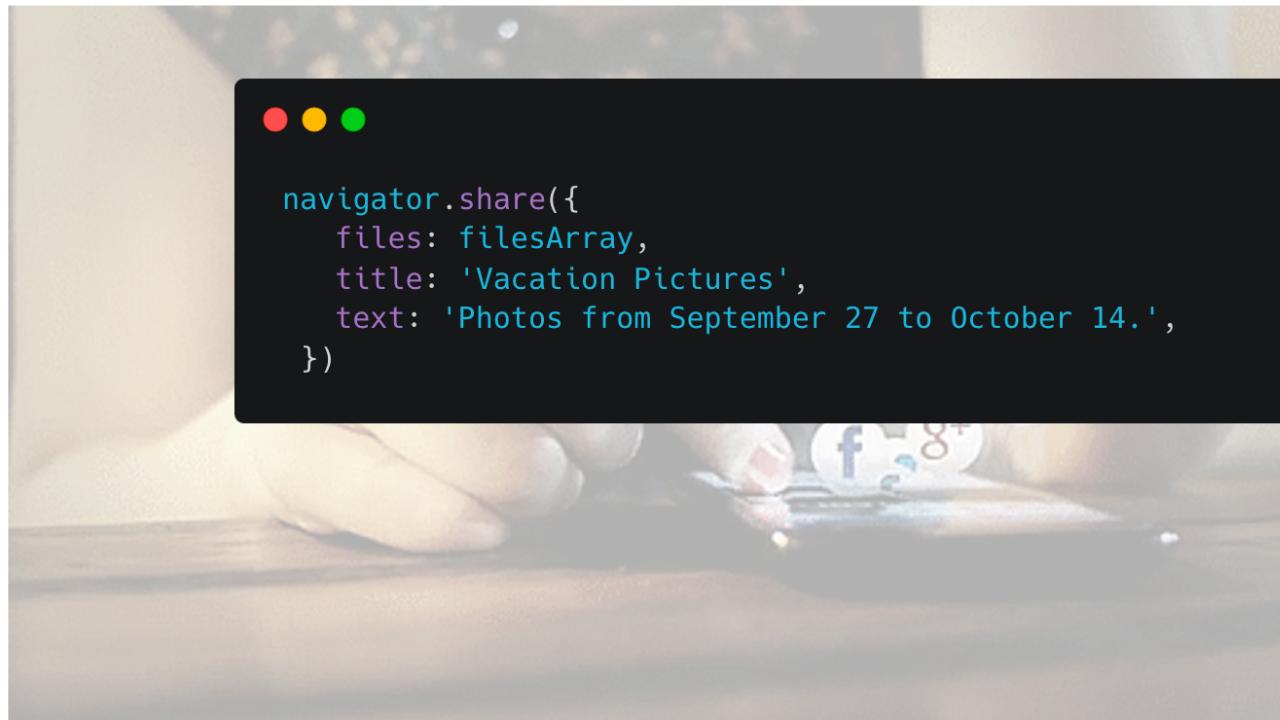
# Once upon a time...



## Web Share API

A screenshot of a mobile web browser interface. At the top, the URL 'linksforisrael.com/' is visible along with a search bar containing the text 'הויספו יוזמה +'. The main header reads 'פורטל-link לשראל' (Portal-link Israel). Below the header are three buttons: one with a gear icon labeled 'ארוחות ותמיינה במפות ניידים' (Meals and themes on mobile maps), another with a megaphone icon labeled 'הסברת' (Explanation), and a third button with a left arrow icon. A large, semi-transparent overlay box covers the middle portion of the screen. This box contains the word 'CONCERT' in large letters, followed by 'קולות מישראל - Voice Of Israel', 'גוף תומך בפעילויות הסברתיות', and 'פרחי-ישראלית מסכבי...'. At the bottom of the overlay, there is a row of social media icons for LinkedIn, X (Twitter), Quora, Facebook, and Instagram, followed by the Israeli flag.

# Once upon a time...



# About me

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

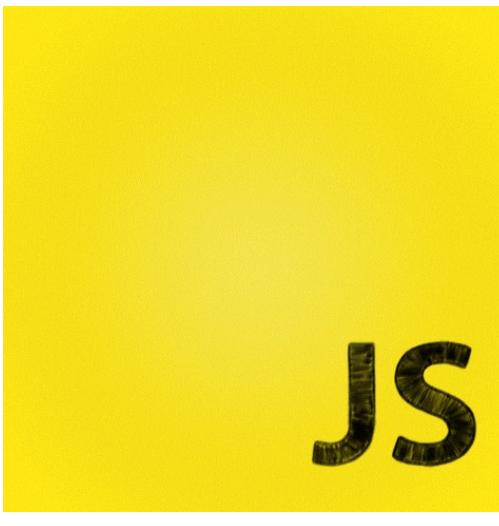


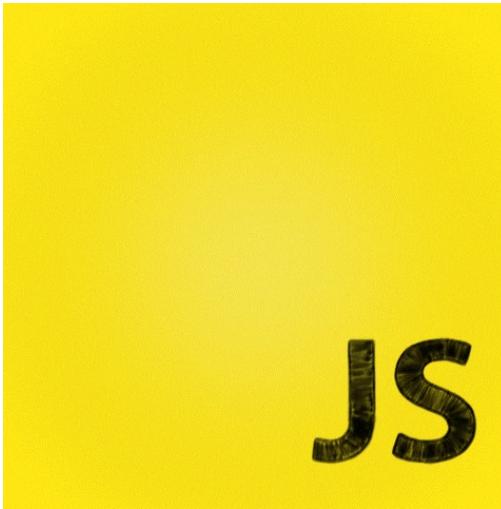
# **Use with cousin**



# Web APIs

# Web APIs





# Unfamiliar API in Web Development

# Let's help Arbel



# App Ideas

Share recipe



# Notifications API

# Notifications API



```
const notification = new Notification("Hello reversim 2024", {
  badge: "https://i.imgur.com/3424582.png",
  icon: "https://i.imgur.com/3424582.png",
  body: "Reversim 2024 - notification demo",
  // for mobile vs service worker
  vibrate: [100, 50, 100],
  actions: [
    {
      action: "open",
      title: "Open the site",
    },
    ...
  ],
});
```

[Demo](#)

# Notifications API

[Demo](#)

( : זה דוגמא לנוטיפיקציה

Notify me!

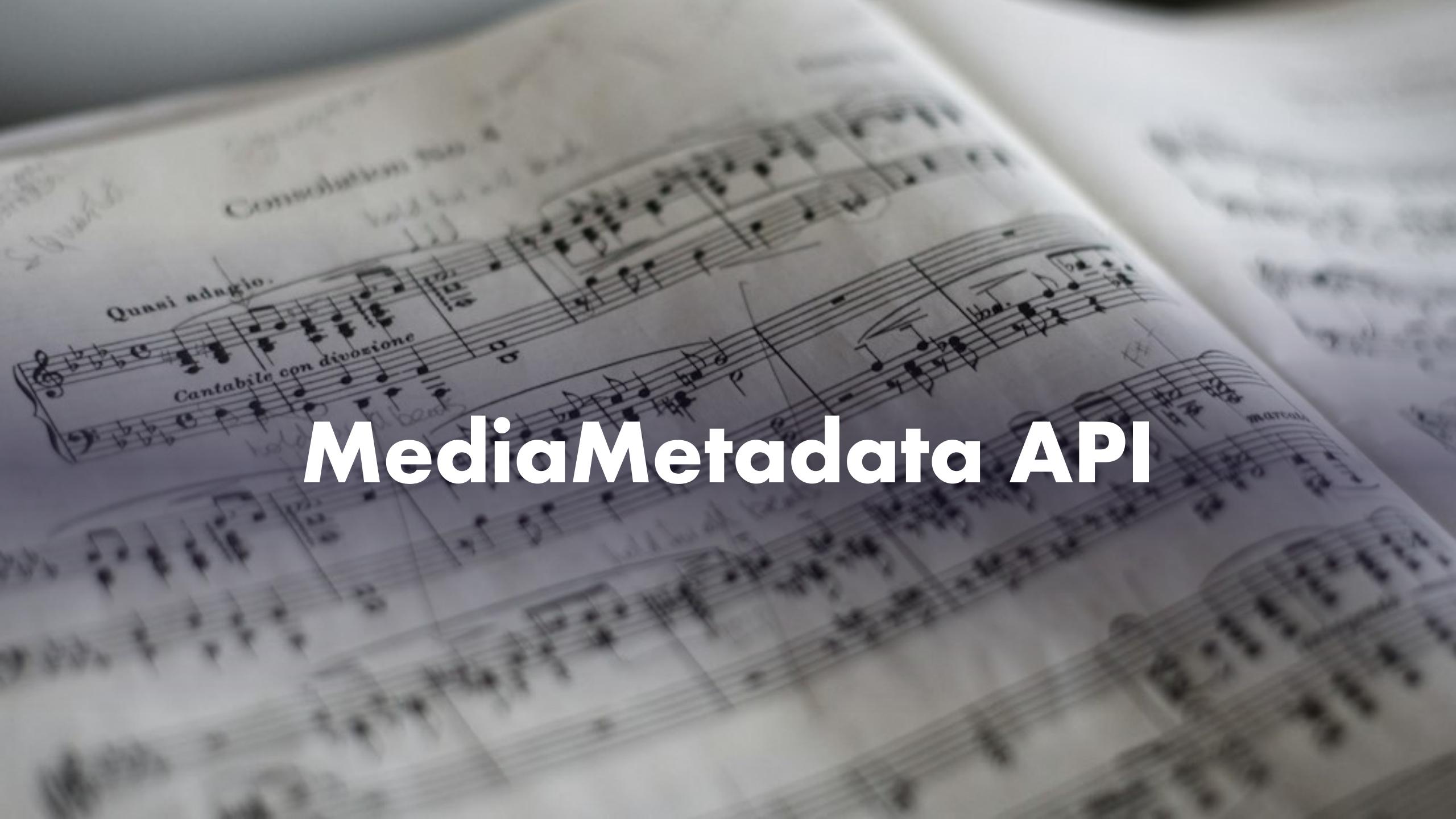
# App Ideas

Share recipe

Notification - Clock alert



# 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

Default levels ▾ 51 Issues: 2 45 4 8 verbose

38 messages

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

This screenshot shows a Chrome browser window with a YouTube video playing in the background. The video is of Rick Astley singing into a microphone. The browser's address bar shows the URL `youtube.com/watch?v=dQw4w9WgXcQ`. The title bar includes the standard Apple logo and menu items like Chrome, File, Edit, View, History, Bookmarks, Profiles, Tab, Window, and Help. The system tray at the top right shows various icons for battery, signal, and date/time (Sun 4 Feb 21:09). The YouTube interface has a dark theme with a search bar and a 'Challenging' button. Below the video player is a developer tools panel. The 'Console' tab is selected, showing a list of 38 messages. The messages include 30 warnings (indicated by an exclamation mark icon) and 2 errors (indicated by an X icon). The messages also mention 'No user messages', 'No errors', and 'No info'. The developer tools also show tabs for Elements, Sources, Network, Performance, Memory, Application, Security, and Lighthouse. A status bar at the bottom indicates 51 issues found across three levels: 2 critical, 45 warning, and 4 info.

# App Ideas

Share recipe

Notification - Clock alert

Video - Describe Cake



# Web Speech API



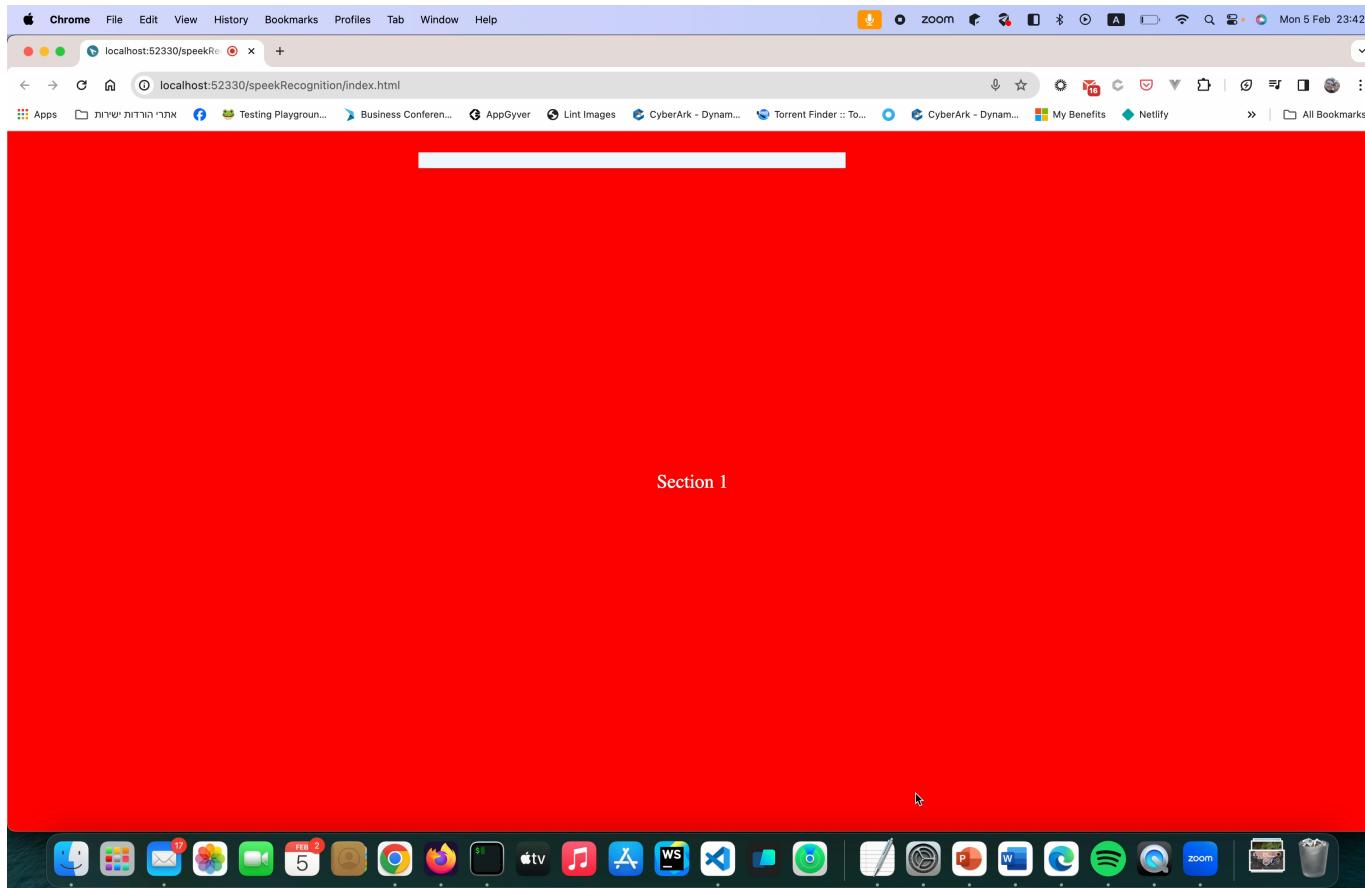
# Web Speech API

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);
```

# Web Speech API- SpeechSynthesis

```
...  
const utterance = new SpeechSynthesisUtterance();  
utterance.pitch = document.getElementById("pitch").value;  
utterance.rate = document.getElementById("rate").value;  
// Use the speech synthesis API to play the text  
speechSynthesis.speak(utterance);  
...
```

[Demo](#)

# App Ideas

Share recipe

Notification - Clock alert

Video - Describe Cake

Voice navigation



# 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**



# App Ideas

Share recipe

Notification - Clock alert

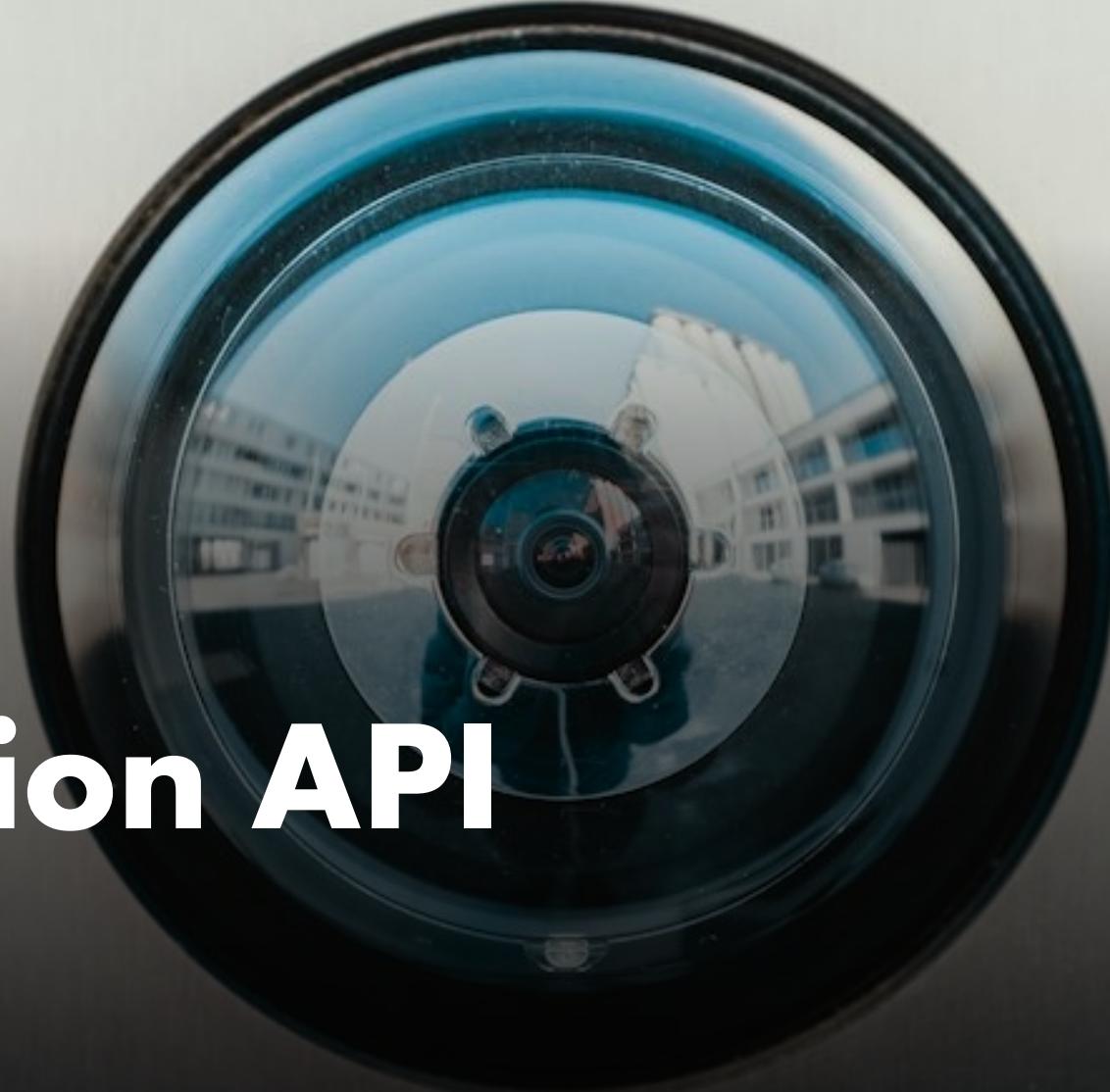
Video - Describe Cake

Voice navigation

Timer Window



# Detection API



# Shape & Face Detection API

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

For Stream Or Static Image

[Demo](#)



# Shape & Face Detection API

## Shape Detection API

Face detection  Barcode detection  Text detection      Analyze image file:

This is the shape detection API implementation demo

The following demo allows you to detect shapes, barcodes and faces from an uploaded image.

Refer to the [How to detect human faces \(and other shapes\) in JavaScript](#) on the CKEditor Blog to find out more about the presented function.

Please bear in mind, this mechanism only works in Chrome (Mac, Windows) and it requires this flag: `chrome://flags#enable-experimental-web-platform-features` to be turned on.

# 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"  
]
```

# App Ideas

Share recipe

Notification - Clock alert

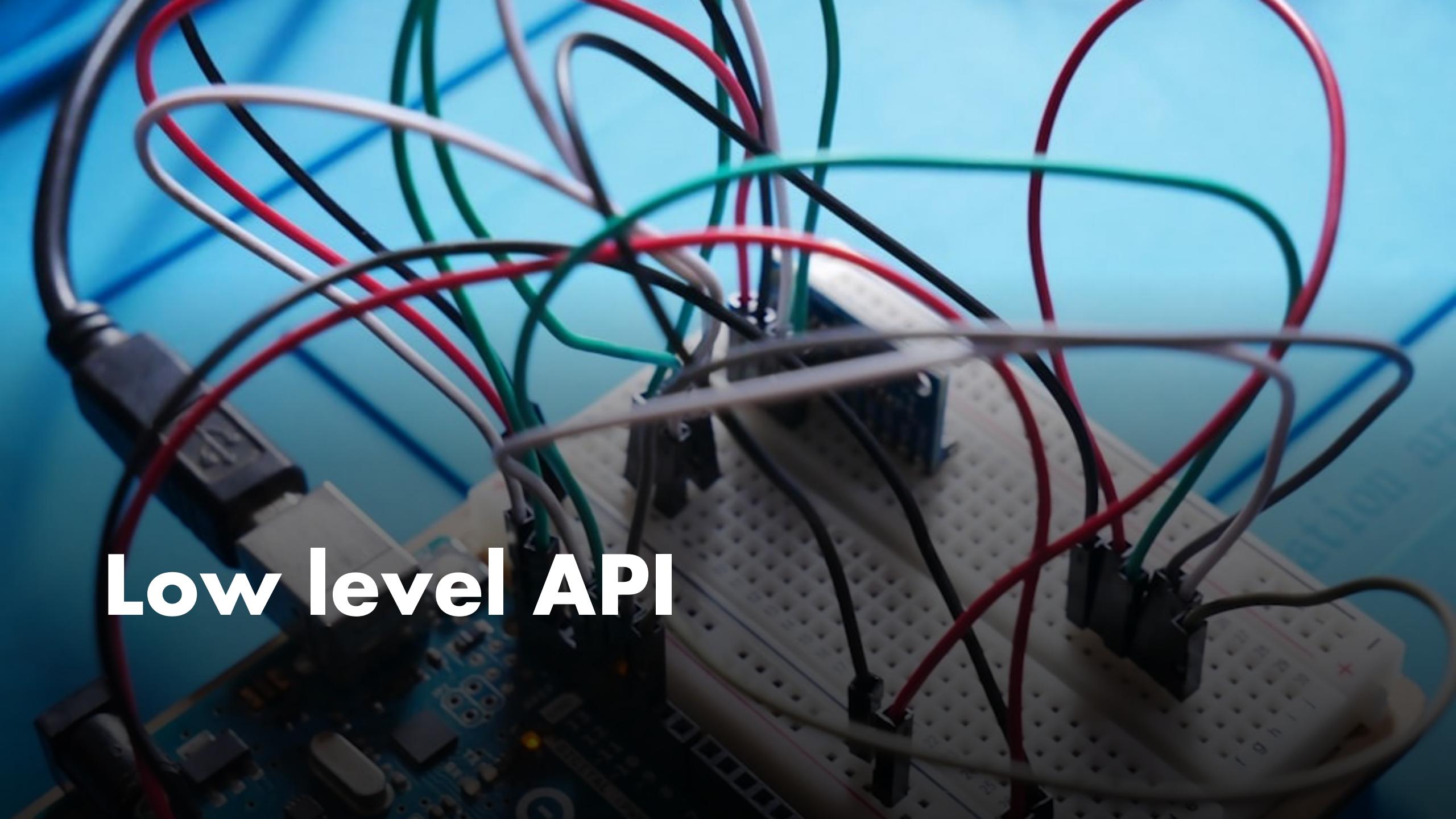
Video - Describe Cake

Voice navigation

Timer Window

Recipes scanner

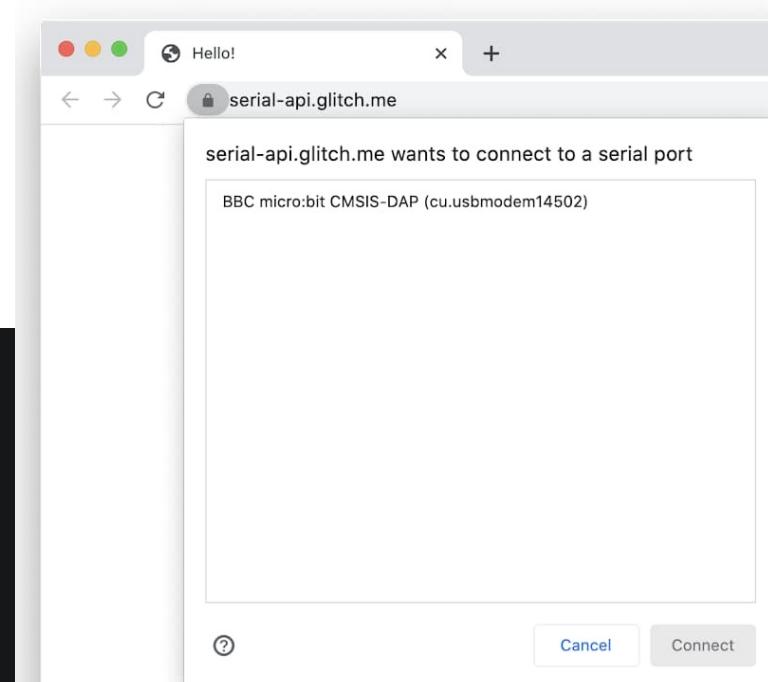




# Low level API

# 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();
```



# WebHID API

Low level API



<https://thebitlink.github.io/WebHID-DS4/>

# 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  
}
```

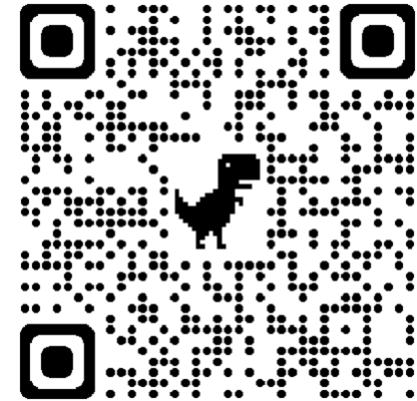


# Gamepad API



# 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,
  );
});
```



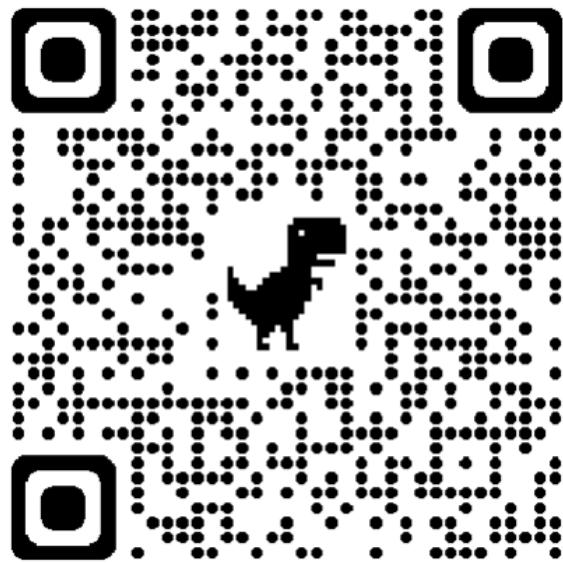
[Demo](#)

[Demo](#)

# WakeLock API



# WakeLock API



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

# App Ideas

Share recipe

Notification - Clock alert

Video - Describe Cake

Voice navigation

Timer Window

Lock screen on





# WebGPU

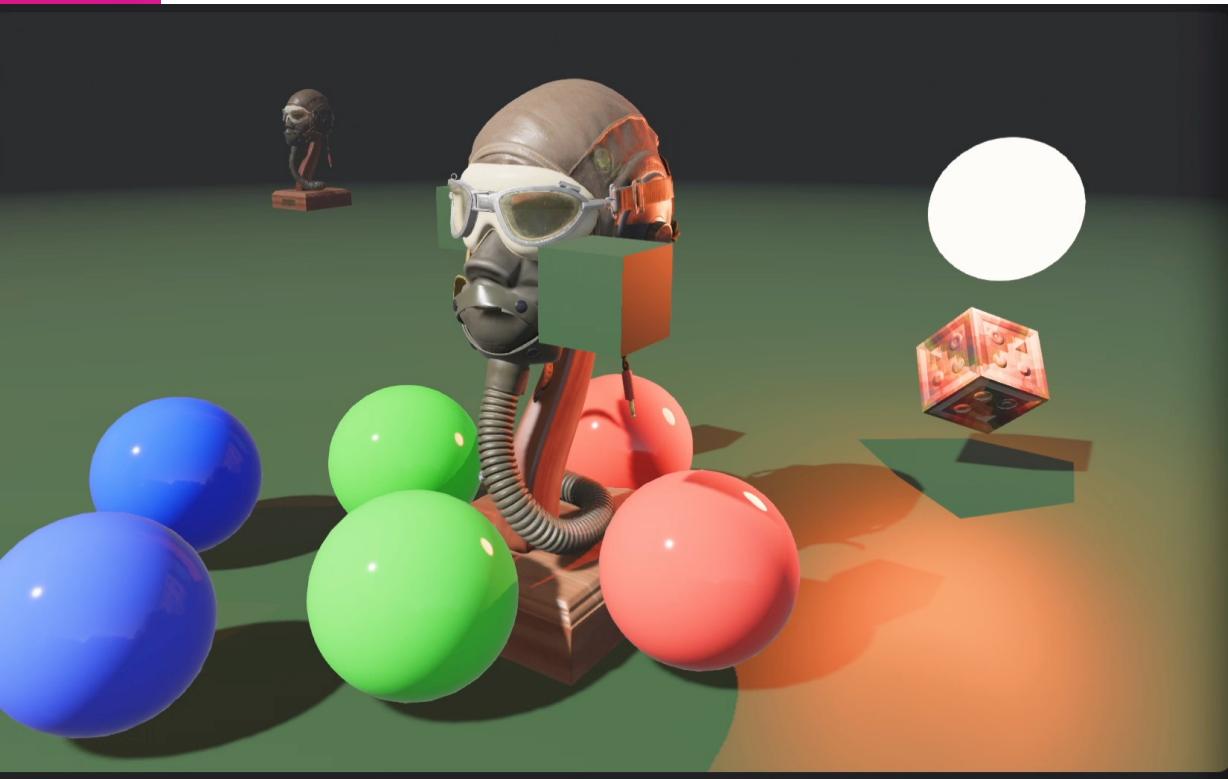
# WebGPU



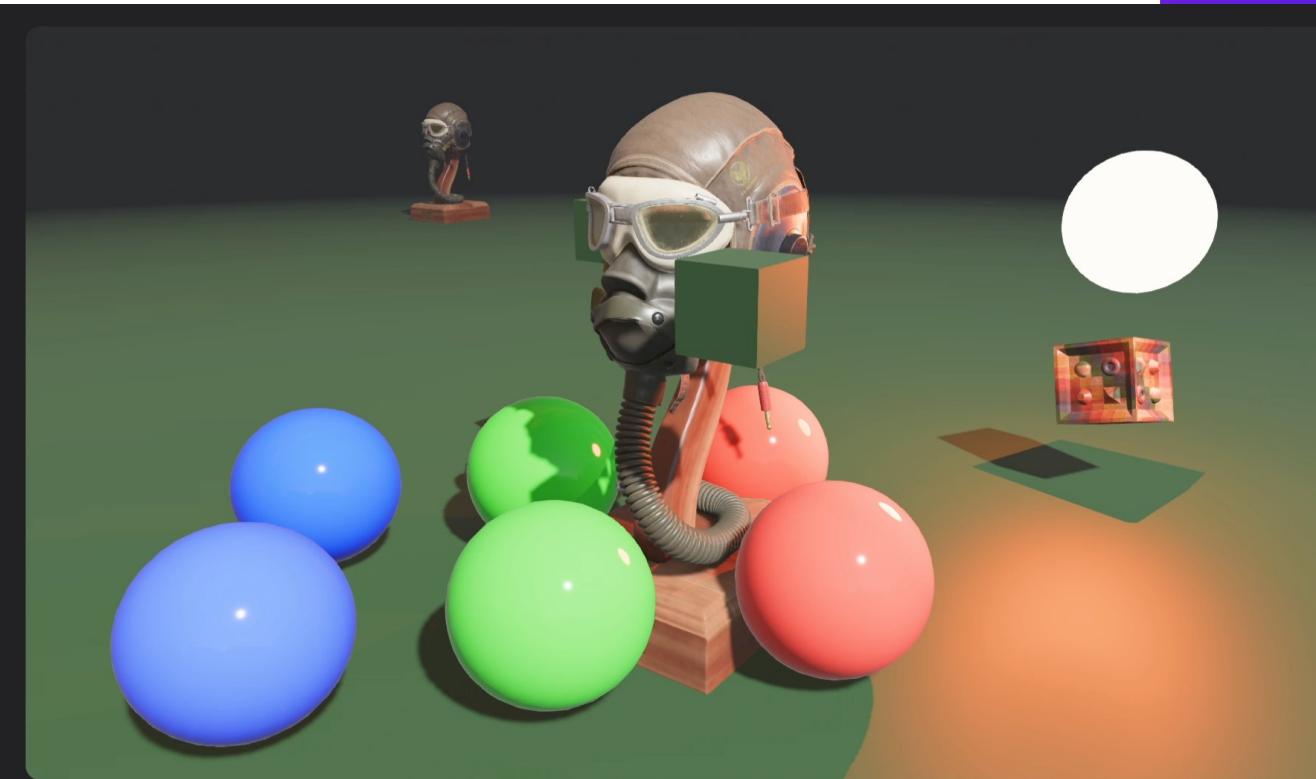
# WebGPU



# WebGl2



# WebGPU



A man with dark hair and a beard, wearing a blue zip-up hoodie, is wearing a white VR headset. He is looking towards the right side of the frame, where the word "WebXR" is displayed in large, white, sans-serif letters. His right hand is raised, pointing towards the text. The background is blurred, showing what appears to be an indoor setting with warm lighting.

WebXR



# WebXR Augmented Reality



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



# WebXR

## <model-viewer>

### <model-viewer>

[DOCUMENTATION](#) | [EXAMPLES](#)

#### Loading

#### Augmented Reality

[WebXR](#)

[AR](#)

[Scene Viewer](#)

[Place on a Wall](#)

[Custom Button](#)

[Transparent Background](#)

#### Staging & Cameras

#### Annotations

#### Lighting & Skybox

#### Animation

#### Materials & Scene

#### Post Process Effects

### WebXR Demo

Customize a WebXR Augmented Reality session with HTML, CSS, and JS in Chrome 83+ on Android.

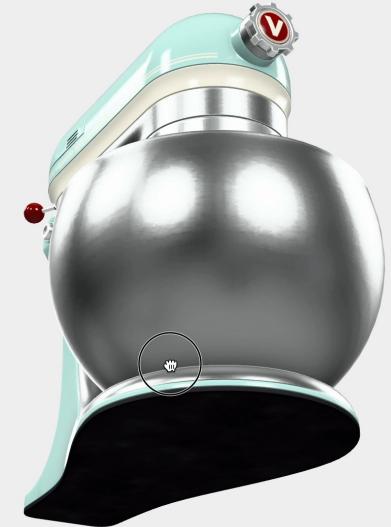
This demonstrates the use of slots, as well as shared and unshared DOM between the 3D and AR modes.

A slot is used here for replacing the default AR button with a custom one - in this case the one recommended by Sceneviewer. This is not the default only because this way localization of the text is left to whatever system you prefer to use. Note the AR button will only be visible on AR-capable devices.

By styling based on the ar-status attribute, you can add DOM that only shows up in certain modes. In this case a CSS animation has been added to prompt the user to move their phone around to help ARCore find their floor so that the object can be placed. User studies show a prompt like this is an important part of guiding users to a good AR experience.

Finally, even complex DOM can easily function in both 3D and AR modes, including interactions via script. In this case a simple carousel of models is demonstrated. Unfortunately none of this DOM content can flow into SceneViewer or QuickLook, as these are native apps. Only through WebXR, now our default AR mode, can this be achieved, as the AR session is still inside of the browser. This also removes the need to redownload the model.

Note that by not specifying an ios-src, the USDZ will instead be auto-generated when the user clicks the AR button on iOS to launch Quick Look.





# WebXR

## <model-viewer>

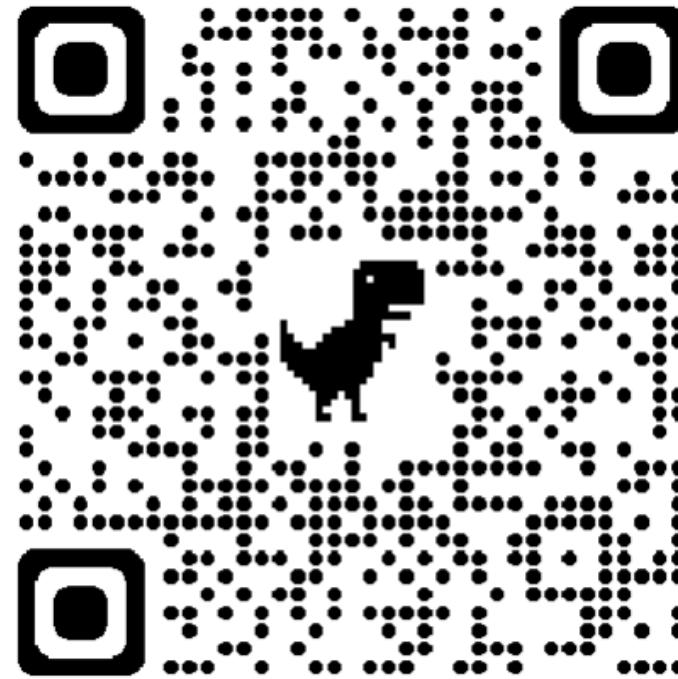
Web demo:

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

Mobile Demo:

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

# <model-viewer>



Please put your  
phone down.



# App Ideas

Share recipe

Notification - Clock alert

Video - Describe Cake

Voice navigation

Timer Window

Lock screen on

Cake Model



Please put your  
phone down.



פה סרטון של האובייקט תلت מימד...  
להקליט באותו יום

# 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



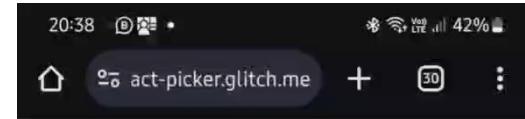
# Web Audio & Web MID

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.

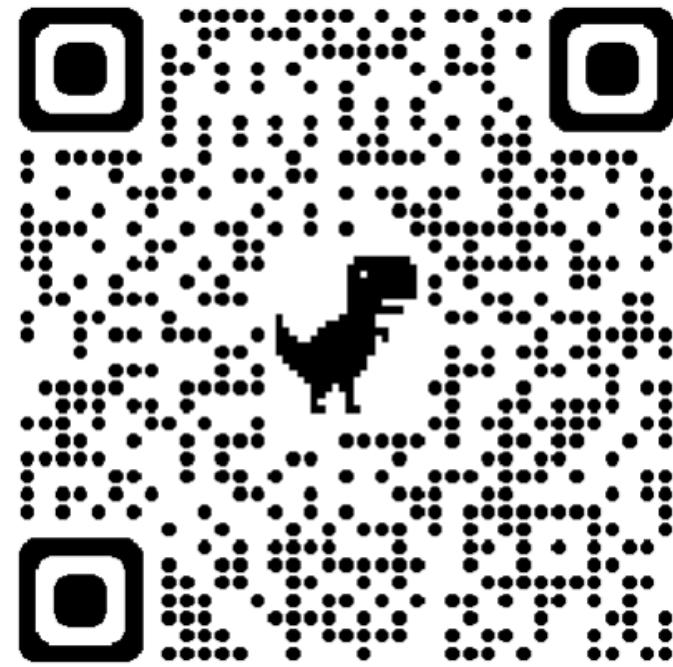
- Multiple results?
- Include name?
- Include email addresses?
- Include telephone numbers?
- Include addresses? (Chrome 84 and later)
- Include icons? (Chrome 84 and later)

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](#)



# Contact Picker API



Try me!

Please put your  
phone down.



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**

**Check it out... ☺**



Please put your  
phone down.



# EyeDropper API

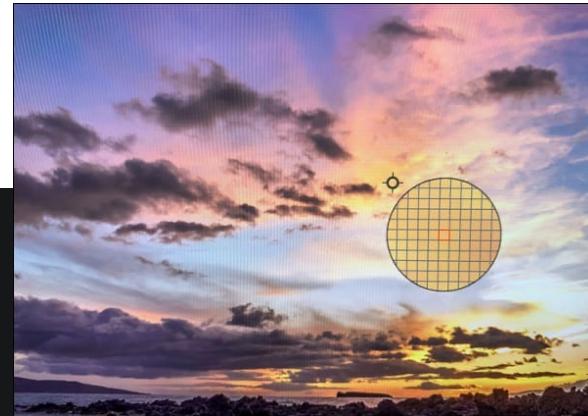


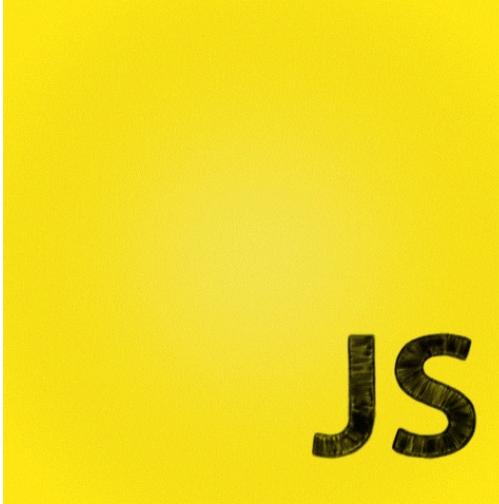
# EyeDropper API

```
const eyeDropper = new EyeDropper();

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

```





**JS**



# Mobile and Progressive Web Apps

# WebO(ne)T(ime)Password API



# WebOTP API



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

```

[Demo](#)

14:03

Web OTP API Demo

Enter your phone number

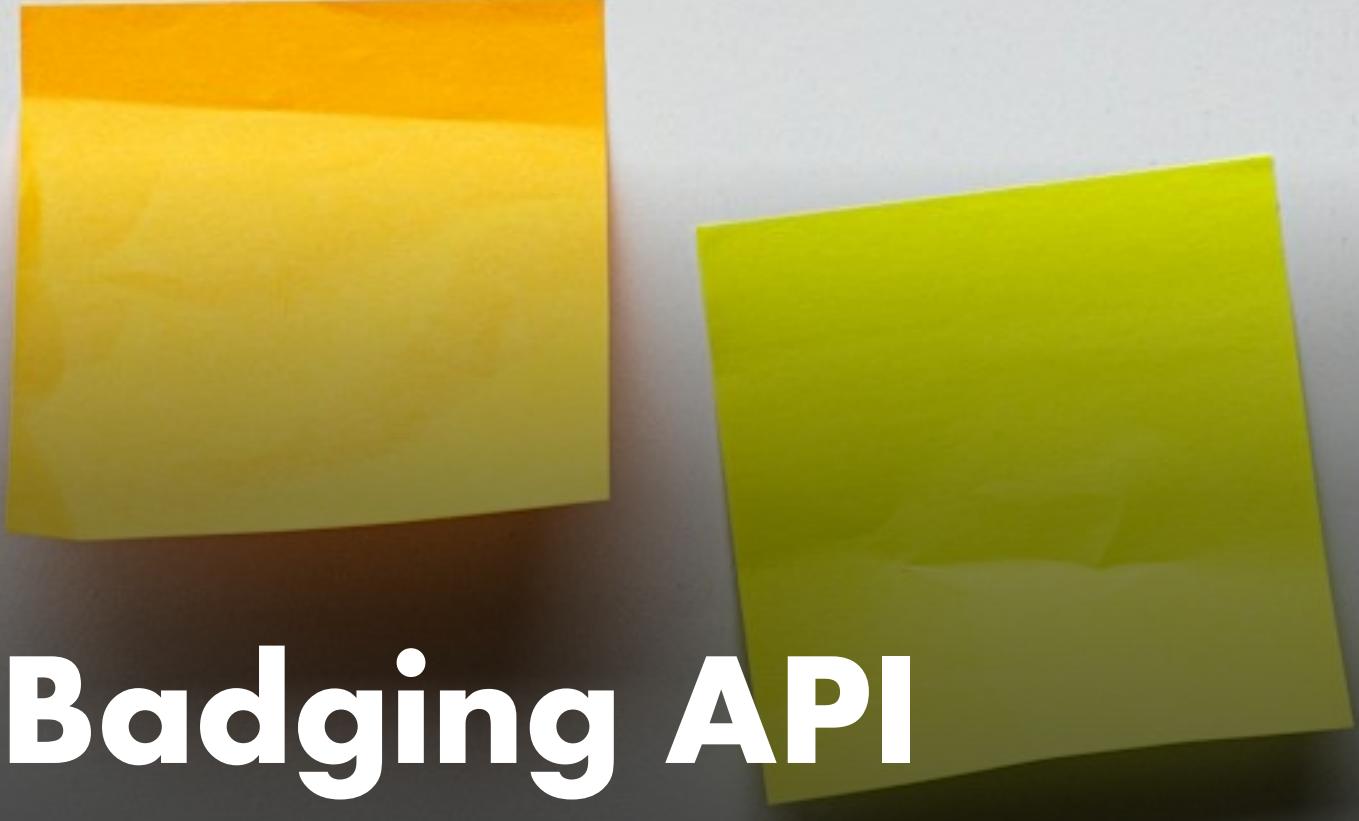
Phone Number \*

+ \* include the country code

VERIFY

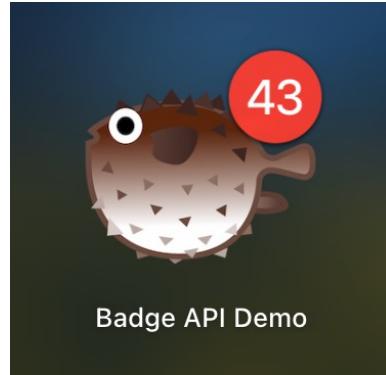
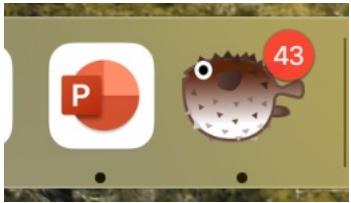
In order to prevent abuse, **please do not share this page's URL in the public place.**

 プライバシー・利用規約

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



# App Ideas

Share recipe

Notification - Clock alert

Video - Describe Cake

Voice navigation

Timer Window

Lock screen on

Cake Model

New recipes



# 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');
});
```



# Web Bluetooth



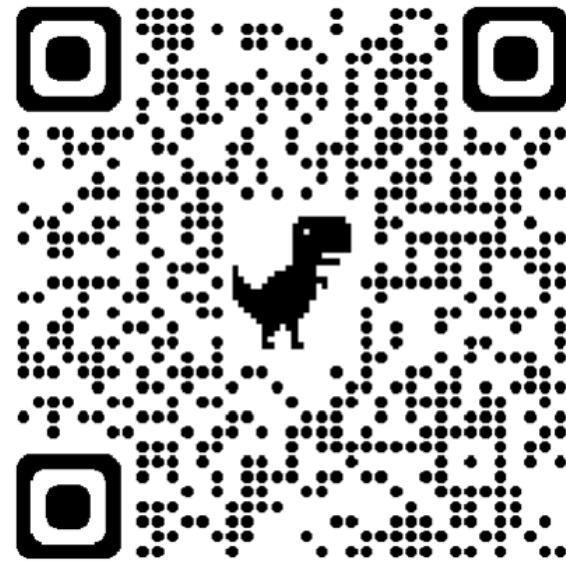
<https://googlechrome.github.io/samples/web-bluetooth/device-info.html>

# Web NFC



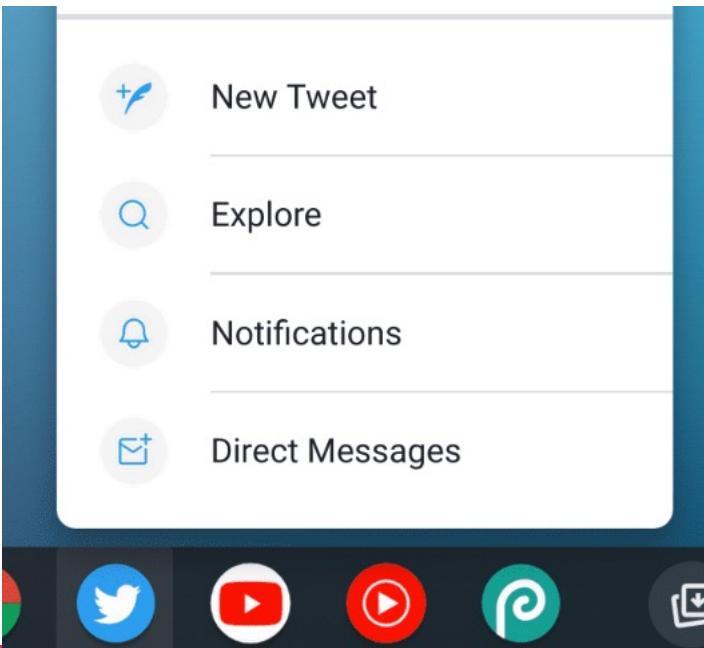
<https://labs-web-nfc.glitch.me/>

# Multi-touch API

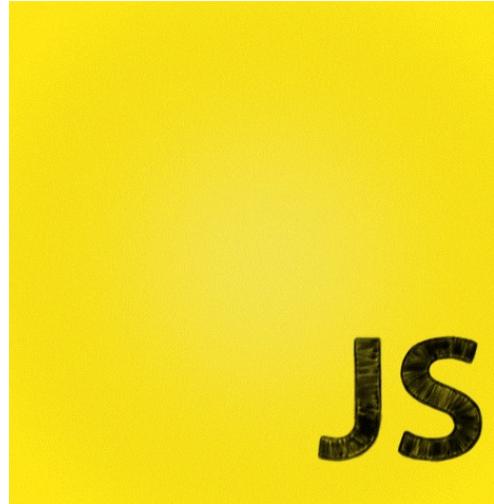


<https://codepen.io/Suniaster/pen/qBmEzOW>

# 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" }]  
    }  
  ]  
}
```

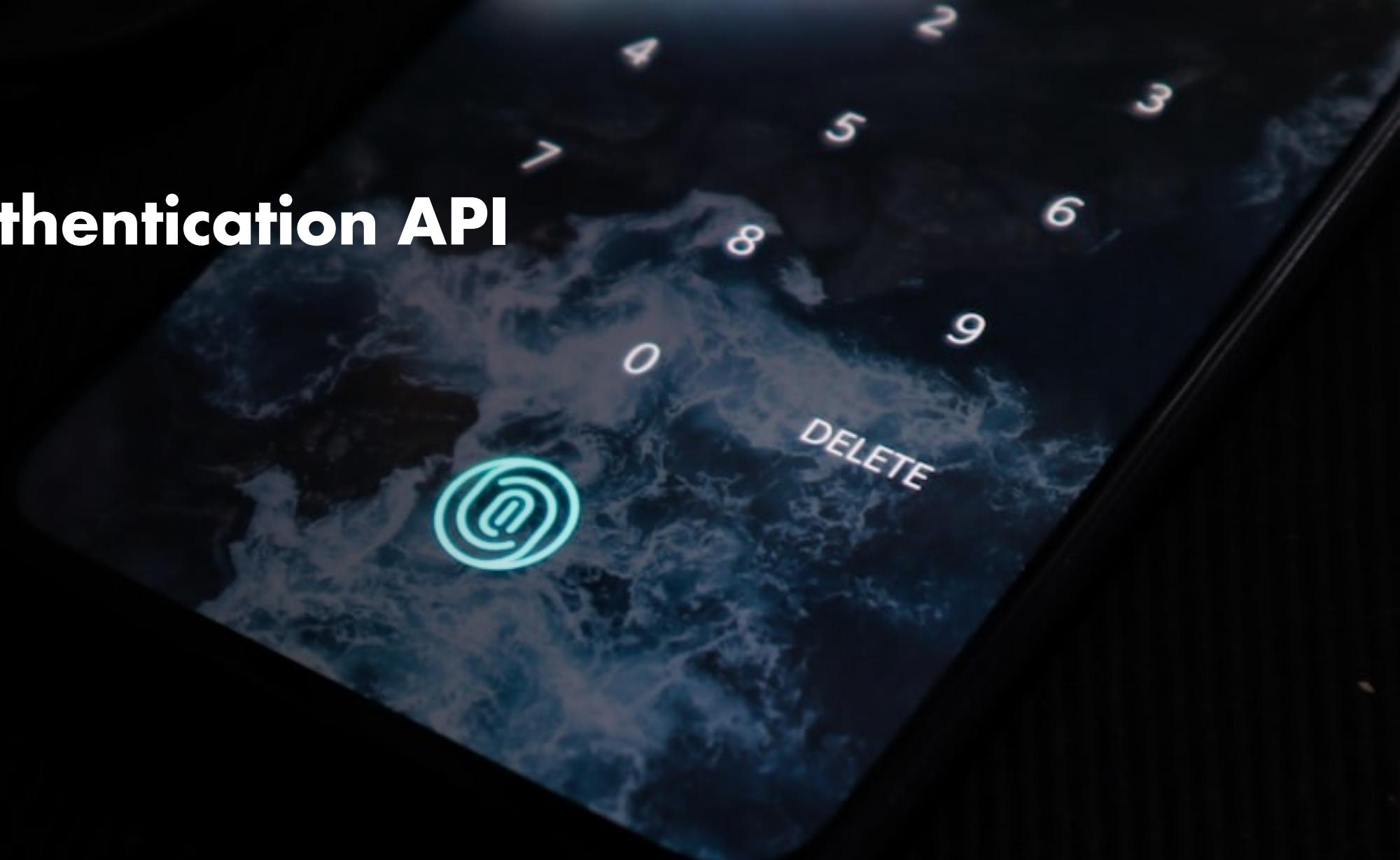


JS

**Native Javascript**



# Web Authentication API



A collection of numerous metal keys of different shapes, sizes, and finishes (silver, gold, brass) are piled together against a solid black background. The keys are oriented in various directions, creating a dense, overlapping texture.

**Please stop using password**

# 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

# App Ideas

Share recipe  
Notification - Clock alert  
Video - Describe Cake  
Voice navigation  
Timer Window  
Lock screen on  
Cake Model  
New recipes  
No password login





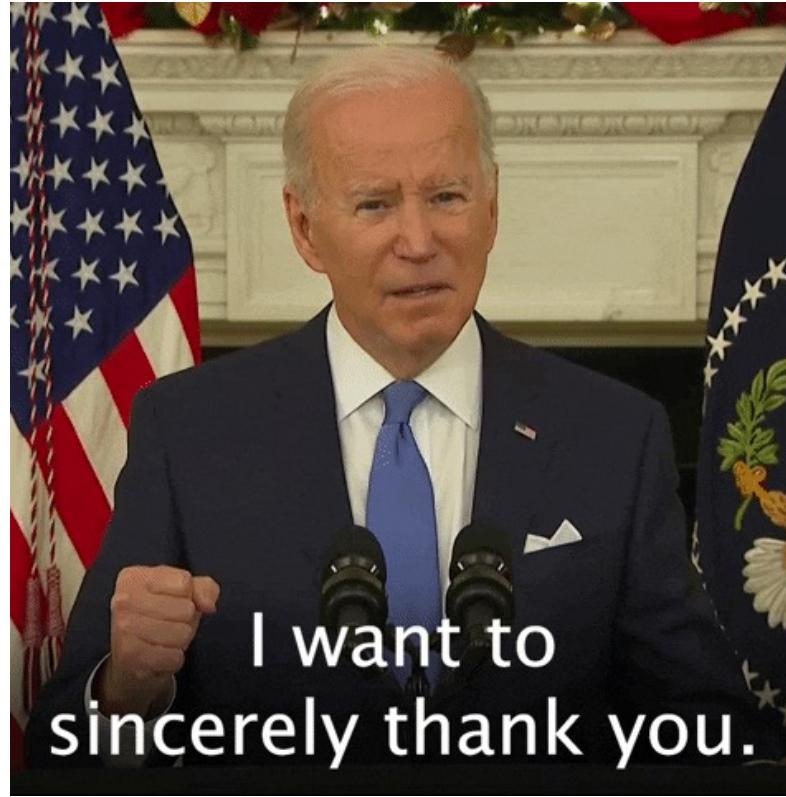
# MediaStream Recording API

# **MediaStream Recording API**

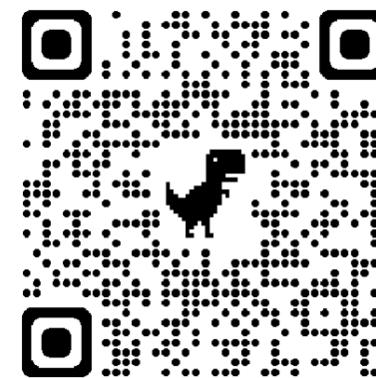
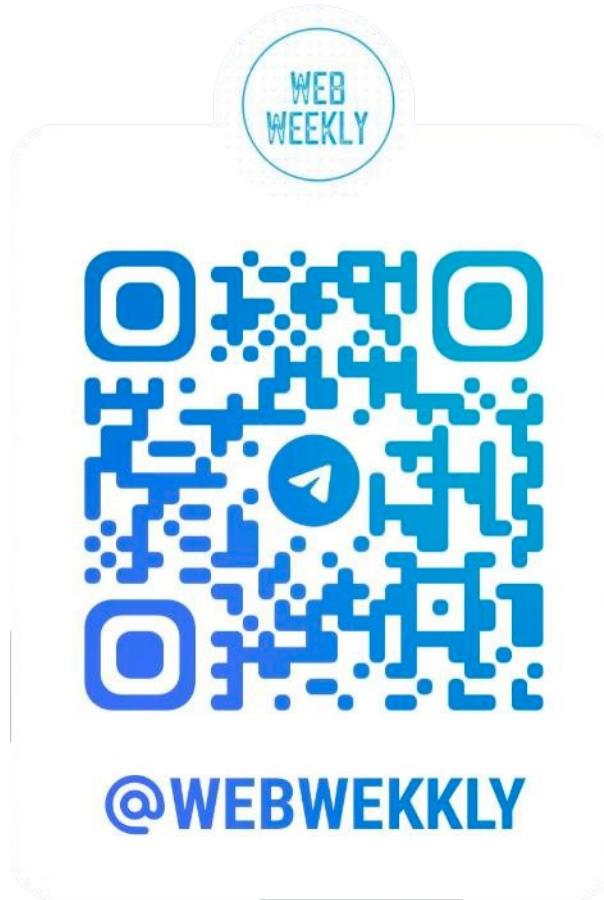
Demo

**תוצאות הסקר...**

# Thanks !!!



I want to  
sincerely thank you.



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