WEB APIs YOU NEED TO KNOW!

★ *Super Helpful..*

Network Information API

Vibrate API

Bluetooth API

Clipboard Async API

Fullscreen API

Battery Status API

Online Offline API

Geolocation API

Learn all in one place →

📺 Fullscreen API

The **Fullscreen API** provides methods to present a specific Element in a **full-screen mode.**

```html
<button onclick="activateFullscreen(document.documentElement);">
  Go fullscreen!
</button>

<button onclick="deactivateFullscreen();">
  Leave fullscreen
</button>
```
HTML

```javascript
function activateFullscreen(element) {
  if(element.requestFullscreen) {
    element.requestFullscreen();
  }
};

function deactivateFullscreen() {
  if(document.exitFullscreen) {
    document.exitFullscreen();
  }
};
```
JavaScript

Network Information API

The Network Information API provides information about the network types *(e.g., 'wifi', 'cellular', etc.)*

Observing user's connection

```
navigator
    .connection
    .effectiveType;
```

Returns the *effective type* of the connection which could be "4g", "3g", "2g", or "slow-2g".

```
navigator
    .connection
    .downlink;
```

Returns the *effective bandwidth* estimated in megabits per second.

@mgechev

JavaScript

```javascript
console.log(navigator.connection);
```

- Network Type: 3g
- Round Trip Time(rtt): 300
- Bandwidth estimate(in MBPS): 1.45
- Max Bandwidth estimate(in MBPS): Infinity
- Save data enabled: false
- Device Connection Type: unknown

Vibrate API

Use this to make your website cool

```
window.navigator.vibrate(300); // vibrate for 300ms
```

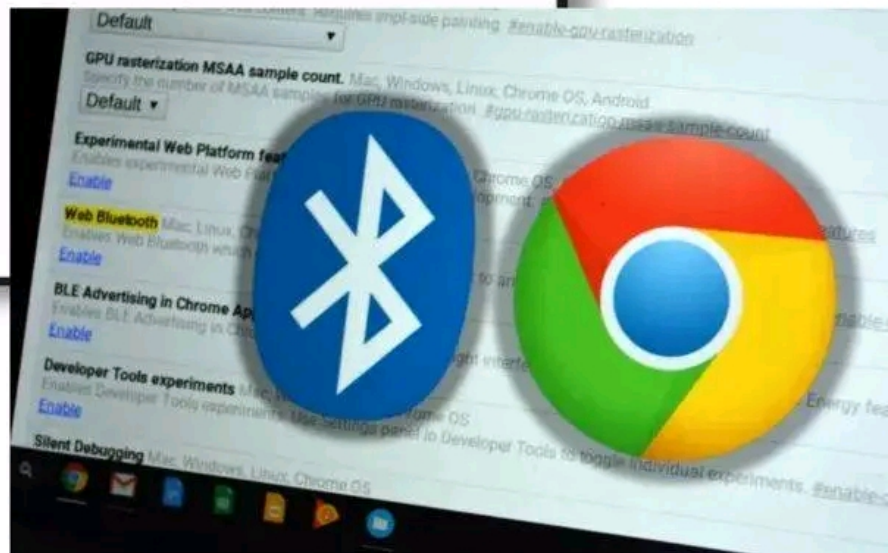The **Navigator.vibrate()** method pulses the vibration hardware on the device, if such hardware exists.

Bluetooth API

This **web API** allows you to connect to the **bluetooth** devices.

```javascript
navigator.bluetooth.requestDevice({
    acceptAllDevices: true
}).then(device => {
    setDeviceName(device.name);
    setDeviceId(device.id)
    setDeviceConnected(device.connected);
}).catch(err => {
    console.log(err);
    setError(true);
})
```

JavaScript

Online Offline API

Checks if the user is online or offline

```javascript
if (navigator.onLine) {
  console.log('online');
} else {
  console.log('offline');
}

//To check if you are
online or offline
```
JavaScript

If the browser doesn't support **navigator.onLine**
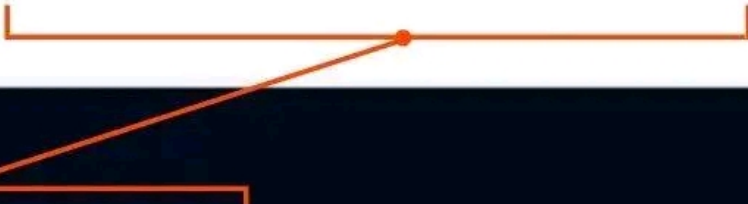the above example will always come out as
false/undefined.

📍 Geolocation API

Use this to get the users location

```
navigator.geolocation.getCurrentPosition(callbackfunction)
```

Checks if geolocation is available

JavaScript

```javascript
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(function(position){
            console.log(position);
    });
}
```

The **geolocation API** allows javascript or web content to access the user's location or device's location.

@developers_community_._   X   @gowsami.dev