



JavaScript

Where the fun starts! *

Mozilla is cool



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>My test page</title>
  <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" type="text/css">
  <link href="styles/style.css" rel="stylesheet" type="text/css">
</head>
<body>
  <h1 id="heading">Mozilla is cool</h1>
  

  <p>At Mozilla, we're a global community of</p>

  <ul> <!-- changed to list in the tutorial -->
    <li>technologists</li>
    <li>thinkers</li>
    <li>builders</li>
  </ul>

  <p>working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.</p>

  <p>Read the <a href="https://www.mozilla.org/en-US/about/manifesto/">Mozilla Manifesto</a> to learn even more about the values and principles that guide the pursuit of our mission.</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>My test page</title>
  <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" type="text/css">
  <link href="styles/style.css" rel="stylesheet" type="text/css">
</head>
<body>
  <h1 id="heading">Mozilla is cool</h1>
  

  <p>At Mozilla, we're a global community of</p>
  https://codeload.github.com/mdn/beginner-html-site-styled/zip/refs/heads/gh-pages
  <ul> <!-- changed to list in the tutorial -->
    <li>technologists</li>
    <li>thinkers</li>
    <li>builders</li>
  </ul>

  <p>working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.</p>

  <p>Read the <a href="https://www.mozilla.org/en-US/about/manifesto/">Mozilla Manifesto</a> to learn even more about the values and principles that guide the pursuit of our mission.</p>
</body>
</html>

<!-- https://codeload.github.com/mdn/beginner-html-site-styled/zip/refs/heads/gh-pages -->
```

Mozilla is cool



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

JavaScript is cooler



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

Elements Console Sources Network >

top ▾ Filter Default lev

No Issues

```
> const heading = document.querySelector("h1")
< undefined
> heading
<   <h1>Mozilla is cool</h1>
> heading.innerHTML = "JavaScript is cooler"
< 'JavaScript is cooler'
>
```

```
document.querySelector("h1").innerHTML = "JavaScript is Cooler";
```



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

Elements Console Sources Network Performance >>

top Filter Default levels

```
> document.querySelector("h1").innerHTML = "JavaScript is Cooler"
<- 'JavaScript is Cooler'
> |
```

```
document.querySelector("h1").innerHTML = "JavaScript is Cooler";
```

mdn web docs References Guides Plus Blog Play AI Help BETA Theme Log in Sign up for free

References > Web APIs > Document > querySelector()

Filter

open()
prepend()
queryCommandEnabled() ▲ ↗
queryCommandState() ▲ ↗
queryCommandSupported() ▲ ↗
querySelector()
querySelectorAll()
releaseCapture() ▲
replaceChildren()
requestStorageAccess()
requestStorageAccessFor() ▲
startViewTransition() ▲
write()
writeln()
Events
afterscriptexecute ▲
beforescriptexecute ▲
copy

Document: querySelector() method

The [Document](#) method `querySelector()` returns the first [Element](#) within the document that matches the specified selector, or group of selectors. If no matches are found, `null` is returned.

Note: The matching is done using depth-first pre-order traversal of the document's nodes starting with the first element in the document's markup and iterating through sequential nodes by order of the number of child nodes.

Syntax

```
JS
querySelector(selectors)
```

Parameters

selectors
A string containing one or more selectors to match. This string must be a valid CSS selector string; if it isn't, a `SyntaxError` exception is thrown. See [Locating DOM elements using selectors](#) for more about selectors and how to manage them.

```
document.querySelector("h1").innerHTML = "JavaScript is Cooler";
```

mdn web docs References Guides Plus Blog Play AI Help BETA Theme Log in Sign up for free

References > Web APIs > Document > querySelector()English (US)

Filter

[open\(\)](#)
[prepend\(\)](#)
[queryCommandEnabled\(\) ▲ ↗](#)
[queryCommandState\(\) ▲ ↗](#)
[queryCommandSupported\(\) ▲ ↗](#)
[querySelector\(\)](#)
[querySelectorAll\(\)](#)
[releaseCapture\(\) ▲](#)
[replaceChildren\(\)](#)
[requestStorageAccess\(\)](#)
[requestStorageAccessFor\(\) ▲](#)
[startViewTransition\(\) ▲](#)
[write\(\)](#)
[writeln\(\)](#)
▼ Events

Return value

An [Element](#) object representing the first element in the document that matches the specified set of [CSS selectors](#), or `null` is returned if there are no matches.

If you need a list of all elements matching the specified selectors, you should use [querySelectorAll\(\)](#) instead.

Exceptions

`SyntaxError DOMException`

Thrown if the syntax of the specified *selectors* is invalid.

Usage notes

If the specified selector matches an ID that is incorrectly used more than once in the document, the first element with that ID is returned.

[CSS pseudo-elements](#) will never return any elements, as specified in the [Selectors API](#).

In this article

Syntax
[Usage notes](#)
[Examples](#)
[Specifications](#)
[Browser compatibility](#)
[See also](#)

```
document.querySelector("h1").innerHTML = "JavaScript is Cooler";
```

mdn web docs References Guides Plus Blog Play AI Help BETA Theme Log in Sign up for free

References > Web APIs > Document > querySelector()

English (US)

Filter

`open()`
`prepend()`
`queryCommandEnabled() ▲ ↗`
`queryCommandState() ▲ ↗`
`queryCommandSupported() ▲ ↗`
`querySelector() ▼`
`querySelectorAll()`
`releaseCapture() ▲`
`replaceChildren()`
`requestStorageAccess()`
`requestStorageAccessFor() ▲`
`startViewTransition() ▲`
`write()`
`writeln()`

Events

`afterscriptexecute ▲`
`beforescriptexecute ▲`
`copy`
`cut`
`DOMContentLoaded`
`fullscreenchange`
`fullscreenerror`

Examples

Finding the first element matching a class

In this example, the first element in the document with the class "myclass" is returned:

```
JS
const el = document.querySelector(".myclass");
```

Complex selectors

Selectors can also be really powerful, as demonstrated in the following example. Here, the first `<input>` element with the name "login" (`<input name="login"/>`) located inside a `<div>` whose class is "user-panel main" (`<div class="user-panel main">`) in the document is returned:

```
JS
const el = document.querySelector("div.user-panel.main input[name='login']);
```

Negation

As all CSS selector strings are valid, you can also negate selectors:

```
JS
const el = document.querySelector(
  "div.user-panel:not(.main) input[name='login']",
);
```

```
document.querySelector("h1").innerHTML = "JavaScript is Cooler";
```

mdn web docs References Guides Plus Blog Play AI Help BETA Theme Log in Sign up for free

References > Web APIs > Document > querySelector() English (US)

ref-for-dom-parentnode-queryselector①

Filter

open()

prepend()

queryCommandEnabled() ⚠️ ↗

queryCommandState() ⚠️ ↗

queryCommandSupported() ⚠️ ↗

querySelector()

querySelectorAll()

releaseCapture() ⚠️

replaceChildren()

requestStorageAccess()

requestStorageAccessFor() ⚠️

startViewTransition() ⚠️

write()

writeln()

Events

afterscriptexecute ⚠️

beforescriptexecute ⚠️

copy

cut

DOMContentLoaded

Browser compatibility

Report problems with this compatibility data on GitHub ↗

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
querySelector	✓ 1	✓ 12	✓ 3.5	✓ 10	✓ 3.1	✓ 18	✓ 4	✓ 10.1	✓ 2	✓ 1.0	✓ 4.4

Tip: you can click/tap on a cell for more information.

✓ Full support

See also

- [Locating DOM elements using selectors](#)
- [Element.querySelector\(\)](#)
- [Document.querySelectorAll\(\)](#)
- [Element.querySelectorAll\(\)](#)

Try it yourself

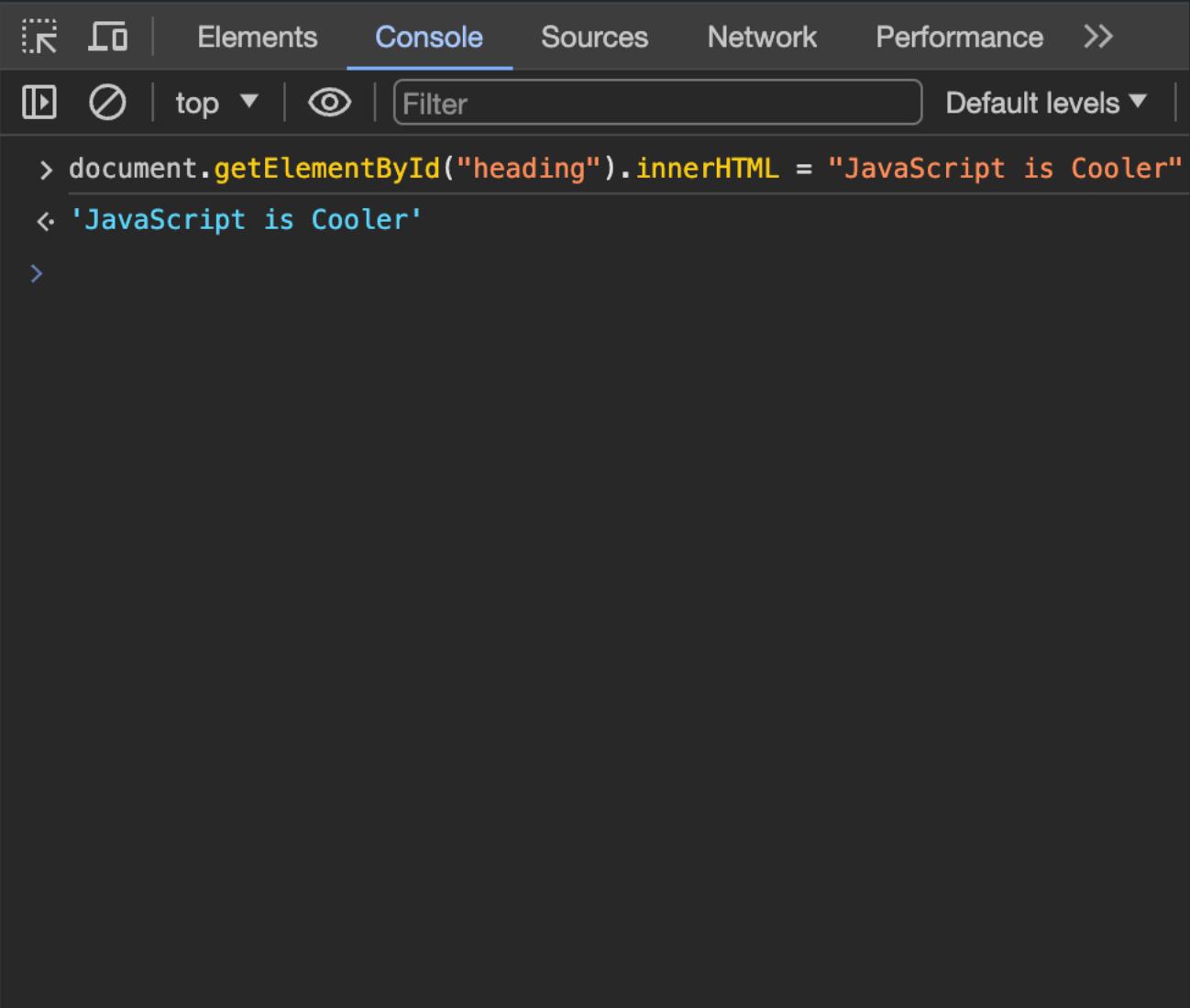
<https://codeload.github.com/mdn/beginner-html-site-styled/zip/refs/heads/gh-pages>

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

```
<h1 id="heading">Mozilla is cool</h1>
```



The screenshot shows the Mozilla homepage with a yellow background. At the top, the text "JavaScript is Cooler" is displayed in large blue letters. Below it is the Mozilla Firefox logo, which is a stylized fox made of orange, yellow, and red flames. The main text on the page reads: "At Mozilla, we're a global community of". Below this is a bulleted list: "• technologists", "• thinkers", and "• builders". Further down, there is a paragraph about Mozilla's mission: "working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future." At the bottom, there is a link: "Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission."



The screenshot shows the Firefox Developer Tools with the "Console" tab selected. The console interface includes tabs for "Elements", "Console" (which is active), "Sources", "Network", and "Performance". There are also icons for "Reset" and "Clear" at the top left, and a "Filter" input field. The console output shows the following command and its result:

```
> document.getElementById("heading").innerHTML = "JavaScript is Cooler"
< 'JavaScript is Cooler'
>
```

Mozilla is cool



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

Elements Console Sources Network

top Filter

```
> document.querySelectorAll("p")
```

```
< ▾ NodeList(3) [p, p, p] i
```

```
  ▶ 0: p
  ▶ 1: p
  ▶ 2: p
    length: 3
  ▶ [[Prototype]]: NodeList
```

```
>
```

Mozilla is cool



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

Elements Console Sources Network

top Filter

```
> document.querySelectorAll("p")
```

```
< ▼ NodeList(3) [p, p, p] i
  ► 0: p
  ► 1: p
  ► 2: p
    length: 3
  ► [[Prototype]]: NodeList
```

```
>
```

```
> document.querySelectorAll("p")
```

```
< ▼ NodeList(3) [p, p, p] i
  ▼ 0: p
    accessKey: ""
    align: ""
    ariaAtomic: null
    ariaAutoComplete: null
    ariaBrailleLabel: null
    ariaBrailleRoleDescription: null
    ariaBusy: null
    ariaChecked: null
    ariaColCount: null
    ariaColIndex: null
    ariaColSpan: null
    ariaCurrent: null
    ariaDescription: null
    ariaDisabled: null
    ariaExpanded: null
```

Mozilla is cool



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

Elements Console Sources Network Performance Memory

top Filter

```
> document.getElementById("heading")
    .addEventListener("mouseover", () => console.log("squeek"))
```

```
< undefined
```

```
② squeek
```

```
>
```

Mozilla is cool



At Mozilla, we're a global community of

- technologists
- thinkers
- builders

working together to keep the Internet alive and accessible, so people worldwide can be informed contributors and creators of the Web. We believe this act of human collaboration across an open platform is essential to individual growth and our collective future.

Read the [Mozilla Manifesto](#) to learn even more about the values and principles that guide the pursuit of our mission.

```
Elements Console Sources Network Performance Memory >
Default levels ▾ | N
document.getElementById("heading")
  .addEventListener("click", (event) => event.target.style.background="pink")
< undefined
>
```

What did we just see?????????



What did we just see?

- We opened our Dev Tools and Selected the JavaScript Console
- We changed the inner HTML of the heading (h1) element
 - `document.querySelector("h1").innerHTML = "JavaScript is Cooler";`
- This is an example of changing the content of our page using JavaScript
- Then we attached multiple event listeners to our heading (h1) element
 - `document.getElementById("heading")`
 - `.addEventListener("mouseover", () => console.log("squeek"))`
 - `.addEventListener("click", (event) => event.target.style.background = "pink")`
- This is an example of responding to user input, either directly or indirectly
- All of this is client-side, browser-based JavaScript
- We are manipulating the DOM or Document Object Model
- We will cover all these examples in detail in the coming lectures
- But for now.....



Step Back



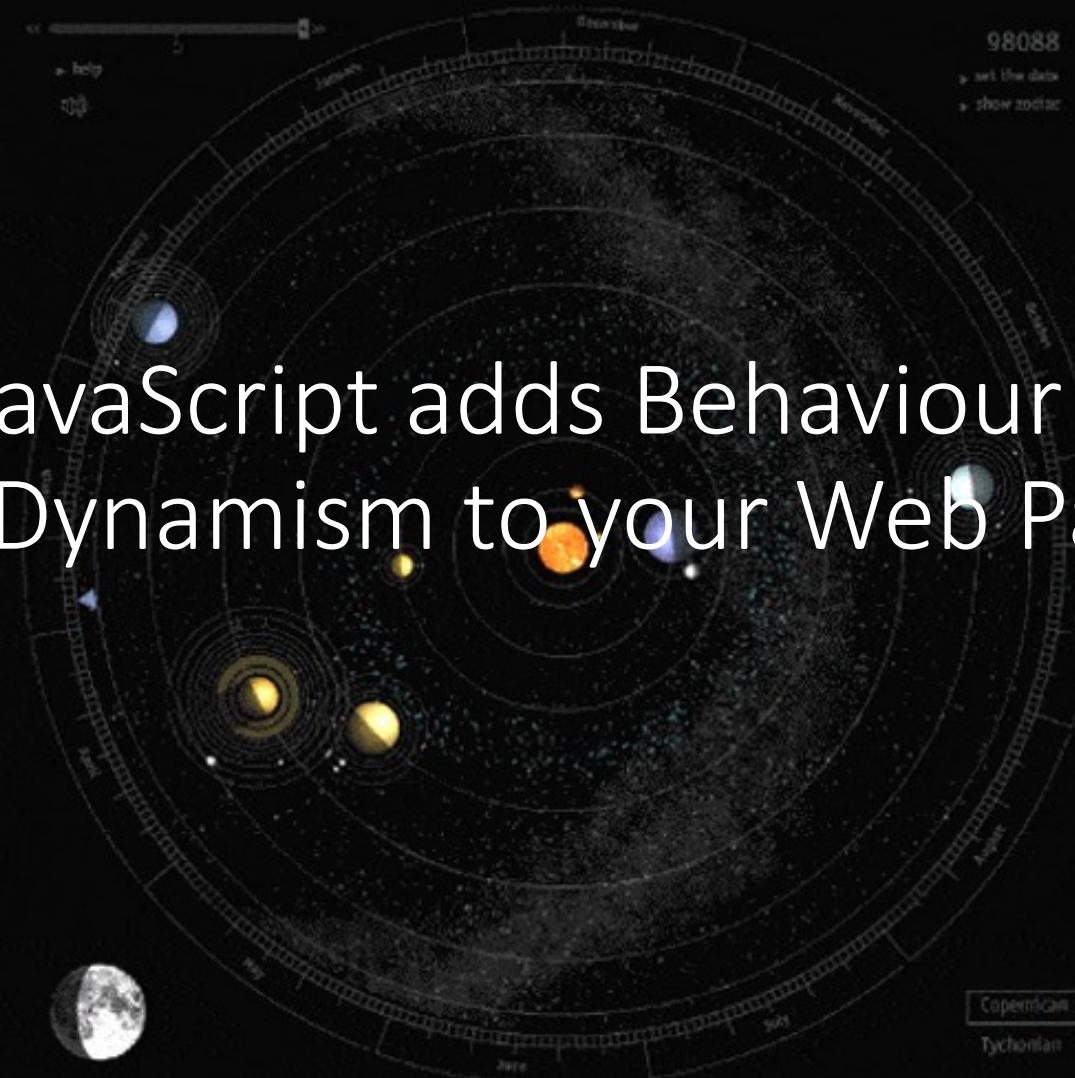
The background image shows the Golden Gate Bridge in San Francisco, California, during a golden hour. The bridge's towers are illuminated by the low sun, casting long shadows and creating a warm glow over the water of the San Francisco Bay. The sky is a mix of deep blues and vibrant oranges. In the foreground, there's a grassy, slightly hilly area with some low-lying plants, suggesting a coastal or hillside vantage point.

HTML defines the Structure
of your Web Page

A reproduction of Vincent van Gogh's painting "The Starry Night". It depicts a dark, swirling night sky filled with numerous small, yellow stars and a large, bright crescent moon in the upper right. In the foreground, a dark, silhouetted cypress tree stands on the left, and a small town with houses and church spires is nestled in a valley below, illuminated by a soft light. The style is characterized by thick, expressive brushstrokes.

CSS is responsible for the
Presentation and Styling of
that Structure

JavaScript adds Behaviour and
Dynamism to your Web Page





JavaScript is a programming language that adds interactivity to your website.

This happens in games, in the behaviour of responses when buttons are pressed or with data entry on forms; with dynamic styling; with animation, etc.



JavaScript is a powerful programming language that can add interactivity to a website. It was invented by Brendan Eich.

It is versatile and beginner-friendly.

With more experience, you'll be able to create games, animated 2D and 3D graphics, comprehensive database-driven apps, and much more!

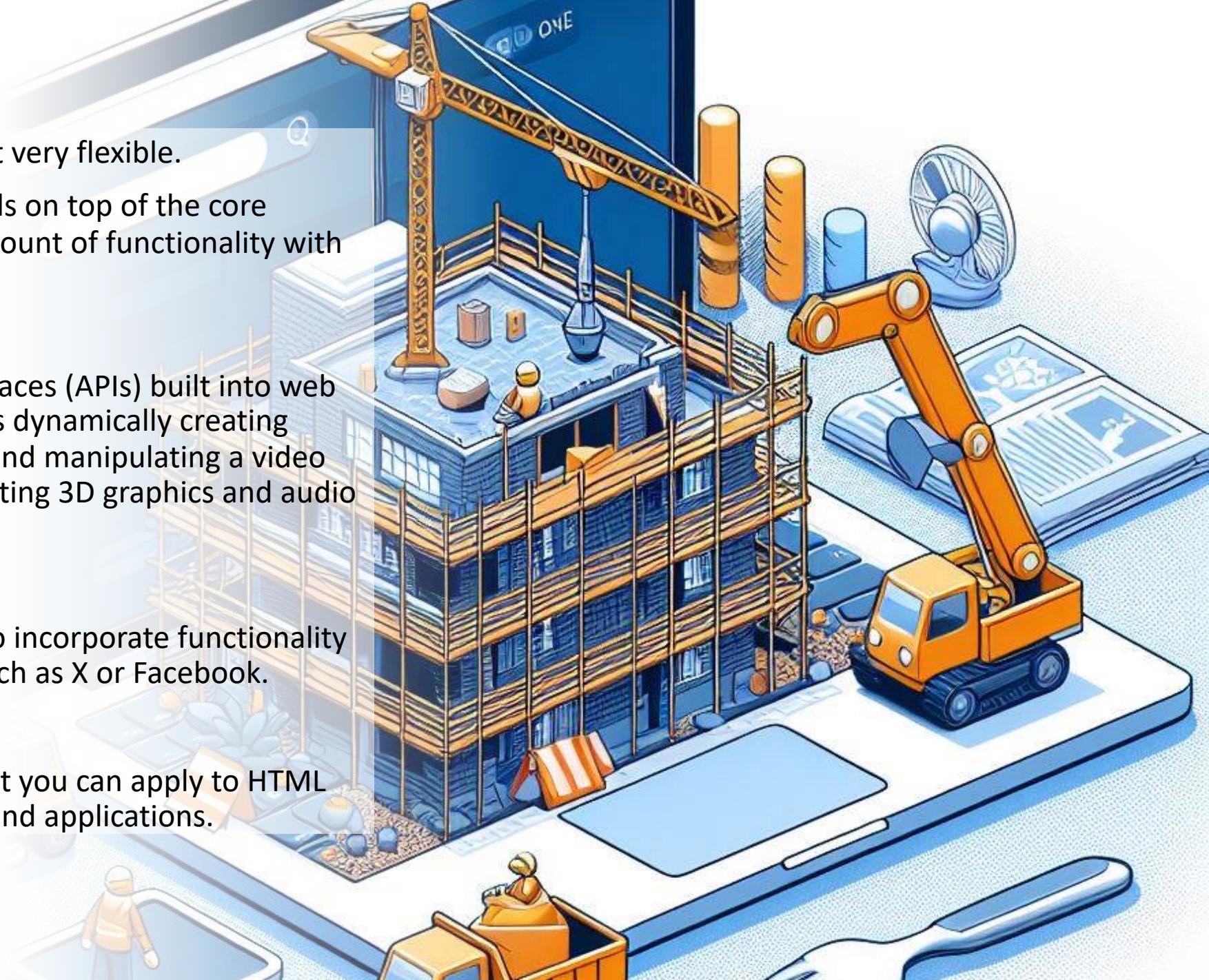
JavaScript itself is relatively compact, yet very flexible.

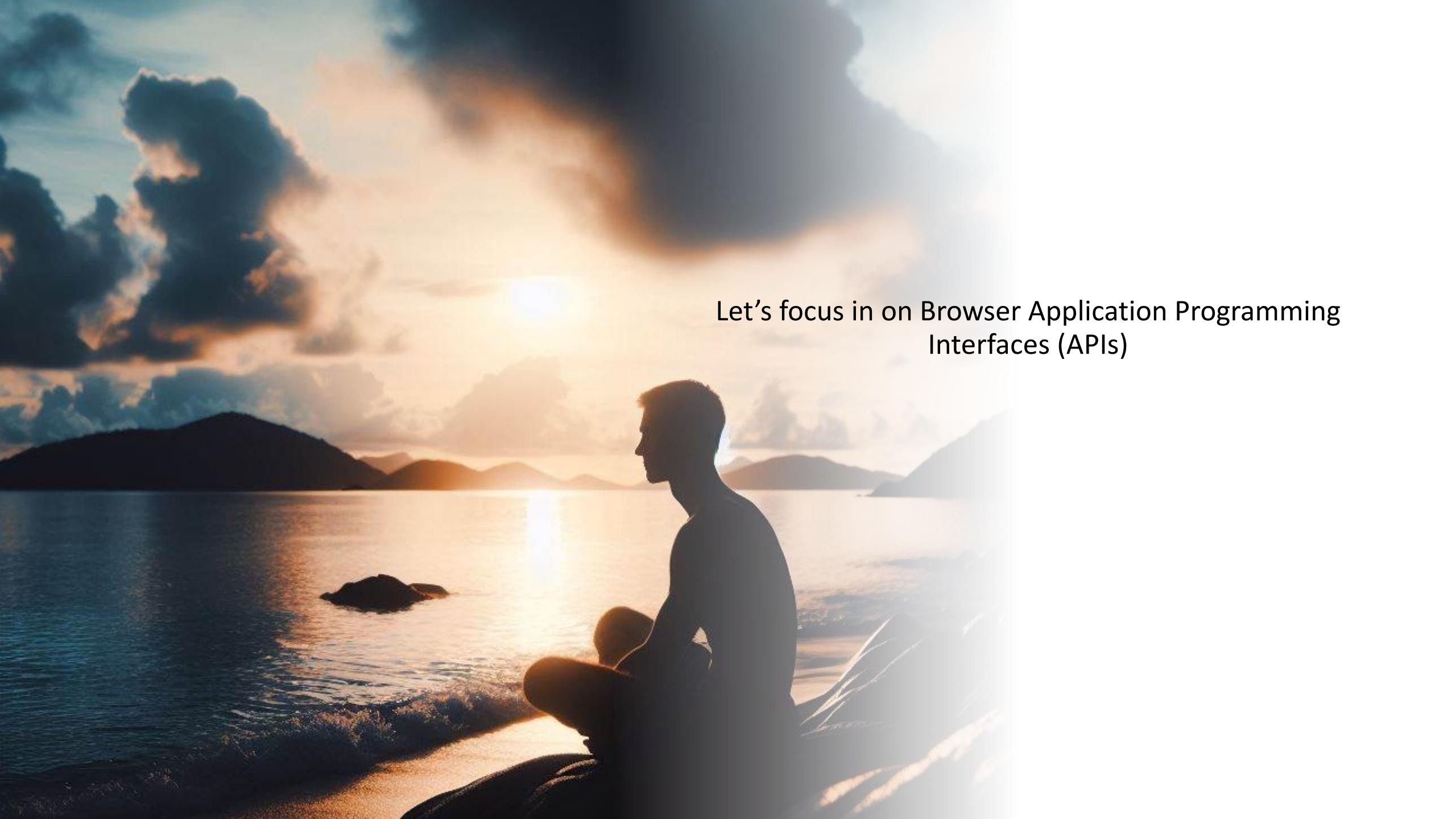
Developers have written a variety of tools on top of the core JavaScript language, unlocking a vast amount of functionality with minimum effort. These include:

Browser Application Programming Interfaces (APIs) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.

Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as X or Facebook.

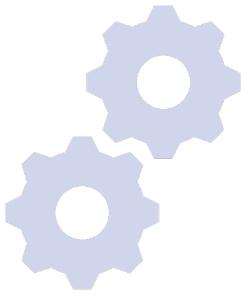
Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications.



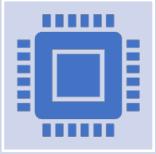
A photograph of a man sitting in a meditative lotus pose on a sandy beach. He is silhouetted against the bright horizon where the sun is setting over a body of water. In the background, there are dark, silhouetted mountains and a sky filled with large, billowing clouds. The overall atmosphere is peaceful and contemplative.

Let's focus in on Browser Application Programming
Interfaces (APIs)

Client-side web APIs



When writing client-side JavaScript for websites or applications, you will quickly encounter Application Programming Interfaces (APIs).



APIs are programming features for manipulating different aspects of the browser and operating system the site is running on, or manipulating data from other websites or services.

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs

What is an API?

- Application Programming Interfaces (APIs) are constructs made available in programming languages to allow developers to create complex functionality more easily.
- APIs abstract more complex code away from you, providing some easier syntax to use in its place.

```
const map = new  
Map(document.getElementById("map"), {  
  zoom: 4,  
  center: position,  
  mapId: "DEMO_MAP_ID",  
});  
  
// The marker, positioned at Uluru  
const marker = new  
AdvancedMarkerElement({  
  map: map,  
  position: position,  
  title: "Uluru",  
});
```



[Get Started](#)[Contact sales](#)[Guides](#) [Reference](#) [Samples](#) [Support](#)[Filter](#)

Maps JavaScript API

[Overview](#)
[Set up your Google Cloud project](#)
[Use API Keys](#)
[Load the Maps JavaScript API](#)

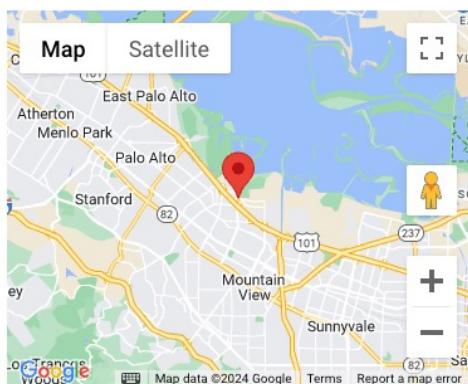
Tutorials

[All tutorials](#)
[Add a Google Map with a marker to your website](#)
[Cluster markers](#)
[Real-time collaborative mapping](#)
[Show current location](#)
▶ [Use data with your map](#)

Concepts

[All concepts](#)
[Map types](#)
[Map and tile coordinates](#)
[Localizing the map](#)
[Versioning](#)
[Best practices](#)
[Using TypeScript](#)
[Promises](#)
[Manage marker label collisions](#)

Customize a map



Maps JavaScript API

Build dynamic, interactive, deeply customized maps, location, and geospatial experiences for your web apps.

 Search Maps JS API docs

Get Started

Start building with commonly used features of the Maps JavaScript API.



[Get started with Google Maps Platform](#)

Follow the Google Maps Platform getting started guide to create an account, generate an API key, and



[Build your first map with a marker](#)

Learn how to load the Maps JavaScript API, and add a map with a marker to your web app.

On this page
[Get Started](#)
[Features](#)
[Libraries](#)
[Services](#)
[Example apps](#)
[Help & support](#)



I thought an API was a Web Service or REST Endpoint?

- API can be used to describe :
 - A software “package” that simplifies or enables a specific task, service or feature
 - Google Maps API
 - Twilio API
 - Browser Document Object Model
 - The Fetch API to retrieve data from the web
 - Connect to a database from your java code
 - A HTTP based Web Service or REST API
 - These are API's hosted on the internet (or a private network)
 - Typically accessed via URLs or “endpoints”
 - Follow a particular protocol (usually HTTP) with a specific spec or set of principal, e.g. REST, SOAP or proprietary
 - We will talk more about these in Web Dev II
 - You will come across more uses of the term API

JavaScript is Cooler

Browser APIs – DOM (Document Object Model) API



- Allows you to manipulate HTML and CSS — creating, removing and changing HTML, dynamically applying new styles to your page, etc.
- Every time you see a popup window appear on a page or some new content displayed, for example, that's the DOM in action

er to keep the Internet alive and accessible, so people
be informed contributors and creators of the Web. We
of human collaboration across an open platform is
individual growth and our collective future.

[The Mozilla Manifesto](#) to learn even more about the values and
guide the pursuit of our mission.

```
Elements Console S
Filter
> document.querySelector("h1").i
< 'JavaScript is Cooler'
> |
```

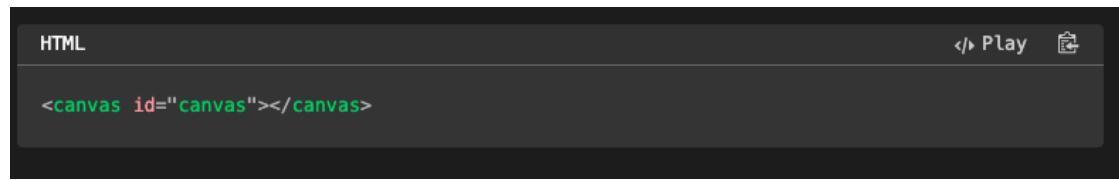
Browser APIs – Fetch API

```
async function logMovies() {  
  const response = await fetch("http://example.com/movies.json");  
  const movies = await response.json();  
  console.log(movies);  
}
```



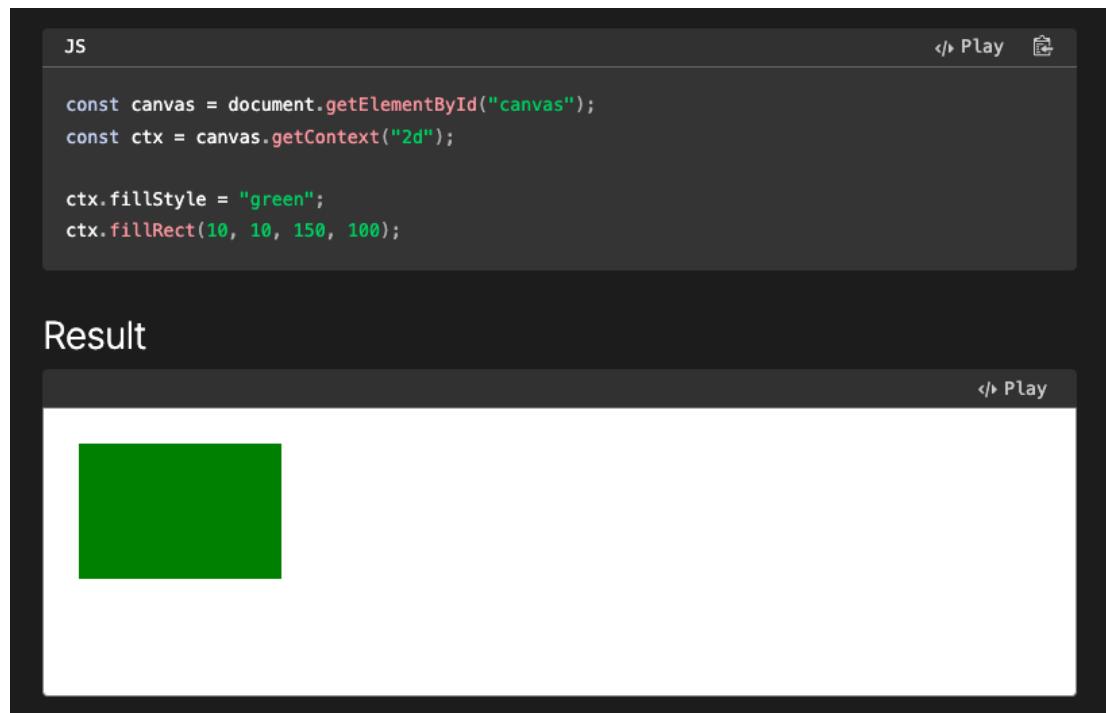
- APIs that fetch data from the server to update small sections of a webpage on their own are very commonly used.
- This seemingly small detail has had a huge impact on the performance and behaviour of sites — if you just need to update a stock listing or list of available new stories, doing it instantly without having to reload the whole entire page from the server can make the site or app feel much more responsive and "snappy".
- The main API used for this is the [Fetch API](#)

Browser APIs - Canvas



HTML

```
<canvas id="canvas"></canvas>
```



JS

```
const canvas = document.getElementById("canvas");
const ctx = canvas.getContext("2d");

ctx.fillStyle = "green";
ctx.fillRect(10, 10, 150, 100);
```

Result



- APIs for drawing and manipulating graphics are widely supported in browsers — the most popular ones are [Canvas](#) and WebGL, which allow you to programmatically update the pixel data contained in an HTML <canvas> element to create 2D and 3D scenes.
- For example, you might draw shapes such as rectangles or circles, import an image onto the canvas, and apply a filter to it such as sepia or grayscale using the Canvas API.

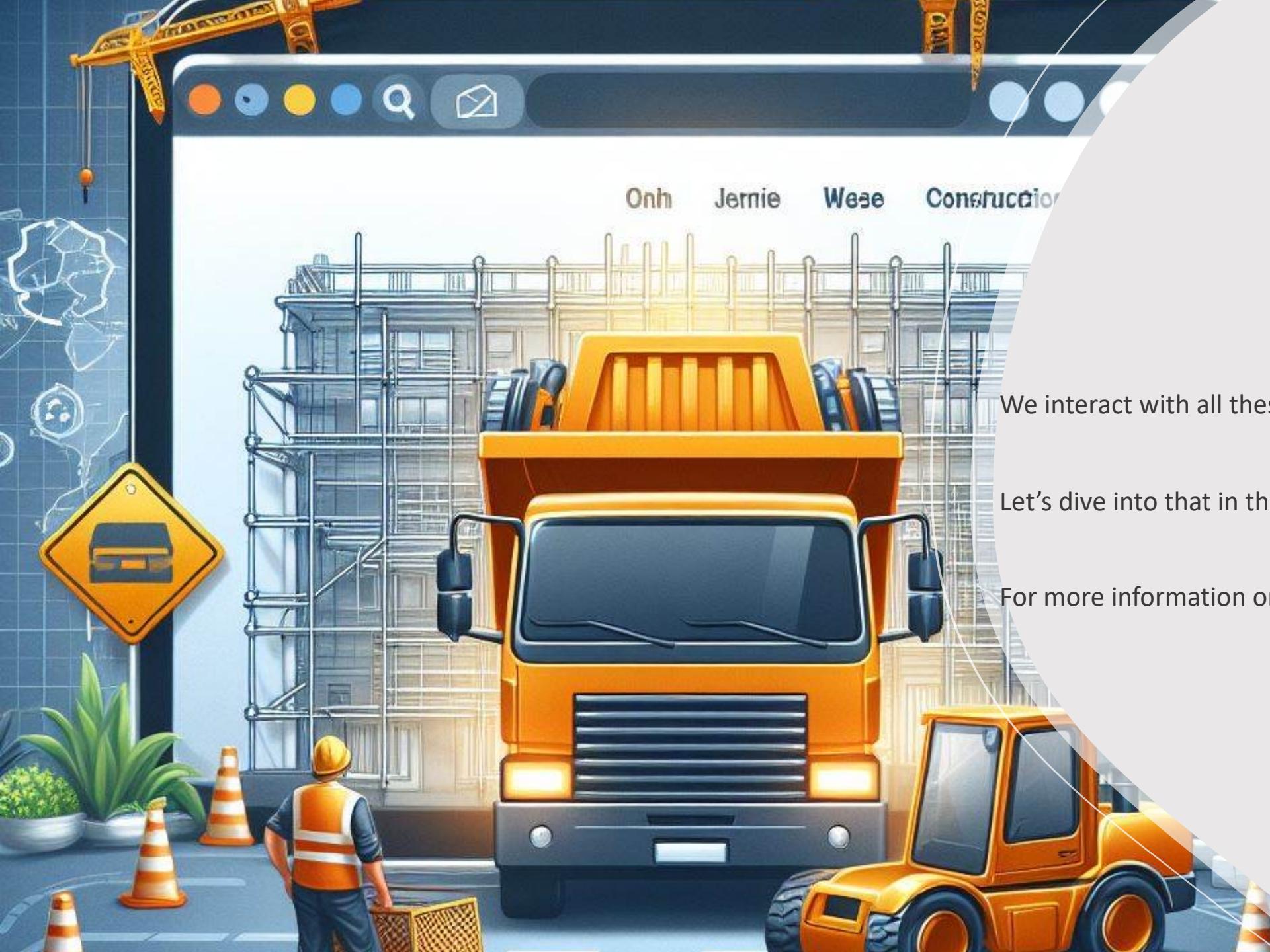
Browser APIs - Geolocation



```
JS
navigator.geolocation.getCurrentPosition((position) => {
  doSomething(position.coords.latitude, position.coords.longitude);
});
```

- Device APIs enable you to interact with device hardware: for example, accessing the device GPS to find the user's position using the [Geolocation API](#).

```
JS
const watchID = navigator.geolocation.watchPosition((position) => {
  doSomething(position.coords.latitude, position.coords.longitude);
});
```



We interact with all these APIs with JavaScript

Let's dive into that in the next section

For more information on Browser APIs → [MDN](#)