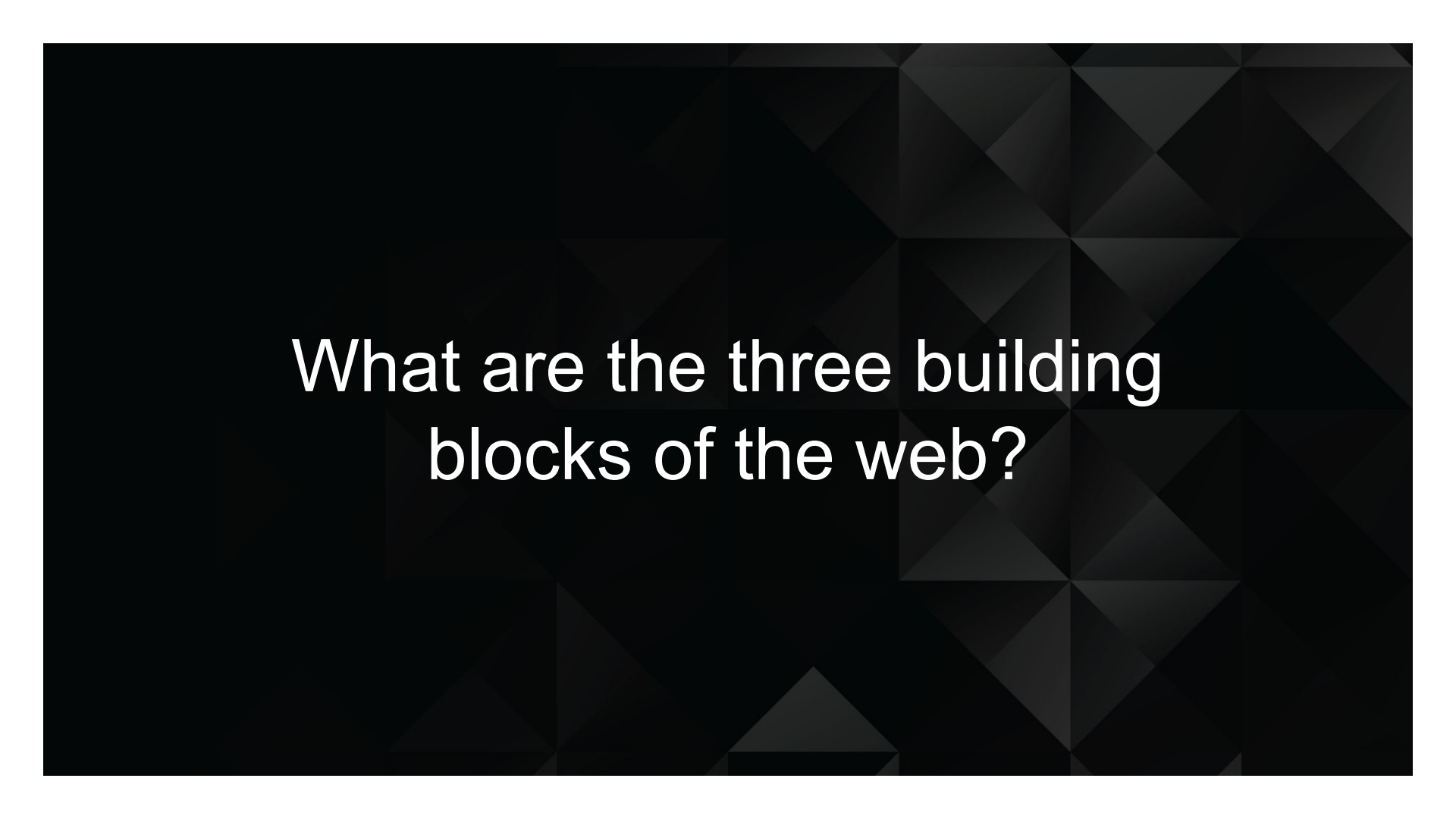




# Introduction to the DOM

Web Development  
Lesson 2.3









What are the three building  
blocks of the web?

# Building Blocks of the Web

---

HTML	CSS	JavaScript
<p data-bbox="131 353 471 386">Used to write content.</p> <div data-bbox="249 579 469 892"><p data-bbox="280 579 438 623"><b>HTML</b></p></div>	<p data-bbox="710 353 1074 386">Used to format content.</p> <div data-bbox="853 579 1072 886"><p data-bbox="904 579 1023 623"><b>CSS</b></p></div>	<p data-bbox="1290 353 1789 554">Used to create dynamic web applications that take in user inputs, change what's displayed to users, animate elements, and much more.</p> <div data-bbox="1425 579 1644 892"><p data-bbox="1497 579 1574 623"><b>JS</b></p></div>

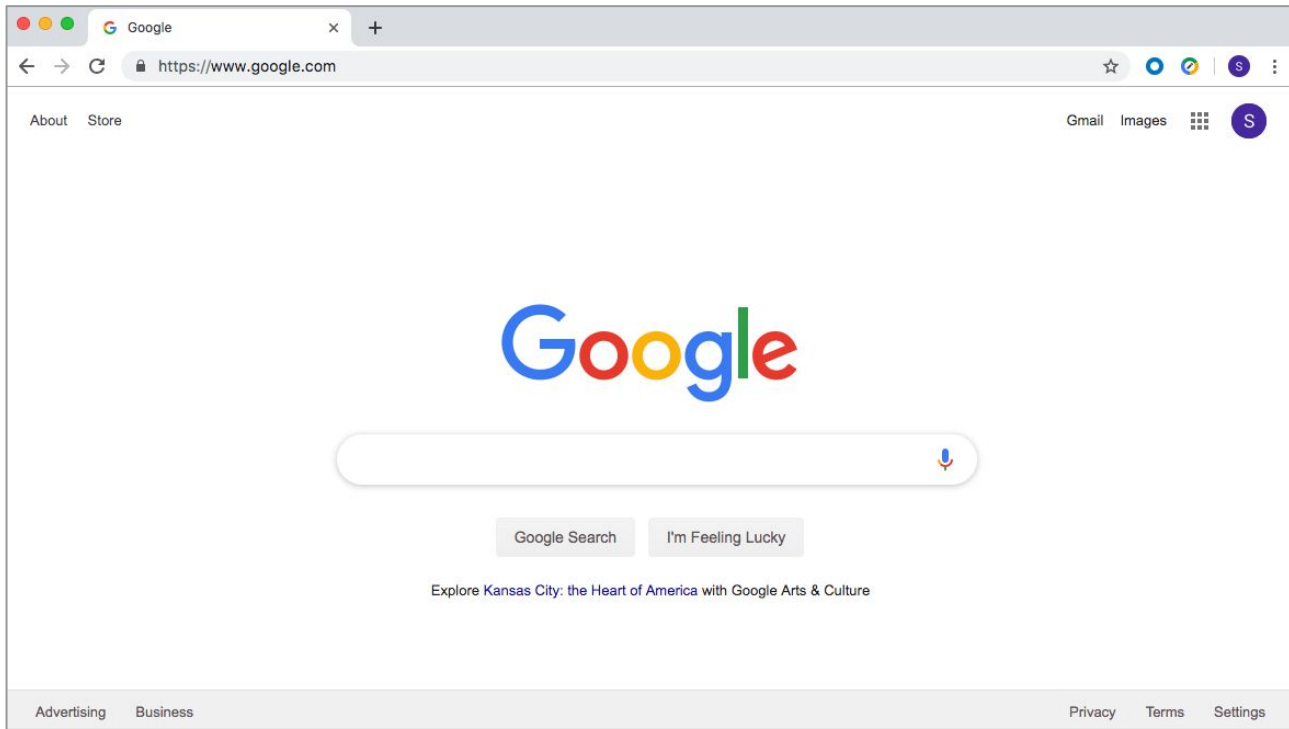


How (or where) do we  
connect all three?

# They Are Connected in the Web Browser

---

The browser brings together HTML, CSS, and JavaScript to create interactive webpages and applications.

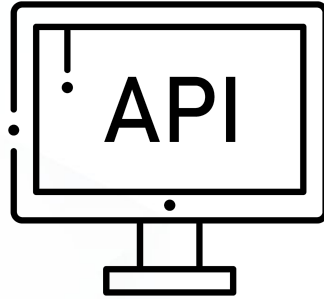




What is a web browser?

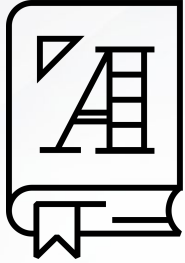


A **web browser**, or **browser**, is a program used to access information on the World Wide Web. Every webpage, image, and video on the web can be accessed via a specific Unified Resource Link (URL). This lets browsers retrieve these resources from a web server and display them on a user's device.



**What is an application  
programming interface (API)?**





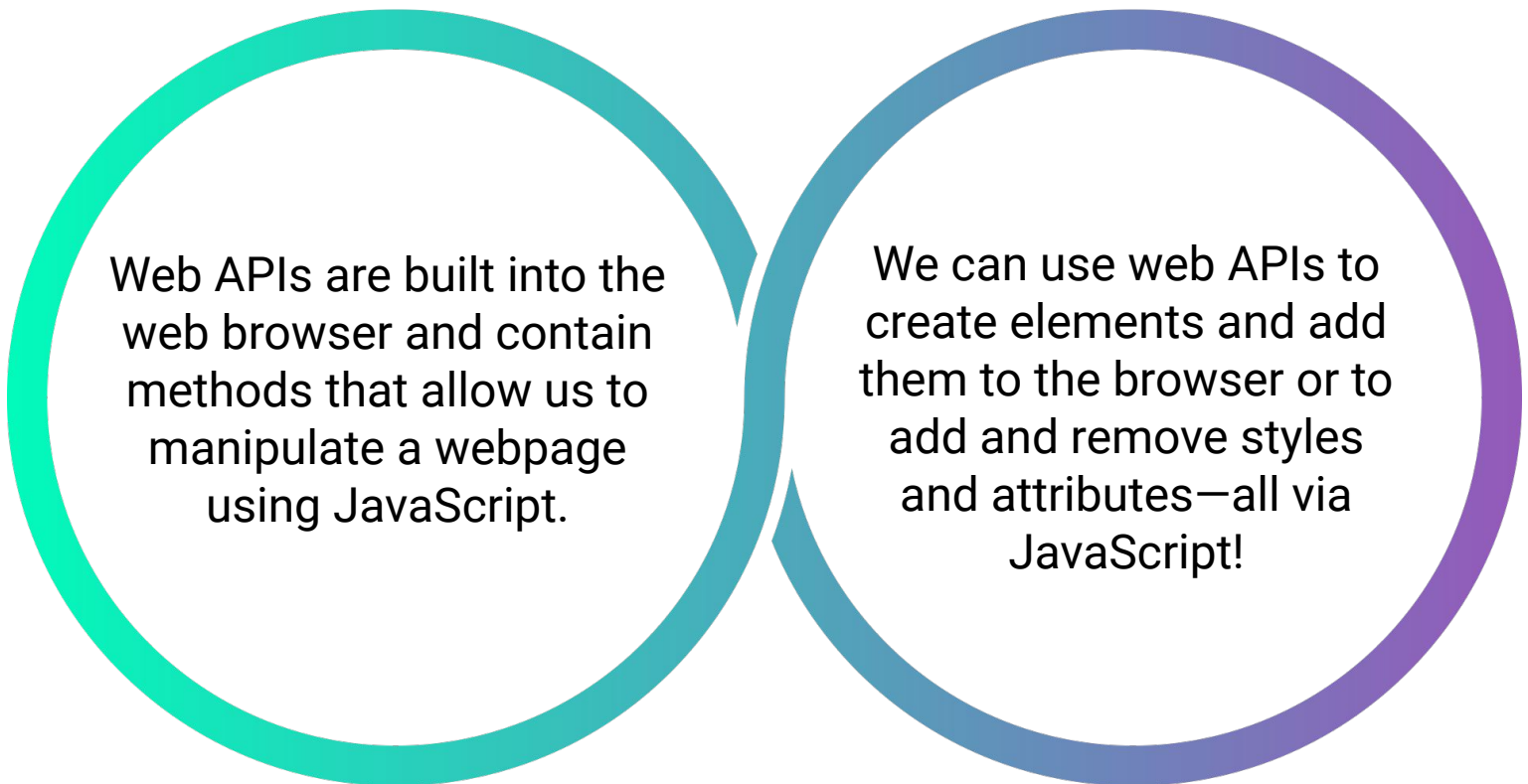
In web development, an **API** is a set of code features (methods, properties, events, and URLs) that developers can use in their apps to interact with components of a user's web browser, data sets, hardware/software on a user's computer, or third-party software and services.



What are web APIs?

# Web APIs

---



Web APIs are built into the web browser and contain methods that allow us to manipulate a webpage using JavaScript.

We can use web APIs to create elements and add them to the browser or to add and remove styles and attributes—all via JavaScript!

---



## **Activity:** This Window

See instructions in `01-Stu_This-Window` in the class repo.

In this activity, you will use `console.log(this)` and dig around inside the returned object, answering some questions along the way.

**Suggested Time:**  
10 minutes



# Activity: This Window

---

## Instructions

- First open the provided `index.html` file in the browser and navigate to the console.

- What is logged?

The `window` object. In this use case, `this` refers to the window. The `window` is an object representation of an open window in a browser.

- Click in the `window` object and begin looking at the numerous properties and methods it contains.
- Make your way down to `document` and click in it.
- Spend some time looking through the properties and methods in `window.document`.





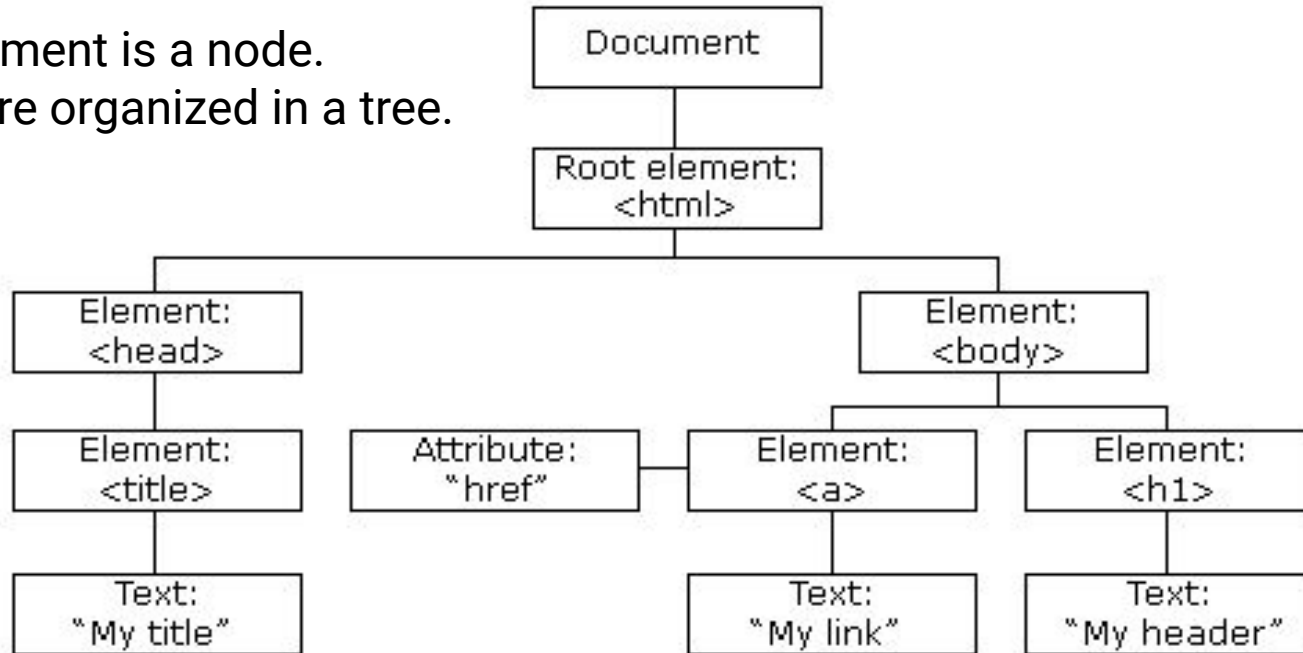
**What is** `window.document`?

# The Document Object Model (DOM)

---

The DOM is an object-oriented representation of HTML (i.e., the HTML document modeled as JavaScript objects).

Each element is a node.  
Nodes are organized in a tree.



# What Is the Node Tree of This HTML?

---

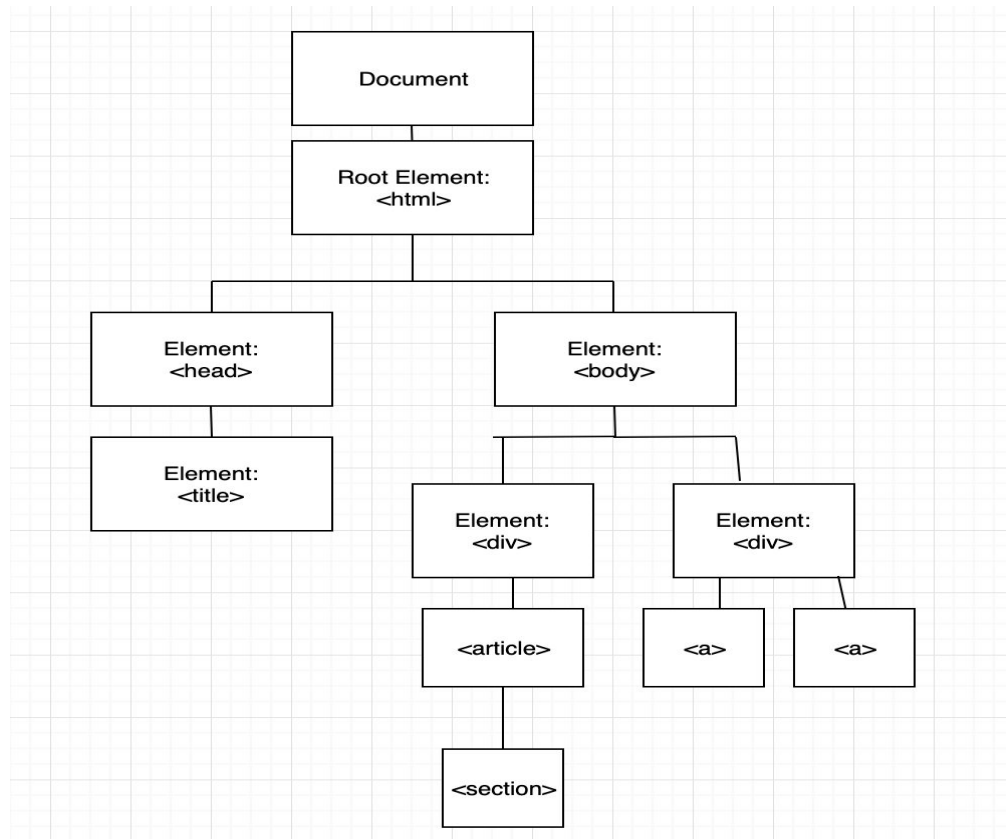
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
    <div>Main div
      <article>
        <section>

          </section>
        </article>
      </div>
      <div>
        <a href="myImg"></a>
        <a href="secondImg"></a>
      </div>
    </body>
  </html>
```

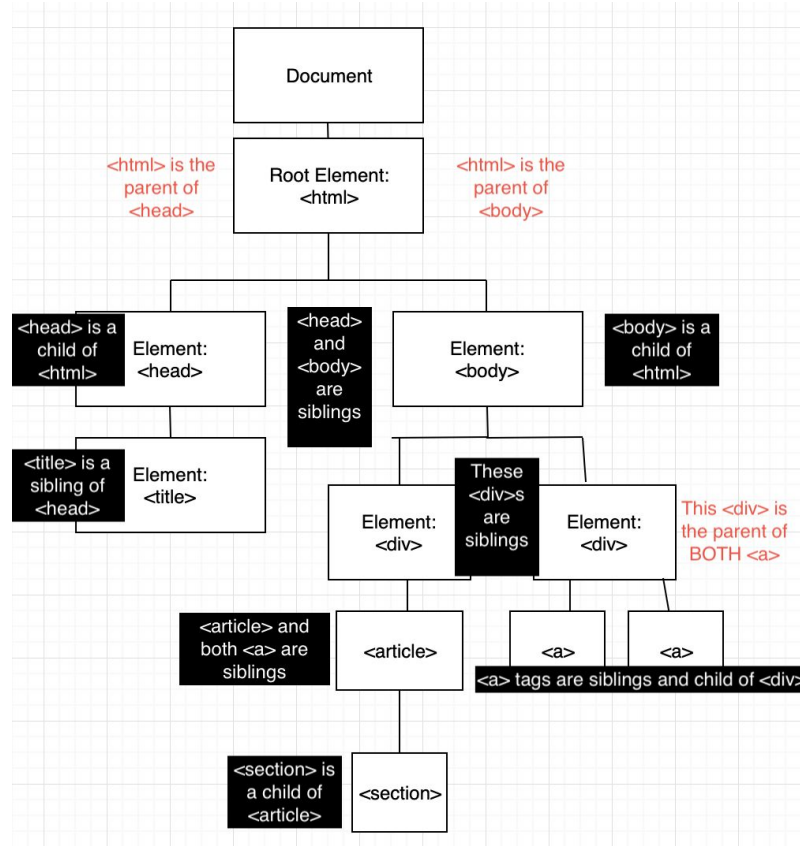


# What Are the Parent, Child, and Sibling Nodes?

---



# Node Relationships: Parents, Children, and Siblings





**What is traversing the DOM?**

# DOM Traversal

---

Navigate to the [MDN DOM Docs](#). Open the Chrome Dev Tools and enter the following commands one by one.

```
console.log(document.body);  
console.log(document.body.children);  
console.log(document.body.children[3]);  
console.log(document.body.children[3].childNodes[7]);  
console.log(document.body.children[3].childNodes[7].style.fontSize = "20px");
```



When using the `style` method, properties with two words (such as `font-size`) become a single word and camelCased. `Font-size` becomes `fontSize`.

Here is one more example of `.style`:

```
console.log(document.body.children[3].childNodes[7].parentElement.style.color = "red");
```

# Our Goal Today

---

Navigate to the deployed [Speed Reader app](#).

