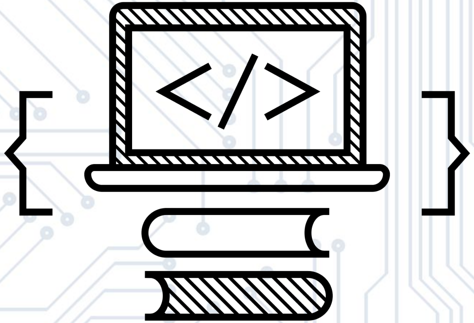


Web APIs

Introduction to the DOM

Web Development

Lesson 4.1

Objectives

In today's class, we'll cover:

- Introducing the DOM / Traversing the DOM
- Setting Attributes
- Create and Append
- Timers and Intervals

What are the three building blocks of the web?

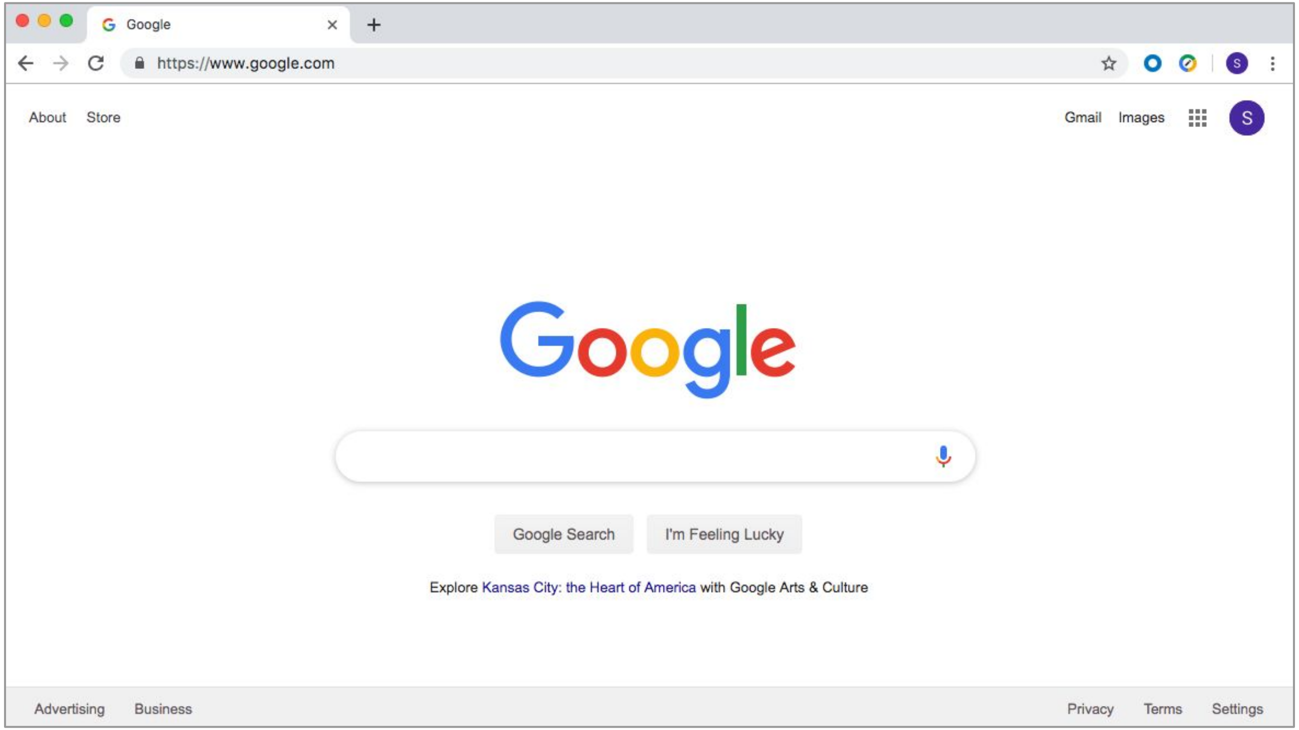
Building Blocks of the Web

HTML	CSS	JavaScript
Used to write content.	Used to format content.	Used to create dynamic web applications that take in user inputs, change what's displayed to users, animate elements, and much more.

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.



Explore Kansas City: the Heart of America with Google Arts & Culture

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.

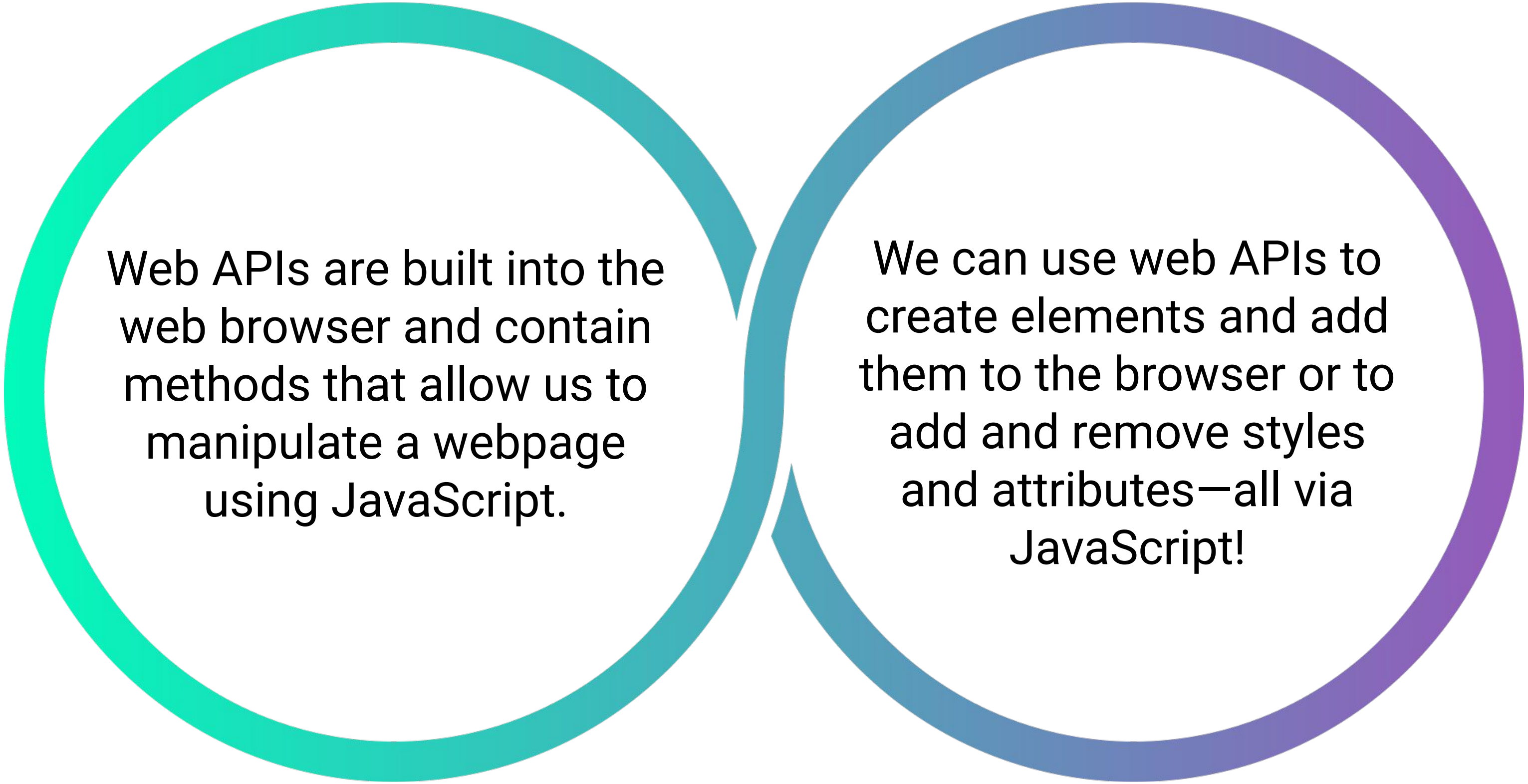
API

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.

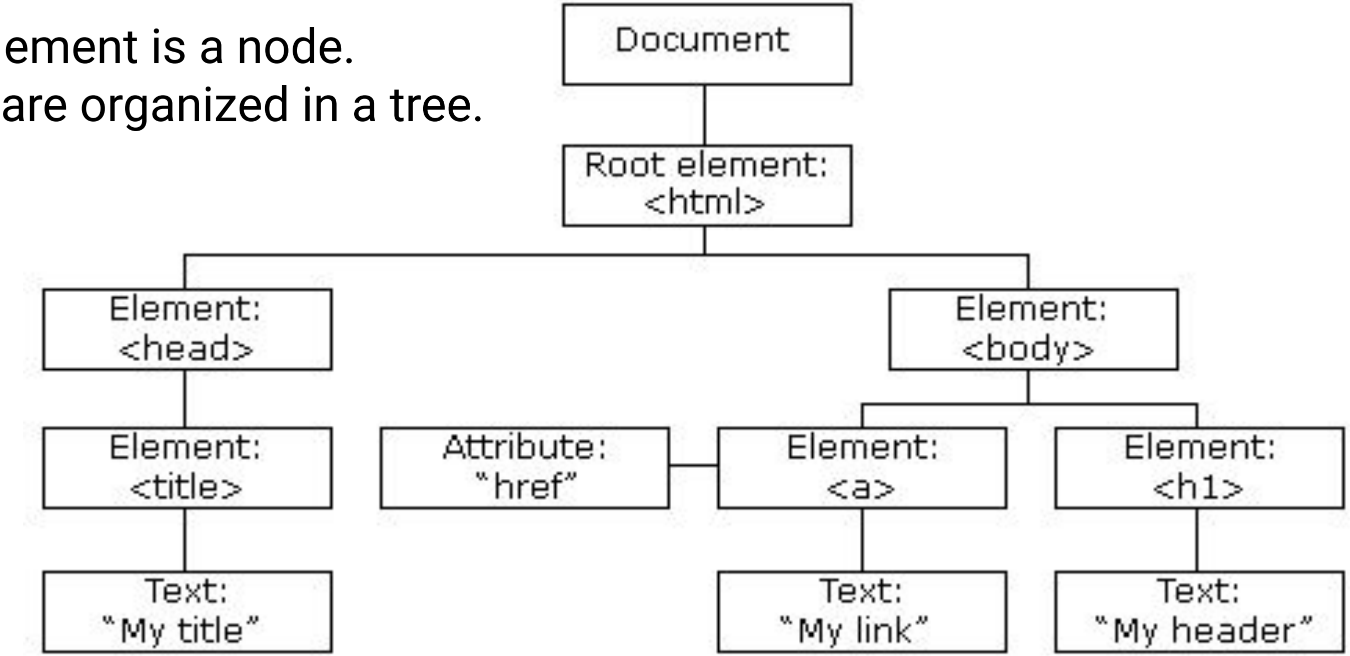
- Spend some time looking through the properties and methods in `window.document`.

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.

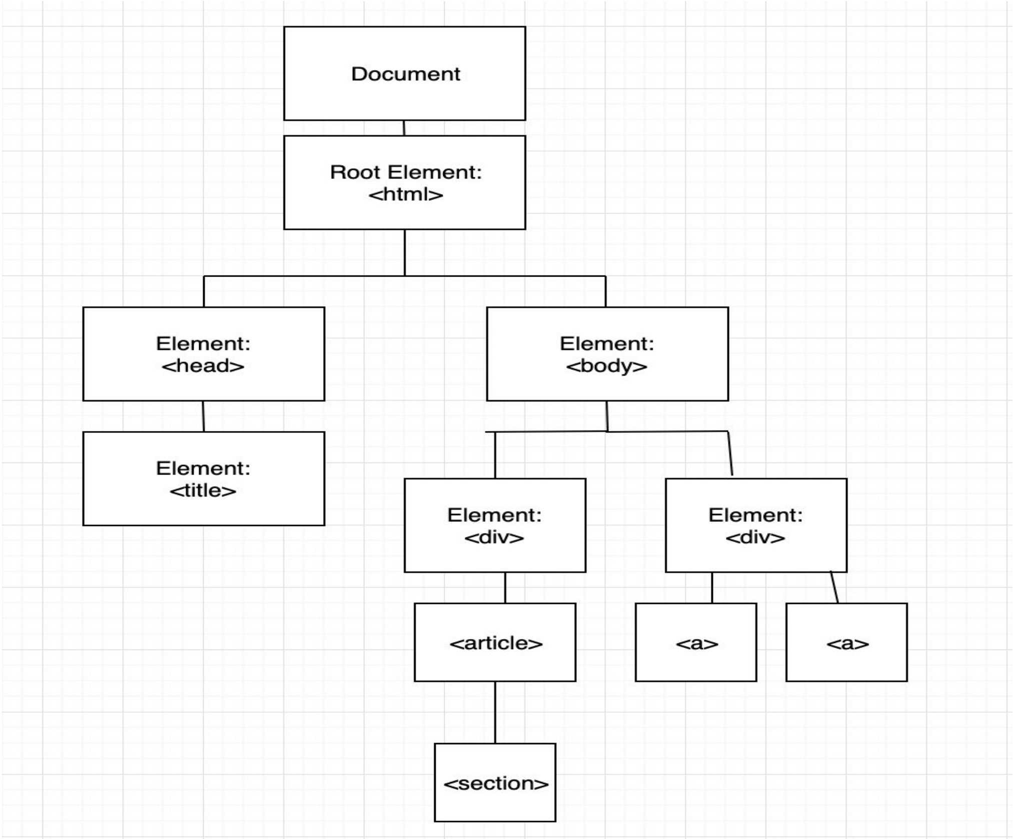


What Is the Node Tree of This HTML?

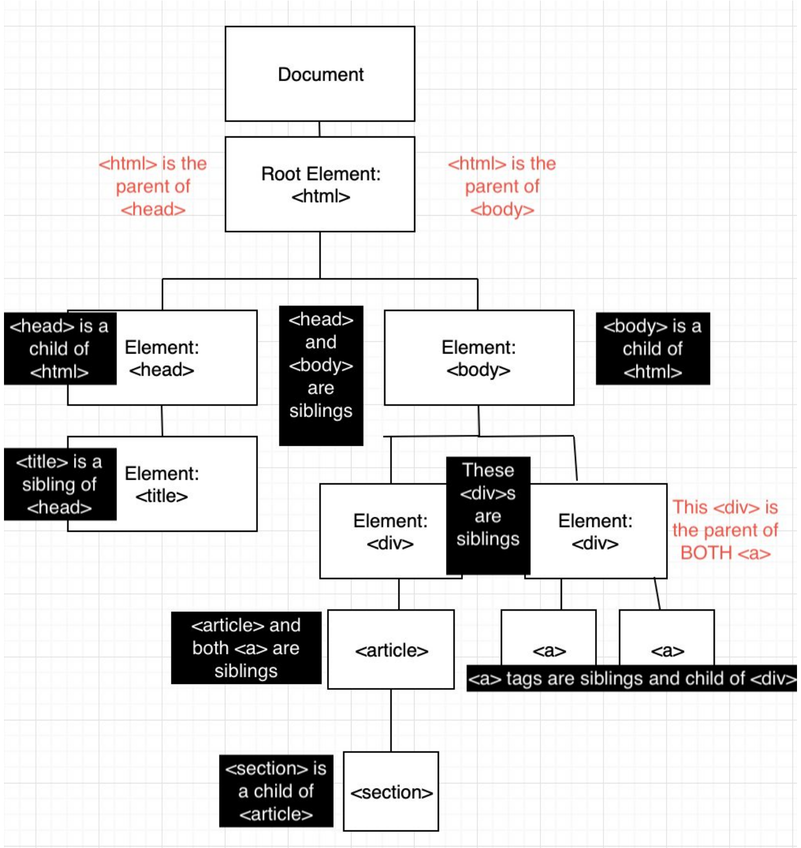
```html
<!DOCTYPE html>
<html lang="en">
   <head>
      <title>Document</title>
   </head>
   <body>
      <div>Main div
         <article>
            <section>

            </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



Document
Root Element:
<html>
<html> is the parent of <head>
<html> is the parent of <body>
<head> is a child of <html>
Element:
<head>
<head> and <body> are siblings
Element:
<body>
<body> is a child of <html>
<title> is a sibling of <head>
Element:
<title>
Element:
<div>
These <div>s are siblings
Element:
<div>
This <div> is the parent of BOTH <a>
<article> and both <a> are siblings
<article>
<a>
<a>
<a> tags are siblings and child of <div>
<section> is a child of <article>
<section>

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");
```

Activity: Traverse That DOM

See instructions in
`03-Stu_Traverse-That-DOM`
in the class repo.

In this activity you are going to use the Chrome Dev Console and `.style.property` to change the styling of elements on our page.

Suggested Time:
10 minutes

Activity: Setting Attributes

See instructions in
`05-Stu_Setting-Attributes`
in the class repo.

In this activity you are going to use JavaScript to add hrefs to a tags, src to img tags, along with some styles all while traversing the DOM.

Suggested Time:
10 minutes

CREATED REPO AND MADE AN INITIAL COMMIT
SO I GUESS YOU COULD SAY THINGS ARE GETTING PRETTY SERIOUS

Activity: Setting Attributes

See instructions in `07-Stu_LookMaNoHTML` in the class repo.

In this activity you are going to use DOM methods to create an entire HTML page..

Suggested Time:
10 minutes

Basics of Timeout

```
function someFunction(){
    alert(“test”);
}

var windowTimeout =
setTimeout(someFunction, 5000);
//clearTimeout(windowTimeout);
```

setTimeout vs setInterval

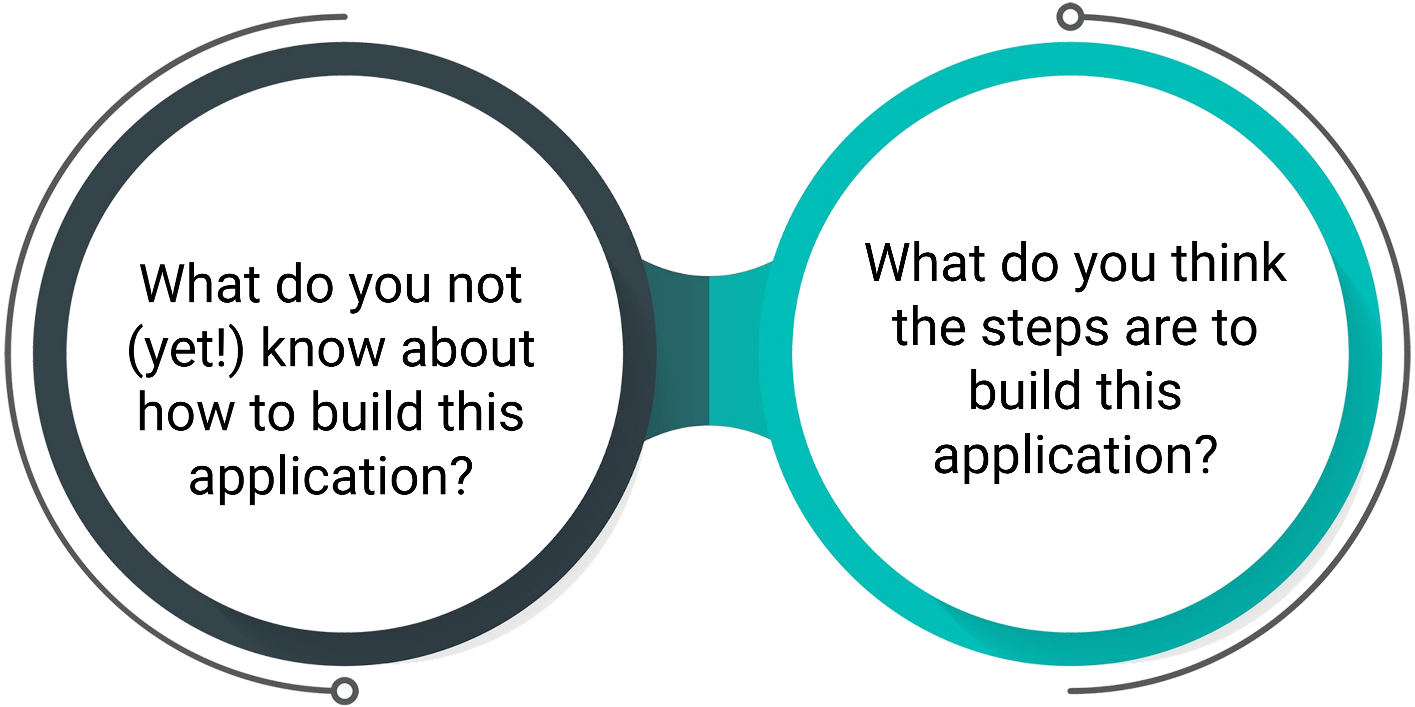
setInterval fires again and again in intervals, while setTimeout only fires once.

setTimeout(expression, timeout);
// runs the code/ function once after the timeout.

setInterval(expression, timeout);
//runs the code/ function in intervals, with the length of the timeout between them.

Our Goal Today

Navigate to the deployed Speed Reader app.



What do you not (yet!) know about how to build this application?

What do you think the steps are to build this application?

Activity: Setting Attributes

See instructions in **09-Stu_SpeedReader** in the class repo.

In this activity you are going to create a speed reading application. It will input a single word on the screen at a time, changing to the next word after 1 second. This equates to reading at 60 Words Per Minute (WPM).

Suggested Time:
10 minutes

<Time to Code>