Front-End Web Development

DOM Manipulation

GENERAL ASSEMBLY

Today's Learning Objectives

In this lesson, you will:

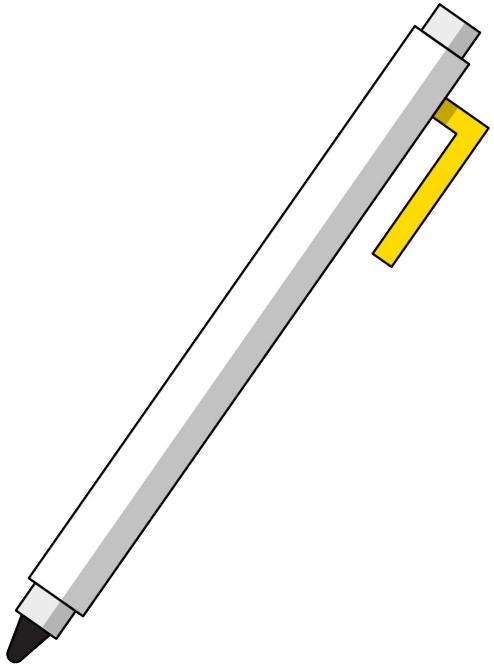
- Identify the role of JavaScript in front-end web development.
- Access properties of the DOM using JavaScript object syntax.
- Use DOM methods to respond to user actions with event listeners.

Group Exercise:
What Is JS?

15 minutes

Now that we've met all three front-end technologies, what exactly does JavaScript do for us? How do HTML, CSS, and JS intersect?

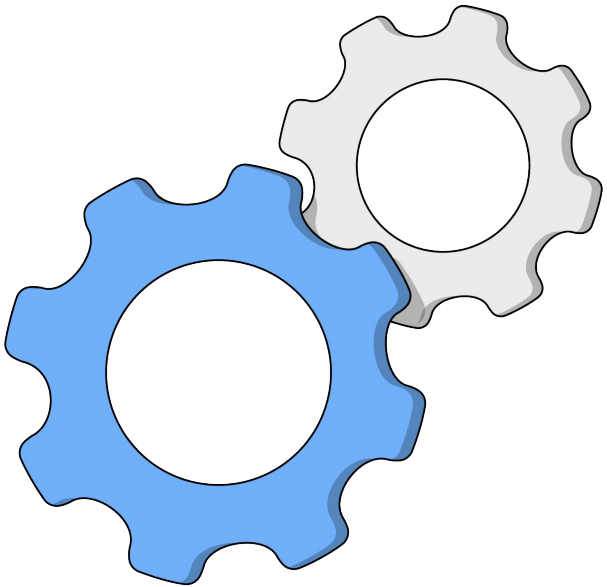
Draw a Venn diagram illustrating the roles of each technology and where they intersect.



 © 2020 General Assembly

What Does JavaScript Do?

- Responds to user actions, called **events**, to change the HTML content/ CSS styles of a webpage.
- Loads data from and sends data to servers, databases, and other websites.
- Programs complex instruction sets like algorithms, machine learning, crypto, etc.

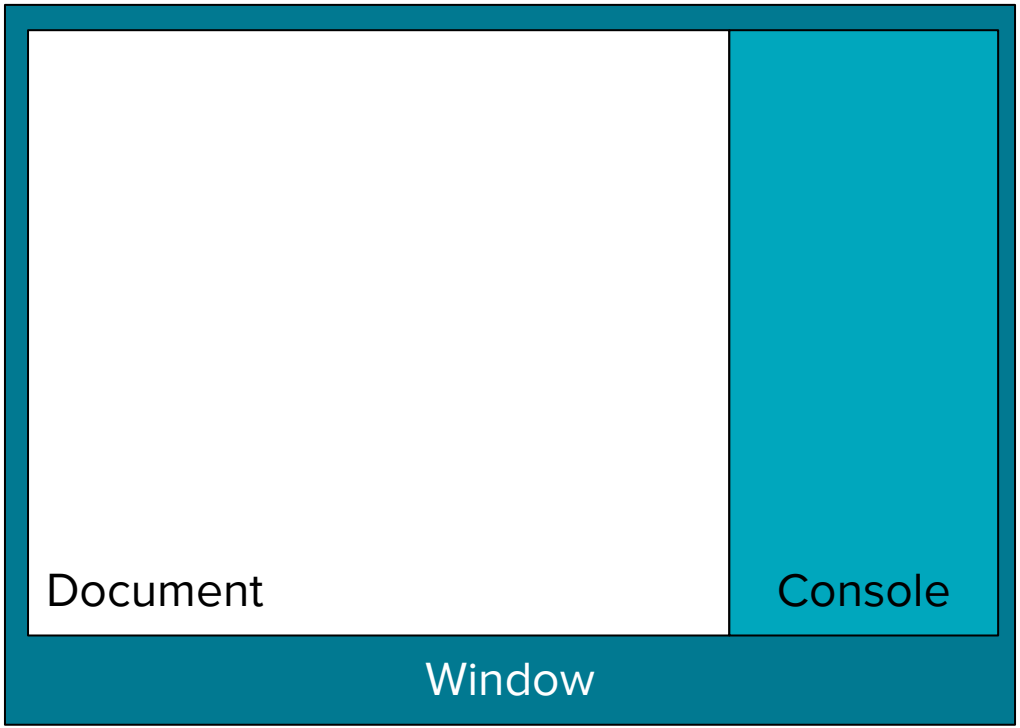


 © 2020 General Assembly

Front-End Web Development

JS on Your Webpage

A Webpage Is One Giant JS Object



 © 2020 General Assembly

Everything you see in the browser is a **JS object**.

Three Big Objects

Window

The whole web browser; mostly used for browser-level settings.

Document

The current webpage. This object has the functionality we want to use when accessing elements on the page.

Console

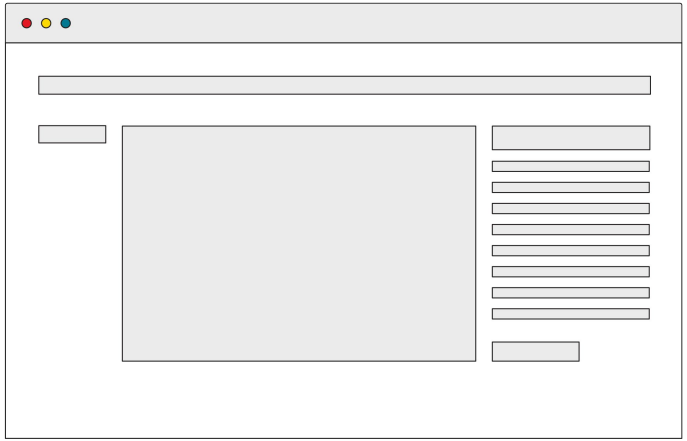
A scratch pad for development-related messages; highly useful in debugging.

 © 2020 General Assembly

The DOM

Browsers read your HTML and create an object in the computer's memory for each part. That HTML layout is called a "data model" because it describes the structure of your webpage.

The **Document Object Model** (DOM) is the browser's JavaScript representation of your HTML elements.



Getters and Setters

The main thing we're doing with JS is getting objects from the DOM and performing actions with them (moving, hiding, etc).

The methods that get something from a webpage are called **getters**.

The methods that change something on the webpage are called **setters**.

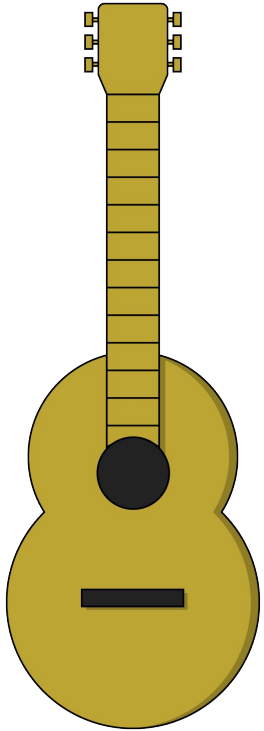
 © 2020 General Assembly

Methods

Getters and setters access and modify objects. They are both types of **methods**. Methods belong to JavaScript objects, including DOM elements.

Think of methods as the **functions** that an object can use. A guitar, for example, might have the following methods:

- `guitar.playChord(chord)`
- `guitar.playNote(note)`
- `guitar.changeTempo(tempo)`
- `guitar.changeVolume(volume)`

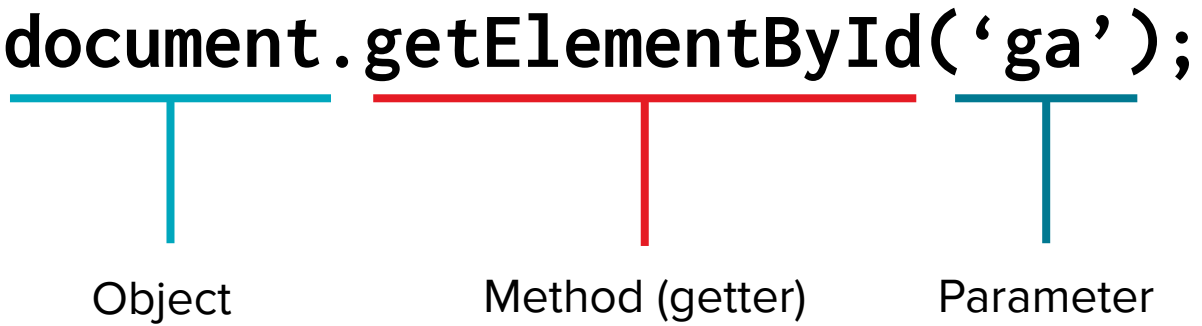


Properties

Objects often have metadata — information that describes the object (height, width, classes, etc). These pieces of information are called **properties**.

Property	Description
`someElement.classList`	A list of the classes belonging to a DOM element.
`someElement.id`	The ID of an element, if it has one.
`someElement.style.color`	The color of an element's text.
`window.location.href`	The window object's location details, including the page's href (hypertext reference/URL).

What Does a Real Piece of Code Look Like?



 © 2020 General Assembly

Getters

Getters often fill in variables in your JS. Once you get something from the DOM, you can use a variable to store it in memory for future manipulation.

```
const gaData = document.getElementById('ga');
```

Now that we have our element, **gaData**, we can access its properties:

```
gaData.style.color;
```

```
gaData.innerText;
```

```
gaData.classList;
```

We're mostly going to manipulate *classes* to make things happen on our pages.

 © 2020 General Assembly

Manipulating an Element's Classes

```
gaData.classList.toggle('show');
```

This statement takes the **ga** object, looks at the element's list of classes, and toggles the **show** class on or off.

 © 2020 General Assembly

Guided Walk-Through:
DOM Reference Code

Let's look at some DOM properties and methods in action! There's no need to feel overwhelmed by details — the **patterns** are what matter.

Reference code:

https://codepen.io/GAmarketing/pen/jOEVBvv?editors=1010

Front-End Web Development

Event Handling

Events and Listeners

Anytime a user interacts with a webpage, the browser classifies that action as an **event**.

In our JS code, we can listen for events in the browser and trigger functions in response using **event listeners**.

```
// When object is clicked, the action function is called
object.addEventListener('click', action)
```

Get, Then Listen

We'll often **get** an element and then **set** an event listener on it. Once the event occurs, the listener will execute the function it was given.

```
const ga = document.getElementById('ga');

function sayHello(){
    console.log("hello!");
}
ga.addEventListener('click', sayHello)
```

Discussion:

Events to Listen For

Click is the most common, but what other events might we want to listen for?

Guided Walk-Through:

Event Listeners With Selectors

Let's see how we can attach event listeners by grabbing elements with either class or ID selectors.

Code with class selectors:

https://codepen.io/GAmarketing/pen/vYEymXW

Code with ID selectors:

https://codepen.io/GAmarketing/pen/QWwGvmX

 © 2020 General Assembly

Partner Exercise:

Color Switcher

30 minutes

In this exercise, we'll use our new event-listening skills to create a color switcher!

Bonus: Try adding more switcher elements that change more colors.

Starter code:

https://codepen.io/GAmarketing/pen/dyPOWgd

Solution code:

https://codepen.io/GAmarketing/pen/xxbRdQr

 © 2020 General Assembly

Solo Exercise:

Traffic Light

45 minutes

In this exercise, write JavaScript that uses buttons to control the traffic light. Only ONE color should be active at a time!

Starter code:

https://codepen.io/GAmarketing/pen/qBEqmzp

Solution code:

https://codepen.io/GAmarketing/pen/RwNoVze

 © 2020 General Assembly

Key Takeaways

Everything Is JavaScript!

- HTML elements are represented as objects in the DOM.
- DOM objects have getters and setters to manipulate properties.
- Event listeners attach to objects and use functions to respond to user actions.

For Next Time

JavaScript Practice!

- We learned a lot of new concepts in this lesson. Next time, we'll put them into practice.


GA