You can think of a web page as consisting of three layers: structure, presentation, and behavior

- HTML is the structure layer
- CSS is the presentation layer
- JavaScript is the behavior layer

JavaScript is a programming language for creating interactivity and functionality in web browsers.
JavaScript

Background

JavaScript was introduced by Netscape in 1995

At that time, the Java language was ascendant and the name “JavaScript” was an attempt to ride this popularity.

Eventually, browsers other than Netscape began to support JavaScript functionality, calling it “ECMAScript”.

Today, JavaScript is not only a lingua franca of the Web but a basis for many other computational media projects.
As with CSS, JavaScript targets HTML elements to do something with them.

There are three ways you can apply JavaScript to HTML:

- Inline JavaScript
- Embedded JavaScript
- External JavaScript

External and embedded JavaScript are preferable for their separation of content and behavior.
Like HTML and CSS, JavaScript is rendered in the web browser.

Because it's rendered in the browser rather than on a server, JavaScript is considered a “front-end language”.

A browser’s “rendering engine” does the work of what you see on screen.

There are different rendering engines for different browsers as well as different JavaScript engines.
<table>
<thead>
<tr>
<th>Browser</th>
<th>Browser Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>Blink</td>
</tr>
<tr>
<td>Edge</td>
<td>Blink</td>
</tr>
<tr>
<td>Firefox</td>
<td>Gecko</td>
</tr>
<tr>
<td>Opera</td>
<td>Blink</td>
</tr>
<tr>
<td>Safari</td>
<td>Webkit</td>
</tr>
</tbody>
</table>
# JavaScript Engines

Executes JS Code

<table>
<thead>
<tr>
<th>Browser</th>
<th>JS Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>V8</td>
</tr>
<tr>
<td>Edge</td>
<td>Chakra</td>
</tr>
<tr>
<td>Firefox</td>
<td>Spidermonkey</td>
</tr>
<tr>
<td>Opera</td>
<td>V8</td>
</tr>
<tr>
<td>Safari</td>
<td>JavaScriptCore</td>
</tr>
</tbody>
</table>
Computationally speaking, there isn’t much JavaScript can’t do; it’s a robust programming language for web development.

We will use JavaScript and associated libraries for interactivity, animation, drawing on the HTML5 canvas, and rendering in 3D.

As with any technology, it’s good to consider when to—and not to—use it.
When a browser loads a web page, it creates a model of that page. This is called a “DOM tree” and it is stored in the browser’s memory. Every element, attribute, and piece of text in the HTML is represented by its own “DOM node”
Document Object Model
Types of DOM Nodes

There are four main types of nodes.

- The Document node, which represents the entire page
- Element nodes, which represent individual HTML tags
- Attribute nodes, which represent attributes of HTML tags, such as class
- Text nodes, which represents the text within an element, such as the content of a `p` tag

We talk about the relationship between `element` nodes as “parents,” “children,” and “siblings.”
Document Object Model

DOM Queries

JavaScript methods that find elements in the DOM tree are called “DOM queries”

DOM queries may return one element, or they may return a “node list”

Which DOM query you use depends on what you want to do and the scope of browser support required
Document Object Model

DOM Queries

JavaScript methods that return a single element node:

- `getElementById()`
- `querySelector()`
Document Object Model

DOM Queries

JavaScript methods that return one or more elements as a node list:

- `getElementsByClassName()`
- `getElementsByTagName()`
- `querySelectorAll()`
Drawing on the Web
CSCI-UA 380

JavaScript
Events
As you navigate the web, your browser registers different types of events. Common events include:
• Clicking or tapping on a link
• Hovering or swiping over an element
• Resizing the browser window
• A web page loading

JavaScript can be used to respond to the multitude of events that occur within the DOM.
Event Types
Event Types

UI Events

load
unload
error
resize
scroll
Event Types

Keyboard Events

keydown
keyup
keypress
Event Types

Mouse Events

click
dblclick
mousedown
mouseup
mousemove
mouseover
mouseout
Event Types
Focus Events

focus
blur
Event Types
Form Events

input
change
submit
reset
cut
copy
paste
select
Event Types

Mutation Events

DOMSubtreeModified
DOMNodeInserted
DOMNodeRemoved
DOMNodeInsertedIntoDocument
DOMNodeRemovedFromDocument
Event Types
Touch Events

touchstart
touchmove
touchend
touchcancel
Event Handling
Event Handling

1. Select an element for the script to respond to.

2. Specify which event will trigger the response.

3. Run code specific to that event.
Event Handling

Binding

Specifying which event will trigger the response is also known as binding.

There are three different ways to bind an event to an element.

• HTML event handlers
• DOM event handlers
• DOM event listeners