avionschool

Lesson 5.0 DOM (Document Object Model) - Introduction

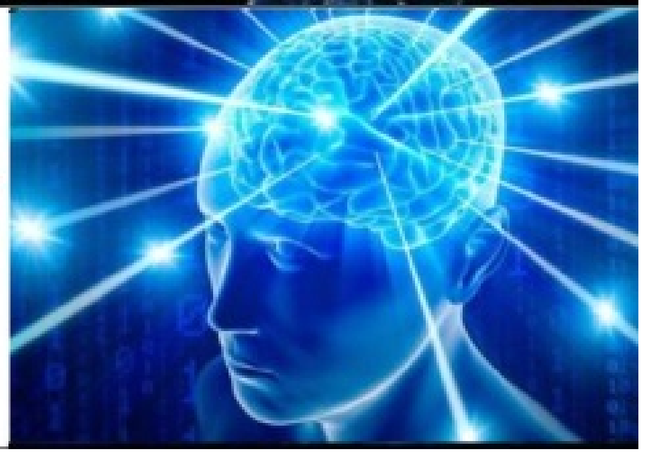
BATCH 4

NOVEMBER 24, 2020

SEMANTIC HTML
CSS SPECIFICITY

RESPONSIVENESS
FLEXBOX
GRID
MEDIA QUERY

WEB INTERACTIVITY
FORMS



A Brief Encounter: API

- Application Programming Interface
- defines interactions between multiple software intermediaries.
- It defines the kinds of calls or requests than can be made, how to make them, the data formats that should be used, the conventions to follow, etc.
- can be entirely custom, extendable, specific to a component or designed based on an industry-standard to ensure interoperability.

an example of a web API: DOM

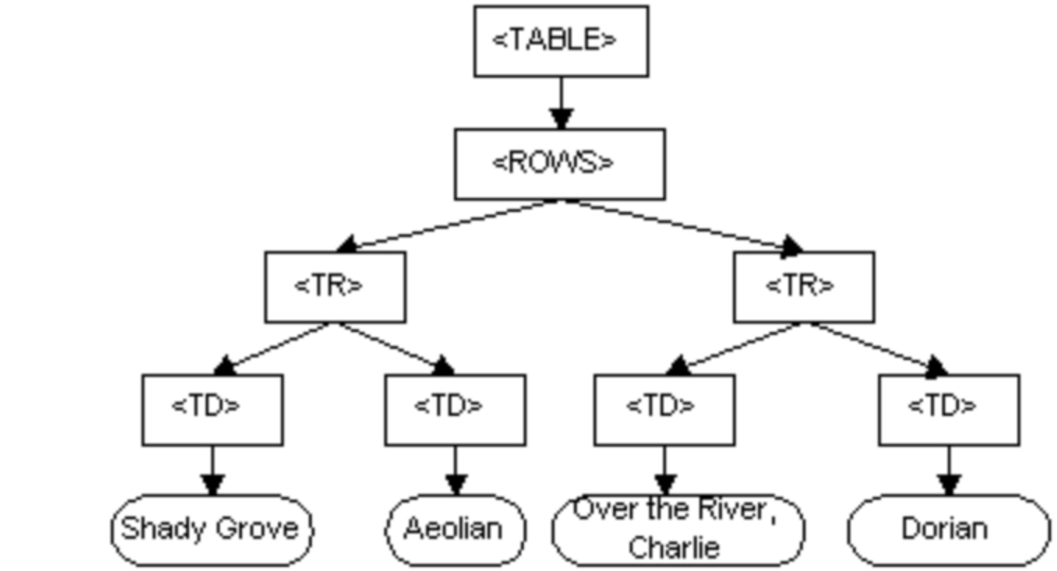
https://developer.mozilla.org/en-US/docs/Web/API

What is the DOM?

- The Document Object Model (DOM) is a programming interface for HTML and XML documents.
- It represents the page so that programs can change the document structure, style, and content.
- The DOM represents the document as nodes and objects. That way, programming languages can connect to the page.

API = DOM + JavaScript

```
<table>
  <rows>
    <tr>
      <td>Shady Grove</td>
      <td>Aeolian</td>
    </tr>
    <tr>
      <td>Over the River, Charlie</td>
      <td>Dorian</td>
    </tr>
  </rows>
</table>
```



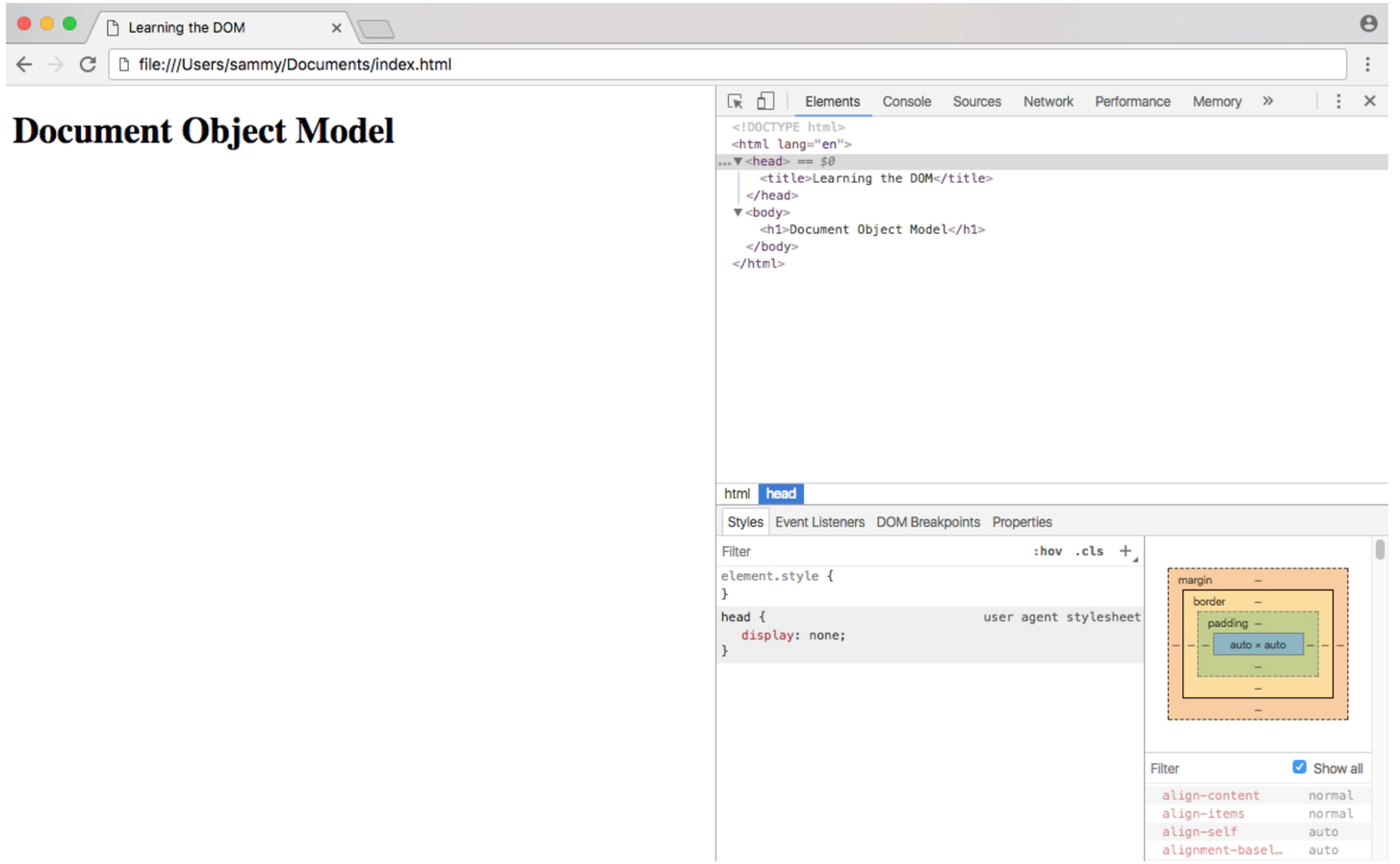
DOM representation of the example table

```html
<!DOCTYPE html>
<html lang="en">

  <head>
    <title>Learning the DOM</title>
  </head>

  <body>
    <h1>Document Object Model</h1>
  </body>

</html>
```



The Document Object

The `document` object is a built-in object that has many **properties** and **methods** that we can use to access and modify websites.

In Developer Tools on `index.html,` move to the *Console* tab. Type `document` into the console and press **ENTER**. You will see that what is output is the same as what you see in the *Elements* tab.

```
> document;
```

```
#document
<!DOCTYPE html>
<html lang="en">

  <head>
    <title>Learning the DOM</title>
  </head>

  <body>
    <h1>Document Object Model</h1>
  </body>

</html>
```

Typing `document` and otherwise working directly in the console is not something that you'll generally ever do outside of debugging, but it helps solidify exactly what the `document` object is and how to modify it, as we will discover below.

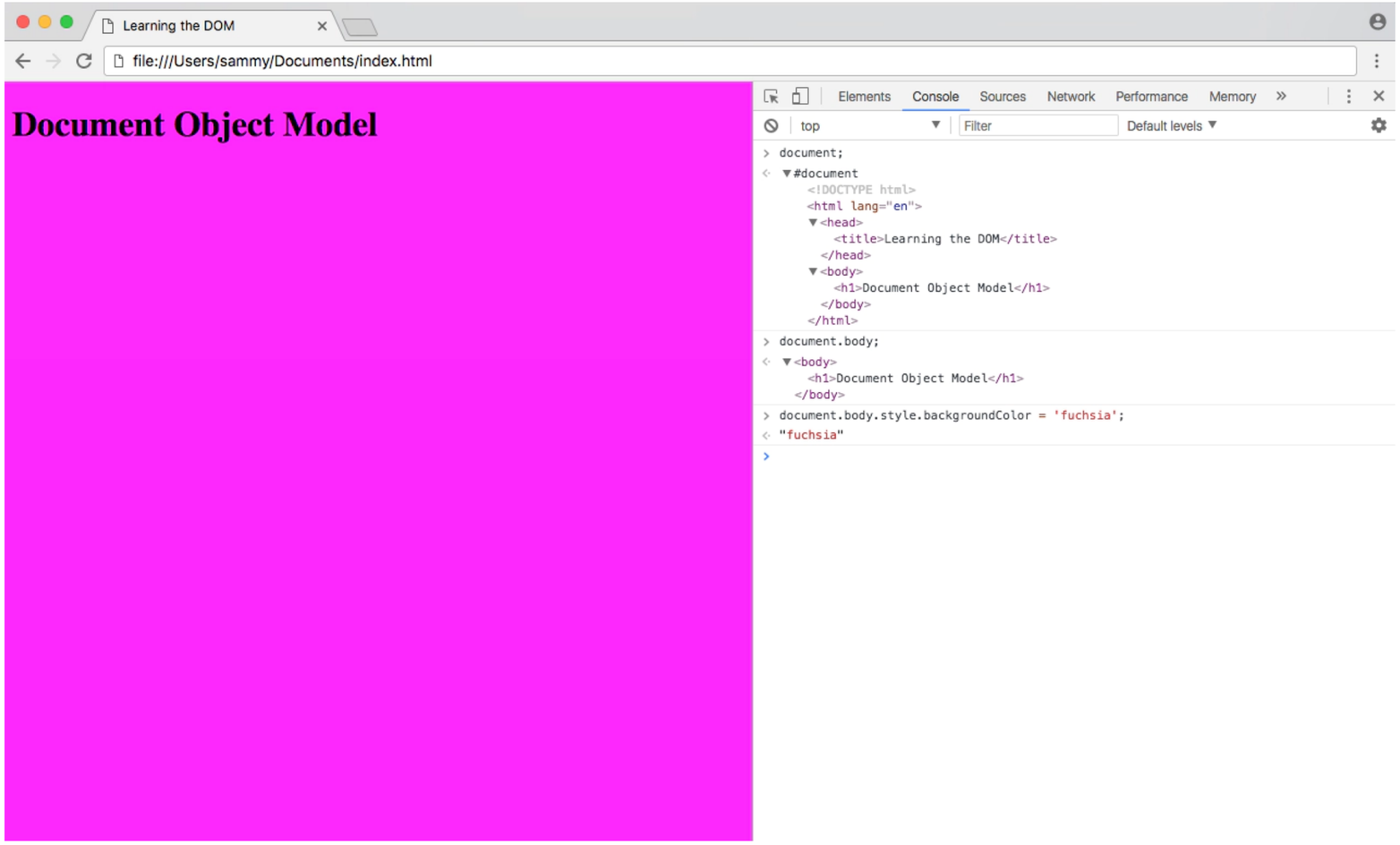
What is the Difference Between the DOM and HTML Source Code?

- The DOM is modified by client-side JavaScript.
- The browser automatically fixes errors in the source code

```
> document.body;
```

```
<body>
  <h1>Document Object Model</h1>
</body>
```

```
> document.body.style.backgroundColor = 'fuchsia';
```



```
> document.body;
```

```
<body style="background-color: fuchsia;">
  <h1>Document Object Model</h1>
</body>
```

Understanding the DOM Tree and Nodes

DOM object properties

- tagName
- className
- innerHTML (plain-text only!)
- outerHTML
- style - camelCase(backgroundColor), not param-case(background-color)
- img.src
- a.href
- input.value (select)
- input.checked (radio, checkbox)
- input.disabled
- input.required
- input.readOnly

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

Understanding the DOM Tree and Nodes

document Selectors

- document.querySelector(selector)
- document.querySelectorAll(name)
- document.createElement(name)
- document.getElementById(id)
- document.getElementsByClassName(className)
- document.getElementsByTagName(tagName)

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

Understanding the DOM Tree and Nodes

node Methods

- Node.appendChild(childNode)
- Node.removeChild(childNode)
- Node.replaceChild(childNode, newChildeNode)

https://developer.mozilla.org/en-US/docs/Web/API/Node

Understanding the DOM Tree, Nodes, Elements

element Properties and Methods

- Element.attributes
- Element.classList
- Element.className
- Element.id
- Element.innerHTML
- Element.outerHTML
- Element.tagName
- Element.querySelector()
- Element.querySelectorAll()
- Element.addEventListener()

https://developer.mozilla.org/en-US/docs/Web/API/Element