

## MODULE 3

# The DOM



# Back to HTML

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
  <link rel="stylesheet" href="style.css" type="text/css" />
  <script src="jquery-3.1.0.js"></script>
</head>
<body>
  <section class="content">

    <div id="box1" class="red box">
      <p class="text">Text inside a red box</p>
    </div>

    <div id="box2" class="blue box">
      <p class="text">Text inside a blue box</p>
    </div>

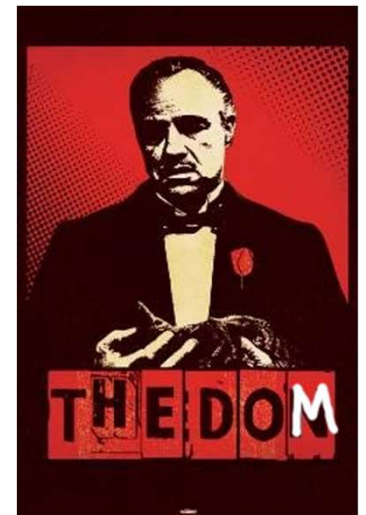
  </section>

  <script src="script.js"></script>
</body>
</html>
```

# Document Object Model (the DOM)

The **Document Object Model (DOM)** is the programming interface for HTML documents that are loaded into the browser

- It is **not** the page source
- It allows developers to:
  - look for an element with JavaScript
  - find an element's parents, siblings, children
  - add/remove css classes via JavaScript
  - add/remove elements from the page
  - Manipulate pretty much anything on the page

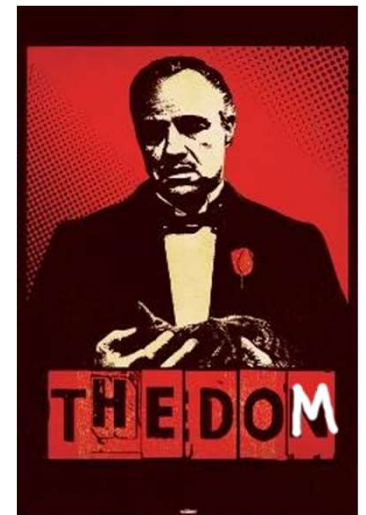


# HTML is Static

- HTML is read once and turned into the DOM
- CSS and JavaScript run against the DOM, not the HTML

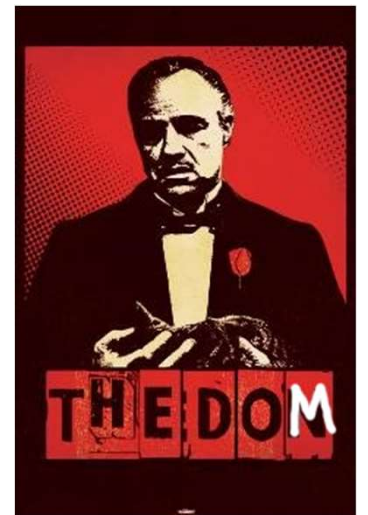
- What does this do?

```
table > tr > td {  
  background-color: red;  
}
```



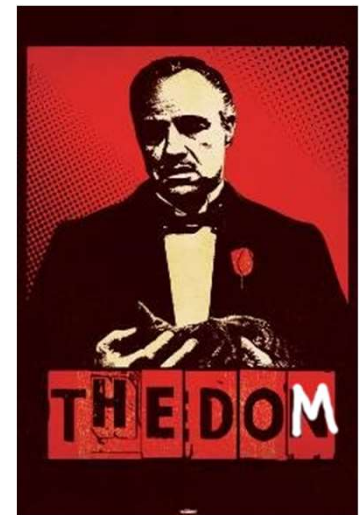
# Document Object Model (the DOM)

```
const element = document.querySelector("#main-content");  
element.innerText = "Hello World!";
```



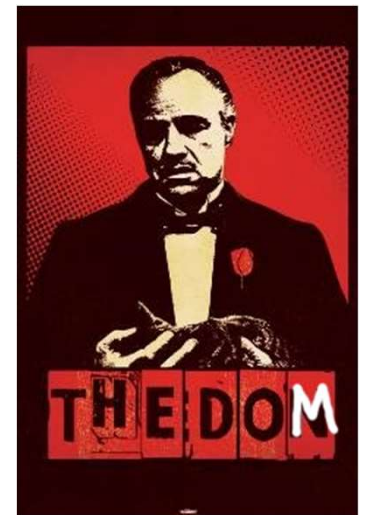
# DOM Selection Functions

- `getElementById()`
  - This function will get a single `HTMLElement` from the DOM and return a reference to it.
- `querySelector()`
  - Takes a standard CSS selector and returns the first element it finds that matches that selector
- `querySelectorAll()`
  - This will return a `NodeList` of all the elements, which you can use as an array



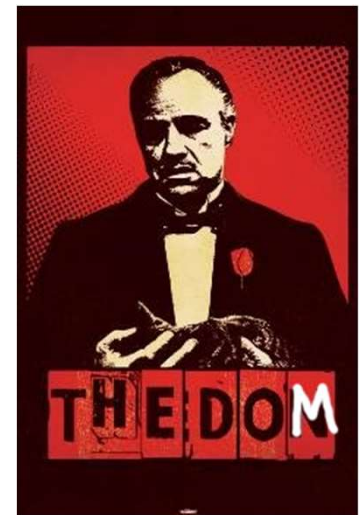
# Changing Elements

- `innerText`
  - Updates any text information on the page
  - All text (including html tags) are replaced!
  - Insert text treated as literals: no interpreting of HTML
- `innerHTML`
  - Updates any text information on the page
  - All text (including html tags) are replaced!
  - Interprets of HTML for display
  - Do not use with user input! (Why)



# Getting and Setting Values

- `.value` will return the value of a selected element
  - `let todoInput = document.querySelector('input[name=newTodo]');`
  - `let newTodoText = todoInput.value;`
- Also used to set a value
  - `todoInput.value = '';`
- Radio and checkboxes use `.checked`
  - `let finishedCheckbox = document.querySelector('input[name=isFinished]');`
  - `let isFinished = finishedCheckbox.checked;`





# Manipulating Classes

- `classList` accesses the classes applied to an element

```
// Get the first line item
```

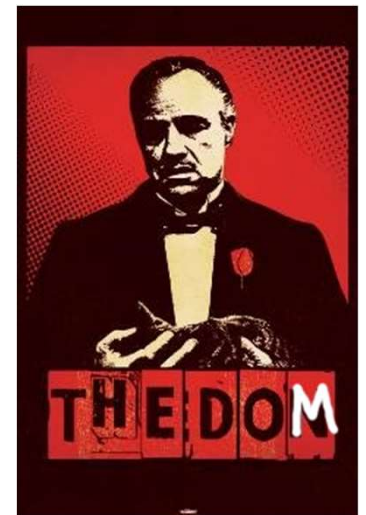
```
let firstListItem = document.querySelector('#todos li');
```

```
// Add the class `done`
```

```
firstListItem.classList.add('done');
```

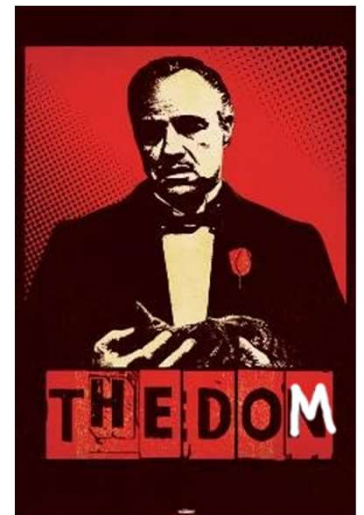
```
// Remove the class `priority`
```

```
firstListItem.classList.remove('priority');
```



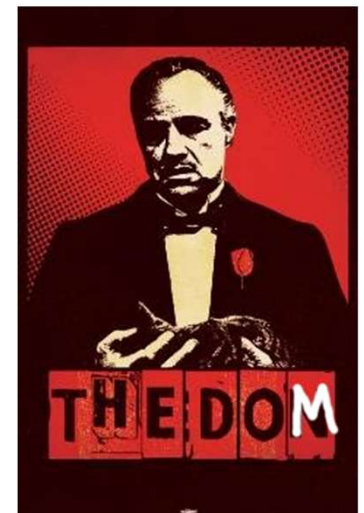
# Adding to the Mob....DOM!

- createElement()
  - Creates a new DOM element and returns it
  - Element is not on the page, but you can manipulate it.
- insertAdjacentElement()
  - Adds the element as the last child of the selected element



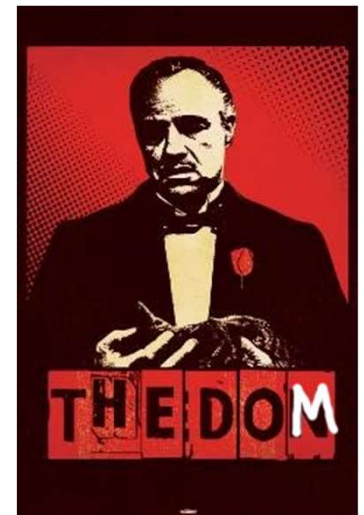
# Placing a New Element

Location	Meaning
beforebegin	Put the element before this one
afterbegin	Put the element inside this one at the top
beforeend	Put the element inside this one at the bottom
afterend	Put the element after this one



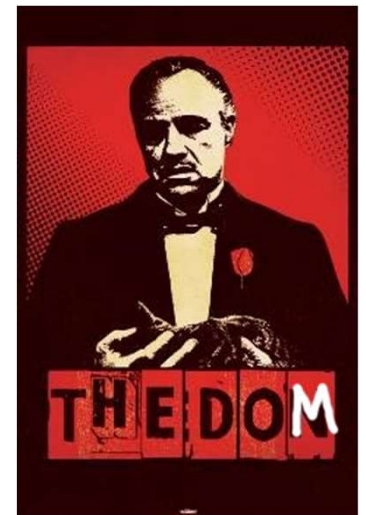
# Traversing the DOM

- `.children` will select the immediate child elements
- Convert to an array to manipulate
  - `map`, `filter`, `reduce`, etc.
  - `let todoItemsArray = Array.from(todoList.children);`
- `.childNodes` will get all nodes inside an element
- `children` vs `childNodes`
  - `children` are elements, only contain HTML not text
  - `childNodes` contain HTML and text/values



# Traversing the DOM

- `.parentNode`
  - Gets the immediate parent node of the element
- `nextElementSibling`
- `previousElementSibling`
- `removeChild`
  - Removes the child element from the DOM



# LET'S CODE!



ELEVATE  YOURSELF

WHAT QUESTIONS DO  
YOU HAVE?



# Reading for tonight:

## **Event Handling**

