

12. DOM and Event Listeners

Table of Content

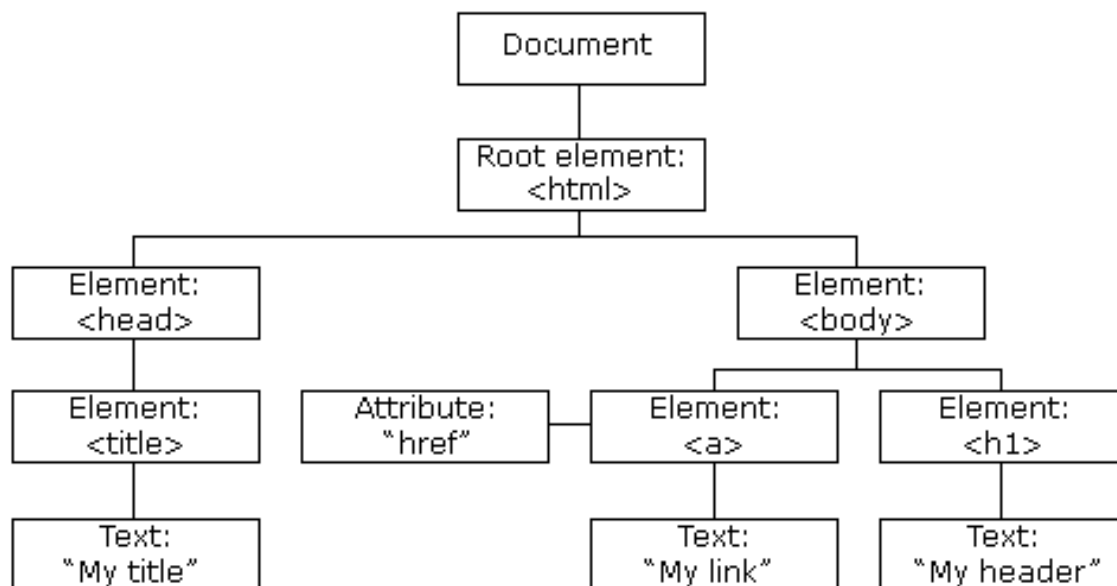
1. The HTML DOM
2. Event Listeners in Javascript

1. The HTML DOM (Document Object Model)

▼ Introduction

When a web page is loaded, the browser creates a **Document Object Model** of the page.

The **HTML DOM** model is constructed as a tree of **Objects**:



With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

The HTML DOM document object is the owner of all other objects in your web page. The document object represents your web page. If you want to access any element in an HTML page, you always start with accessing the document object.

▼ Finding the HTML Elements

- Finding HTML elements **by id**

This example finds the element with `id="intro"`

```
const element = document.getElementById("intro");
```

- Finding HTML elements **by tag name**

This example finds all `<p>` elements:

```
const element = document.getElementsByTagName("p");
```

- Finding HTML elements **by class name**

This example returns a list of all elements with `class="intro"`.

```
const x = document.getElementsByClassName("intro");
```

- Finding HTML elements **by CSS selectors**

If you want to find all HTML elements that match a specified CSS selector (id, class names, types, attributes, values of attributes, etc), use the

`querySelectorAll()` method.

This example returns a list of all `<p>` elements with `class="intro"`.

```
const x = document.querySelectorAll("p.intro");
```

▼ Changing HTML

- changing HTML Content using `innerHTML`

To change the content of an HTML element, use this syntax:

```
document.getElementById(id).innerHTML = new HTML
```

- changing HTML attributes value

To change the value of an HTML attribute, use this syntax:

```
document.getElementById( id ). attribute = new value
```

This example changes the value of the src attribute of an `` element:

```
document.getElementById("myImage").src = "landscape.jpg";
```

- adding a class in HTML

using the `classList` property of Elements, we can **add**, **remove** or **toggle** classes from HTML

```
const element = document.getElementById("myDIV");
element.classList.add("mystyle");
element.classList.remove("mystyle");
element.classList.toggle("mystyle");
```

- creating and appending new HTML

We can create a new Element using the `createElement()` method of the document. We can append this new Element (node) inside any other element.

```
<div id="div1">
  <p id="p1">This is a paragraph.</p>
  <p id="p2">This is another paragraph.</p>
</div>

<script>
  const para = document.createElement("p");
  const node = document.createTextNode("This is new.");
  para.appendChild(node);

  const element = document.getElementById("div1");
  element.appendChild(para);
</script>
```

▼ Changing HTML Style

To change the style of an HTML element, use this syntax:

```
document.getElementById(id).style.property = new style
document.getElementById("p2").style.color = "blue";
```

2. Event Listeners in Javascript

▼ What is an Event?

Events allow you to write JavaScript code that reacts to certain situations. Examples of events include:

- The user clicks the mouse button
- The Web page loading
- A form field being changed

▼ How to use Event Listeners?

As the name suggests, the event listener listens for events and gets triggered when an event occurs.

```
addEventListener(type, listener, useCapture)

const button = document.querySelector(".btn")

button.addEventListener("click", () => {
  console.log("Button clicked.");
})
```

The `addEventListener()` method attaches an event handler to an element without overwriting existing event handlers.

You can add many event handlers to one element.

You can add many event handlers of the same type to one element, i.e two "click" events.

A complete list of all HTML DOM Events:

https://www.w3schools.com/jsref/dom_obj_event.asp

Assignments

1. Build a NavBar that changes the background color after the page is scrolled 100px
2. Build a Toggle NavBar Menu with a Hamburger menu like this one (https://www.w3schools.com/howto/howto_js_mobile_navbar.asp)
3. Build a working calculator Application.
4. Build an Image Slider With Two Buttons for the Left and Right Slide like this one (https://www.w3schools.com/howto/howto_js_slideshow.asp)