JS

DOM Manipulation

Outline

- Introduction to HTML DOM
- DOM Methods
- DOM Document
- DOM Elements

AkiraChix
SHE BUILDS • SHE SERVES • SHE LEADS

Introduction to HTML DOM

What is the HTML DOM?

When a web page is loaded, the browser creates a Document Object Model of the page. The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

A programming language can be used to access and modify this object model, and this action is called DOM manipulation. And we will do that with JavaScript because JS is awesome!

DOM tree

AkiraChix
SHE BUILDS • SHE SERVES • SHE LEADS

The HTML DOM model is constructed as a tree of Objects

The backbone of an HTML document are tags e.g. head, body, p

Every HTML-tag is an object. Nested tags are called “children” of the enclosing one.

The text inside a tag it is an object as well.

All these objects are accessible using JavaScript.

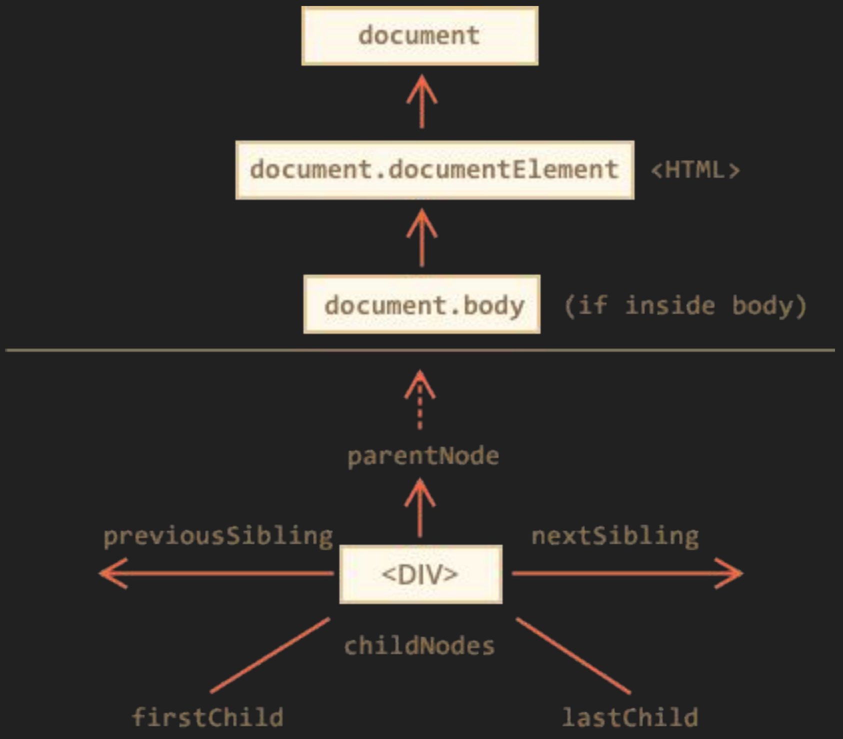
An example of DOM

```
<!DOCTYPE HTML>
<html>
<head>
  <title></title>
</head>
<body>
</body>
</html>
```

To explore the DOM you can use the browser developer tools. Actually, that’s what we use when developing.

DOM nodes have properties and methods that allow to:

- Change all the HTML elements in the page
- Change all the HTML attributes in the page
- Change all the CSS styles in the page
- Remove existing HTML elements and attributes
- Add new HTML elements and attributes
- React to all existing HTML events in the page
- Create new HTML events in the page


document
document.documentElement
<HTML>
document.body
(if inside body)
parentNode
previousSibling
nextSibling
<DIV>
childNodes
firstChild
lastChild

Traversing the DOM

The topmost tree nodes are available directly as document properties:

<html> = document.documentElement
The topmost document node is document.documentElement. That’s DOM node of <html> tag.
<body> = document.body
Another widely used DOM node is the <body> element – document.body.
<head> = document.head
The <head> tag is available as document.head.

What if we want to access an element next to an element that we have already accessed, or access the parent node of a previously accessed element? The properties firstChild, lastChild, parentNode, nextSibling, and previousSibling can get this job done for us.

```
document.getElementById("myDiv").firstChild;
document.getElementById("myDiv").lastChild;
document.getElementById("myDiv").parentNode;
document.getElementById("myDiv").nextSibling;
document.getElementById("myDiv").previousSibling;
```

DOM Properties and Methods

In the DOM, all HTML elements are defined as objects.

The programming interface is the properties and methods of each object.

A property is a value that you can get or set (like changing the content of an HTML element).

A method is an action you can do (like add or deleting an HTML element).

The following example changes the content (the innerHTML) of the <p> element with id="demo":

```
<html>
    <body>
    <p id="demo"></p>
    <script>
        document.getElementById("demo").innerHTML = "Hello World!";
    </script>
    </body>
</html>
```

In the example above, getElementById is a method, while innerHTML is a property.

The getElementById method used id="demo" to find the element.

The innerHTML property is useful for getting or replacing the content of HTML elements.

DOM Document

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.

Below are some examples of how you can use the document object to access and manipulate HTML.

Finding HTML Elements

- document.getElementById(id) Find an element by element id
- document.getElementsByTagName(name) Find elements by tag name
- document.getElementsByClassName(name) Find elements by class name

Changing HTML Elements

- Change the innerHTML of an element
  - element.innerHTML = new html content
- Change the attribute value of an HTML element
  - element.attribute = new value
- Change the style of an HTML element
  - element.style.property = new style
- Change the attribute value of an HTML element
  - element.setAttribute(attribute, value)

Adding and Deleting Elements

- Create an HTML element
  - document.createElement(element)
- Remove an HTML element
  - document.removeChild(element)
- Add an HTML element
  - document.appendChild(element)
- Replace an HTML element
  - document.replaceChild(new, old)
- Write into the HTML output stream
  - document.write(text)

Adding Events Handlers

- Adding event handler code to an onclick event
  - document.getElementById(id).onclick = function(){code}

DOM Elements

Often, with JavaScript, you want to manipulate HTML elements. To do so, you have to find the elements first. There are several ways to do this:

- Finding HTML elements by id
    - var myElement = document.getElementById("intro");
- Finding HTML elements by tag name
    - var x = document.getElementsByTagName("p");
- Finding HTML elements by class name
    - var x = document.getElementsByClassName("intro");
- Finding HTML elements by CSS selectors
    - var x = document.querySelectorAll("p.intro");

Now we know how to handle elements, so let’s learn how to handle the attributes of these elements.

- getAttribute as its name may suggest, it is used to get an attribute. Like the class name, the id name, the href of a link or any other HTML attribute.
- setAttribute is used to set a new attribute to an element. It takes two arguments, first the attribute and second the name of the attribute.
- hasAttribute used to check if an attribute exists, takes an attribute as an argument.
- removeAttribute used to remove an attribute, it takes an attribute as an argument.

- id this property is used to set or get the id of an element.
- className is used to set or get the class of an element.

Assignment

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Greens Kiosk</title>
</head>
<body>
  <h1 id="title">Welcome to Greens Kiosk</h1>
  <p>We sell fruits and vegetables</p>
  <h3>Fruits</h3>
  <ul id="fruList">
    <li>Mangoes</li>
    <li>Bananas</li>
    <li>Water Melons</li>
  </ul>
  <h3>Vegetables</h3>
  <ul id="vegList">
    <li>Onions</li>
    <li>Tomatoes</li>
    <li>Kales</li>
  </ul>
</body>
</html>
```

Questions:

1. Change document background color to silver
2. Change the font color for h1 title tag to green
3. Change the font case for h3 title tags to uppercase
4. Add one more fruit to the fruits list
5. Add one more vegetable to the vegetables list

Next class

- DOM Animations
- DOM Events
- DOM Event Listener
- DOM Navigation
- DOM Nodes
- DOM Collections
- DOM Node Lists

Video of the week - https://www.youtube.com/watch?v=wiozYyXQEVk