JavaScript

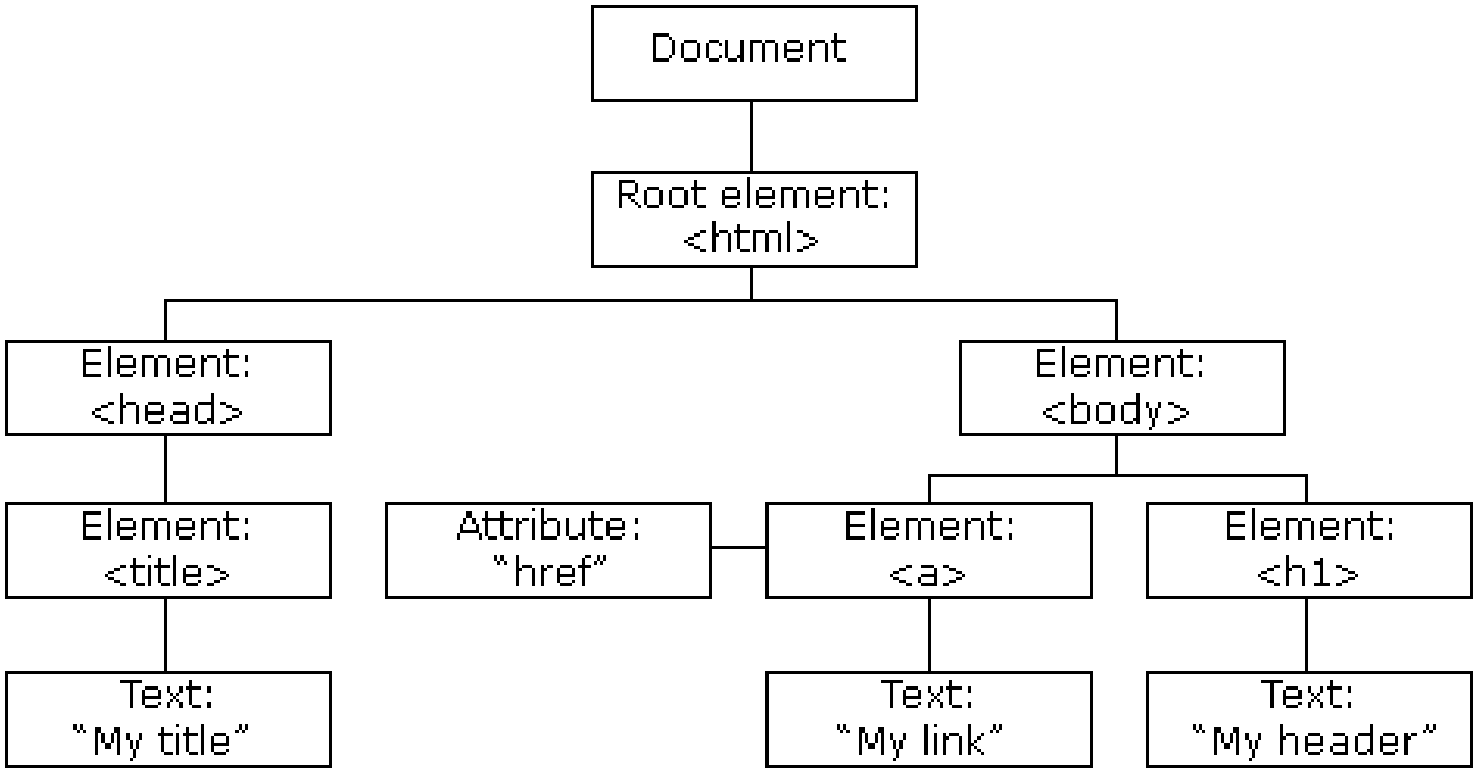
- With the HTML DOM,

  JS can access and change all the elements of an HTML document.

HTML DOM (Document Object Model)

- When a web page is loaded,
  the browser creates a **D**ocument **O**bject **M**odel of the page.

The HTML DOM Tree of Objects



- With the object model, JS (JavaScript) gets all the power it needs to create dynamic HTML.

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

What is the DOM?

- The DOM is a W3C standard.

(World Wide Web Consortium)

- The DOM defines a standard for accessing documents:

"The W3C Document Object Model (DOM) is a platform that allows programs to dynamically access and update the content, structure, and style of a document."

What is the HTML DOM?

- The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

HTML DOM Methods

- HTML DOM methods are **actions** you can perform on HTML Elements.

- HTML DOM properties are **values** (of HTML Elements) that you can set or change.

Example

```
<p id="p1"></p>

<script>
    document.getElementById("p1").innerHTML =
    "Hello from JavaScript. ";
</script>
```

Note : getElementById is a **method**,
innerHTML is a **property**.

Output

- getElementById method used to find the element.
- innerHTML property can be used to get or change any HTML element.

HTML 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.

Example

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

Property	Description
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element
Method	Description
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

Adding and Deleting Elements

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

Finding HTML Elements

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by CSS selectors
- Finding HTML elements by HTML object collections

Finding HTML Element by Id

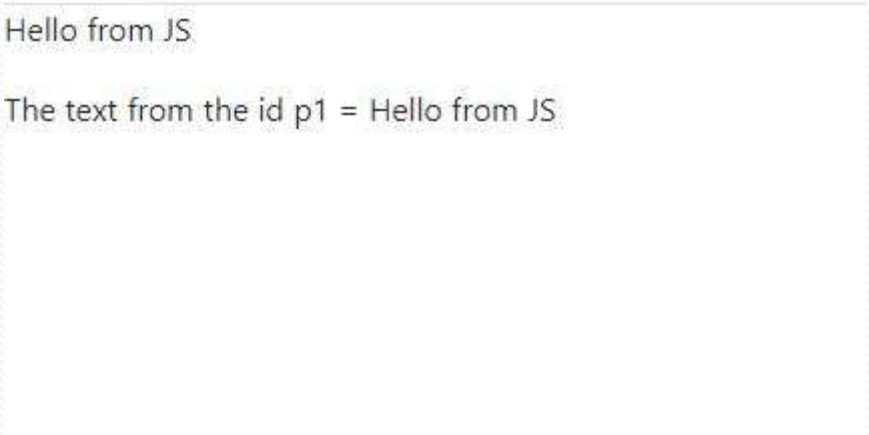
```
<p id="p1">Hello from JS</p>

<p id="p2"></p>

<script>
    var myElement = document.getElementById("p1");

    document.getElementById("p2").innerHTML =
            "The text from the id p1 = " + myElement.innerHTML;
</script>
```

Output



****