SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

Course Code and Name : 19TS601 FULL STACK DEVELOPMENT

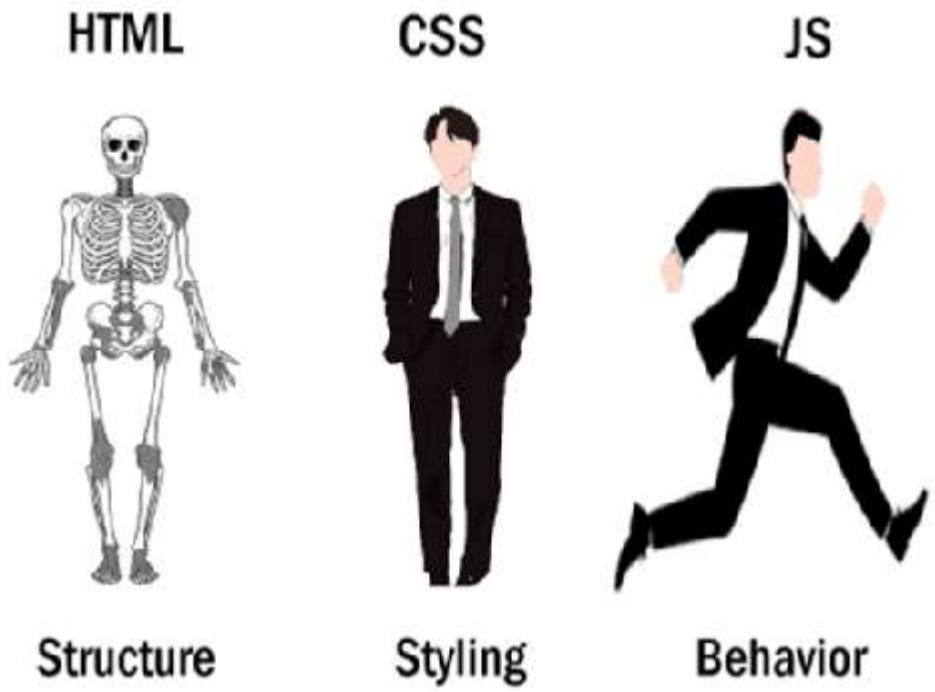
Unit 1 : JAVASCRIPT AND BASICS OF MERN STACK

Topic : JavaScript DOM

JavaScript DOM

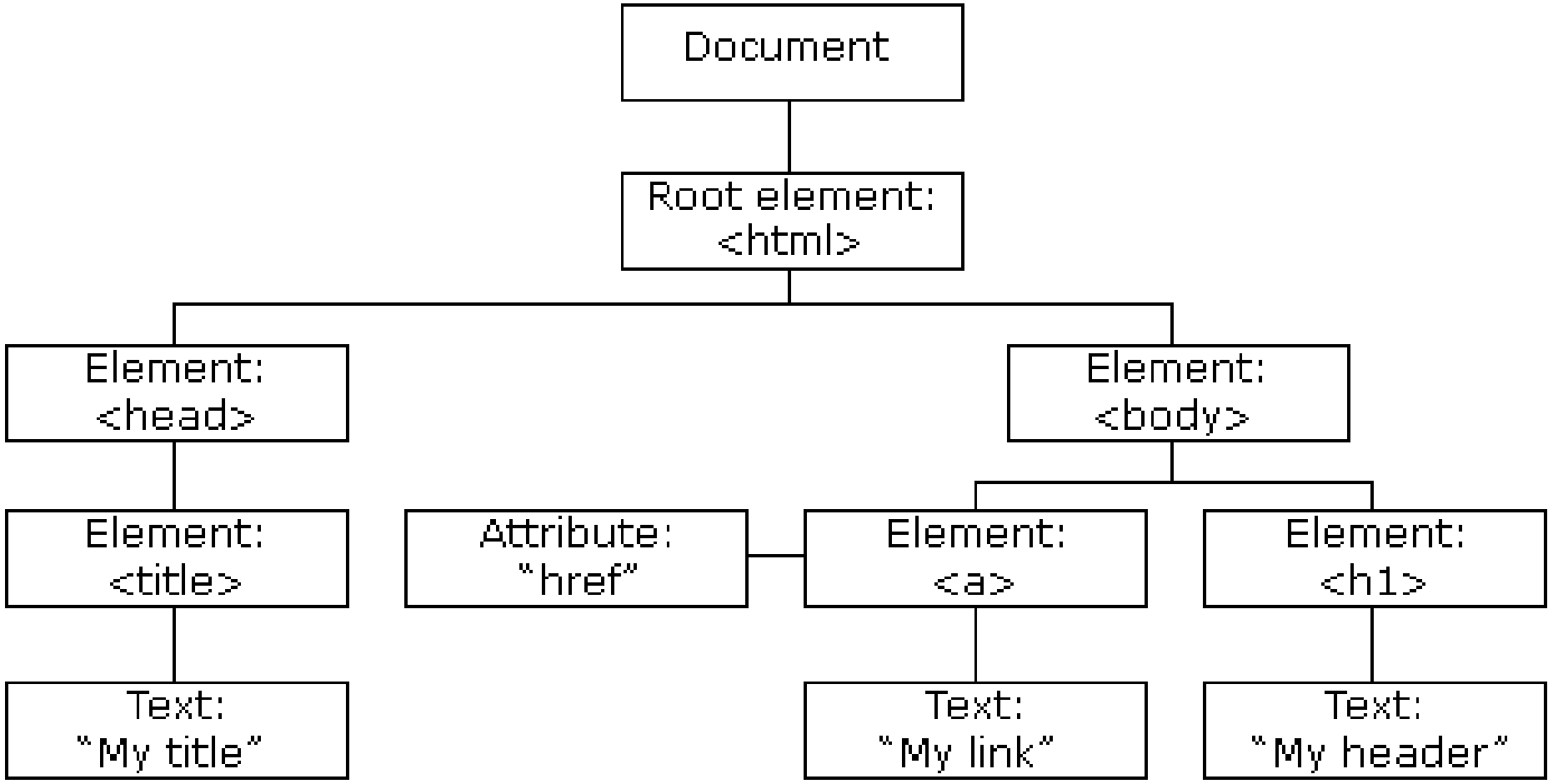
- The DOM stands for Document Object Model.
- It’s Bridge between web page and JavaScript.
- The DOM allows JavaScript to “talk” to a web page and control its content, appearance, and behavior.
- For example
- a toy house. The DOM is like a set of instructions that tells you how to take apart and rebuild the house. You can add windows, change the color of walls, move rooms around, and even change furniture—all with the DOM and JavaScript.

The relationship between HTML, CSS, and JavaScript



HTML (Structure)	**CSS** (Style)	**JavaScript** (Behavior)
HTML: HyperText Markup Language Defines the content and structure of the webpage using elements like headings, paragraphs, images, and links.	**CSS: Cascading Style Sheets** It controls the appearance and layout of HTML elements, adjusting properties like color, size, positioning, and fonts.	**JavaScript** Adds interactivity and dynamic functionality to the webpage by responding to user actions (like clicks or key presses) and changing content or behavior.

The HTML DOM Tree of Objects



- The document is at the top—this is the entire webpage.
- HTML is the first big branch—this is the structure of your webpage.
- Inside HTML, you have smaller branches like <head> and <body>. Inside <body>, you have branches like <Title>, <h1>, <script>, and .
- These branches can have more branches too, like text nodes or images.
- The DOM tree organizes everything on your webpage from top to bottom, with each part being an object or node in the tree.

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

DOM Important

The DOM is like a magical tool that helps JavaScript to:

- **Change the content:** JavaScript to change the text or images on a webpage without reloading the whole page. For example, you can show a “Thank You” message after a user clicks a button.
- **Respond to user actions:** The DOM allows JavaScript to react to what users do on the page. For example, if you click a button or typ something in a form, JavaScript can use the DOM to update the pag
- **Control forms:** If a user fills out a form, JavaScript can check the data, make changes, or even submit the data to a server using the DOM.

The DOM Work

- When you open a browser webpage, the browser looks at the HTML code of that page and turns it into an internal structure called the DOM tree.
- This tree is a map of all the elements on the page (like buttons, paragraphs, images, etc.), and you can use JavaScript to modify or interact with this structure.
- Every part of the webpage (like a <p>, , or <img>) is a node in the DOM tree.
  The root of the tree is the Document object, and each branch leads to other elements like text, images, buttons, and links.

document.write() Method

- The document.write() method is a built-in method in JavaScript that allows you to write a string of text directly to the HTML document.
- It is commonly used for testing or debugging, but it’s generally not recommended for use in production websites.
- It can overwrite the entire document if called after the page has loaded, potentially causing unexpected behavior.

For example (index.html)

```
<!DOCTYPE html>
<html>
 <head>
  <title>Document Write Example</title>
 </head>
<body>
 <script>
  document.write(“<h1>Hello, World!</h1>”);
 </script>
</body>
</html>
```

Methods of document object

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

Accessing field value by document object

- to get the value of input text by user. Here, we are using **document.form1.name.value** to get the value of name field.
- Here, **document** is the root element that represents the html document.

form1 is the name of the form.

- **name** is the attribute name of the input text.
- **value** is the property, that returns the value of the input text.

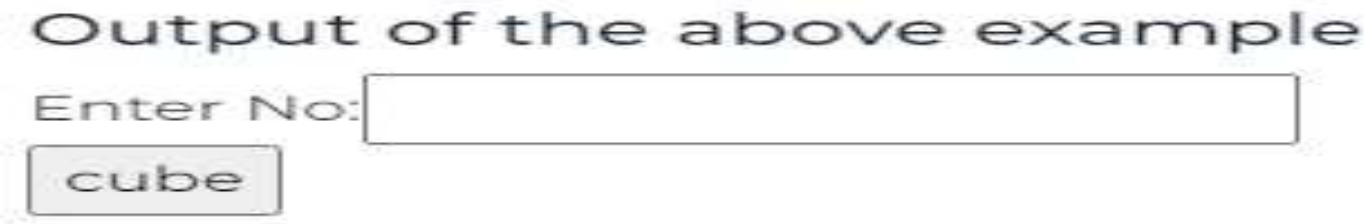
```
<script type="text/javascript">
function printvalue(){
var name=document.form1.name.value;
alert("Welcome: "+name);
}
</script>
 <form name="form1"> Enter Name:<input type="text" name="name"/>
<input type="button" onclick="printvalue()" value="print name"/>
</form>
```

Output of the above example

- **document.form1.name.value** to get the value of the input value.
- Instead of this, we can use document.getElementById() method to get value of the input text. But we need to define id for the input field.

```
<script type="text/javascript">
function getcube(){
var number=document.getElementById("number").value; alert(number*
   number*number);
}
</script>
<form>
Enter No:<input type="text" id="number" name="number"/><br/>
<input type="button" value="cube" onclick="getcube()"/>
</form>
```

Output of the above example



1.What is DOM?

Text Book:

1.Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, A Press Publisher, 2019.

Reference:

David Flanagan, “Java Script: The Definitive Guide”, O’Reilly Media, Inc, 7 th Edition, 2020

2. Matt Frisbie, “Professional JavaScript for Web Developers” Wiley Publishing, Inc, 4th Edition, ISBN: 978-1-119-36656-0, 2019


Thank
You!