**What is DOM?**

*The Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.*

**What is the DOM in HTML?**

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content.

**Why is HTML DOM important?**

It provides a way to access, manipulate, and interact with the structure and content of an HTML document.

**Which is the type of HTML DOM?**

The HTML DOM model is constructed as a tree of Objects. In the HTML DOM everything is a node. The document itself is a document node. All HTML elements are element nodes.

Example: Imagine you're looking at a webpage, and you want to break it down into smaller parts so you can understand and work with it better. You can think of each part of the webpage, like paragraphs, images, or even just text, as a "node."

In simple terms, everything is a node in the HTML DOM because it helps organize and make sense of the different parts of a webpage.

The DOM representation of this HTML document looks something like this:

Document

├── html (Element Node)

│ ├── head (Element Node)

│ │ └── title (Element Node)

│ │ └── "Example" (Text Node)

│ └── body (Element Node)

│ └── p (Element Node)

│ └── "Hello, world!" (Text Node)

The root node is the Document node.

The <html>, <head>, <title>, <body>, and <p> elements are element nodes.

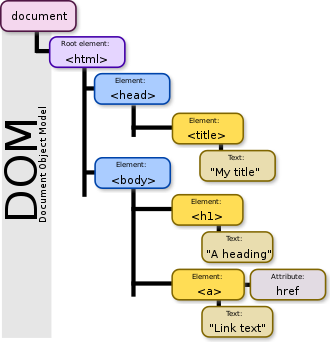
The text "Example" and "Hello, world!" are text nodes.

By treating everything as nodes, the DOM provides a structured way to interact with and manipulate the HTML document programmatically.

**When DOM is Created ?**

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.

**HTML DOM TREE** 

**How to Target DOM ?**

1. ID -> document.getElementById(id)
2. Class Name -> document.getElementsByClassName(name)
3. Tag Name -> document.getElementsByTagName(name)

**Other DOM Target method**

Document.links, document.all, document, document.head, document.title, document.body, document.images, document.anchors etc

-

var data = document.body

console.log(data);

**getElementsByClassName**

<script>

var data = document.getElementsByClassName("strero")

for (let i = 0; i < data.length; i++) {

console.log(data.item(i))

data.item(i).style.background="blue";

}

</script>

**getElementById**

var data = document.getElementById("stu\_form")

console.log(data.innerText);

**Change the element:**

var data = document.getElementById("san")

data.innerText = " anurag "

**Added tag:**

var data = document.getElementById("san")

data.innerHTML= "<h1> anurag </h1>"

**Change the Image:**

let el = document.getElementById("image").src= "dow.jpg"

**Add new class:**

data.classList.add("stan");

**Add two new class:**

data.classList.add("stan", "\_line");

**Remove class:**

data.classList.remove("page");

**Underline added:**

data.style.textDecoration= "underline";

**Color added:**

data.style.color="green"

**Excess parentElement:**

data.parentElement.style.background = "yellow"

**Set attribute:**

data.setAttribute("src", "img")

**Remove all css:**

data.remove()

console.log(data.parentElement);

**Adding New Element:**

let addNew = document.createElement("p");

addNew.innerText="Hello i am new paragraph";

let addElement = document.getElementById("header");

addElement.appendChild(addNew);

**getElementsByTagName**

var element = document.getElementsByTagName("div")

console.log(element);

**New DOM Target Method**

**querySelector -> document.querySelector(CSS Selector)**

**querySelectorAll -> document.querySelectorAll(CSS Selector)**

let selector = document.querySelector(".container .then")

console.log(selector);

let selector = document.querySelectorAll(".container .then")

console.log(selector);

**querySelector**

let element = document.querySelector("#next\_heading")

element.innerHTML="<h1> Hello </h1>"

element = document.querySelector("next\_heading").getAttribute("class")

console.log(element)

**Show classes**

element = document.querySelector(".list")

console.log(element);

**Using ClassName**

element = document.getElementsByClassName("list")

console.log(element);

**Using querySelectorAll**

element = document.querySelectorAll(".list")

console.log(element)

**Get Innertext**

element = document.querySelectorAll(".list")[0].innerHTML

console.log(element)

**Access Tag**

element = document.querySelector("#header li")

console.log(element)

element = document.querySelectorAll("#header li")

console.log(element)

element = document.querySelectorAll("#header li")[1].innerHTML

console.log(element)

**DOM CSS Styling methods**

1. Style
2. ClassName
3. classList

element= document.querySelector("#san").style.color

console.log(element)

element = document.querySelector("#header").style.backgroundColor="yellow"

element = document.querySelector("#header").style.color="blue"

console.log(element)

**By using className**

let element = document.querySelector("#para").className="abc"

document.querySelector("#para").className

console.log(element)

**Double Class** but priority will be given to last class

let element = document.querySelector("#para").className="abc xyz"

document.querySelector("#para").className

console.log(element)

**classList**

let element = document.querySelector("#para").classList="abc xyz"

element = document.querySelector("#para").classList

console.log(element)

Note: classList will return an array in console as DOMTokenList

**Add Multiple Classes**

let element = document.querySelector("#para").classList.add("list","and")

element = document.querySelector("#para").classList

console.log(element)

**Remove any Class**

let element = document.querySelector("#para").classList.remove("list")

element = document.querySelector("#para").classList

console.log(element)

# **DOM Get & Set Value Methods**

**What we can Get with DOM ?**

* HTML
* Text
* Attribute

**DOM Get Method**

* InnerHTML
* innerText
* getAttribute
* getAttributeNode
* Attributes

**innerHTML**

let element = document.getElementById("stu\_form").innerHTML

console.log(element)

**innerText**

let element = document.getElementById("stu\_form").innerText

console.log(element)

**getAttribute**

let element = document.getElementById("stu\_form").getAttribute("class")

console.log(element)

let element = document.getElementById("stu\_form").getAttribute("onsubmit")

console.log(element)

**getAttributeNode**

let element = document.getElementById("stu\_form").getAttributeNode("style")

console.log(element)

let element = document.getElementById("stu\_form").getAttributeNode("style").value

console.log(element)

**attributes**

let element = document.getElementById("stu\_form").attributes

console.log(element)

This will return all values in object.

**Access attribute**

let element = document.getElementById("stu\_form").attributes[1]

console.log(element)

**Access attribute value**

let element = document.getElementById("stu\_form").attributes[1].value

console.log(element)

**Access attribute name**

let element = document.getElementById("stu\_form").attributes[1].name

console.log(element)

**DOM Set Method**

* innerHtml
* innerText
* setAttribute
* attribute
* removeAttribute

**innerText**

let element = document.getElementById("stu\_form").innerText="Hiiii"

console.log(element)

let element = document.getElementById("stu\_form").innerText=

"<h1> Hiiii </h1>"

console.log(element)

**innerHTML**

let element = document.getElementById("stu\_form").innerHTML=

"<h1> Hiiii </h1>"

console.log(element)

**setAttribute**

element = document.getElementById("stu\_form").setAttribute("class","and")

element= document.getElementById("stu\_form").getAttribute("class")

console.log(element)

element = document.getElementById("stu\_form").setAttribute("style","border:2px solid green")

element= document.getElementById("stu\_form").getAttribute("style")

console.log(element)

**What is an Event ?**

Event allows you to write javascript code that reacts to certain situations. Examples of event includes:

* User clicks the mouse button
* Web page loading
* A form field being changed

There are two ways to handle events in javascript:

* By using an event handler
* By using an event listener

**Event Handler**

**Javascript provides various kinds of event handlers that get triggered based on specific actions on html elements. Few of the event handlers are:**

* **Onclick**
* **Onload**
* **Onmouseover**
* **Onmouseout**
* **Onkeypress**
* **Onkeydown**
* **Onkeyup**

**const handelButton1=(()=>{**

**console.log("Button number 1 clicked");**

**});**

**const handelMouse=(()=>{**

**console.log("i am mouse over on 2nd button");**

**});**

**const handelMousemove=(()=>{**

**console.log("Mouse is moved from 3rd button");**

**});**

**const handelKeypress=(()=>{**

**console.log("Key is pressed");**

**});**

**const handelKeyup=(()=>{**

**console.log("Key is Up");**

**});**

**const handelKeydown=(()=>{**

**console.log("Key is Down");**

**});**

**Event Listener:-**

Event Listener in JavaScript is a procedure or method executed when any event occurs like a click event. An event is something that occurs in HTML elements. For example: When a button is clicked, when a key is pressed, etc. Event Listeners listen to those occurring events and execute the function performing any task.

let data = document.getElementById("click\_me");

data.addEventListener("click", ()=>{

console.log("Click by eventListener");

});

data.addEventListener("click", ()=>{

console.log("Click by eventListener 2");

});

data.addEventListener("click", ()=>{

console.log("Click by eventListener 3");

});

data.addEventListener("click", ()=>{

console.log("Click by eventListener 4");

});

**Benefit of Eventlistener**

1. it will run in javascript only

2. you can run same event twice

3. in callback function we can add event object

**Running same event twice**

let data = document.getElementById("click\_me")

data.addEventListener("click", ()=>{

console.log("Click by eventListener 1");

})

data.addEventListener("click", ()=>{

console.log("Click by eventListener 2");

})

**Adding event object**

let detail = document.getElementById("btn\_click")

detail.addEventListener("click", (e)=>{

console.log("event object",e)

console.log("Click by eventListener");

});