

## Day 17



### Table Navigation:

#### **What is Table Navigation?**

Table navigation means accessing and moving through parts of an HTML table using JavaScript. The browser treats table elements (`<table>`, `<thead>`, `<tbody>`, `<tr>`, `<td>`, etc.) as nested DOM elements, so you can easily navigate between them.

#### **Why Use Table Navigation?**

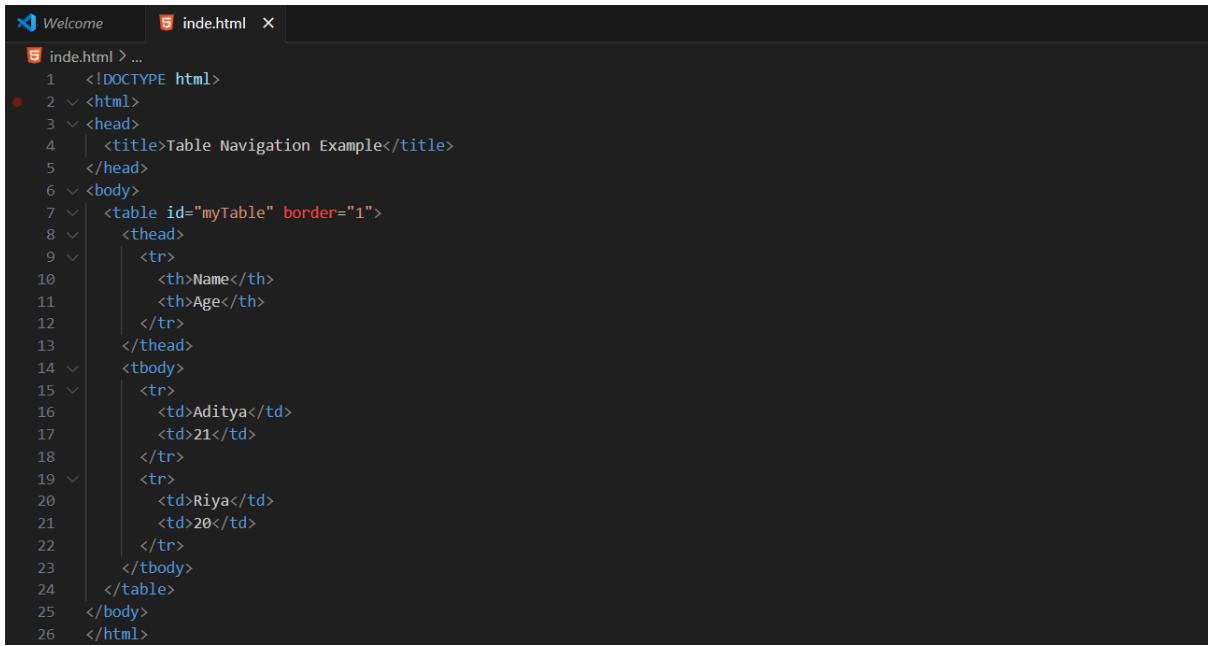
We use it to:

- Read or modify table data
- Highlight or style specific rows/cells
- Access elements like `thead`, `tbody`, `tr`, `td` directly

#### **Table Navigation Properties**

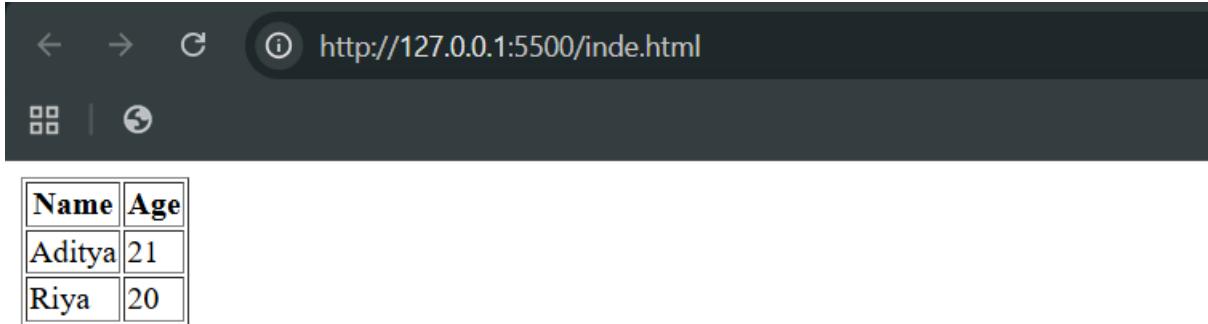
Property	Description	Example
<code>table.rows</code>	All rows in the table	<code>table.rows</code>
<code>table.tHead</code>	Refers to the <code>&lt;thead&gt;</code> section	<code>table.tHead</code>
<code>table.tBodies</code>	All <code>&lt;tbody&gt;</code> sections	<code>table.tBodies</code>
<code>table.tFoot</code>	Refers to the <code>&lt;tfoot&gt;</code> section (if any)	<code>table.tFoot</code>
<code>table.caption</code>	Refers to <code>&lt;caption&gt;</code> element (if any)	<code>table.caption</code>
<code>tbody.rows</code>	All rows in a specific <code>&lt;tbody&gt;</code>	<code>table.tBodies[0].rows</code>
<code>row.cells</code>	All cells ( <code>&lt;td&gt;</code> or <code>&lt;th&gt;</code> ) in a row	<code>table.rows[0].cells</code>

Now, to explain this let's understand this:



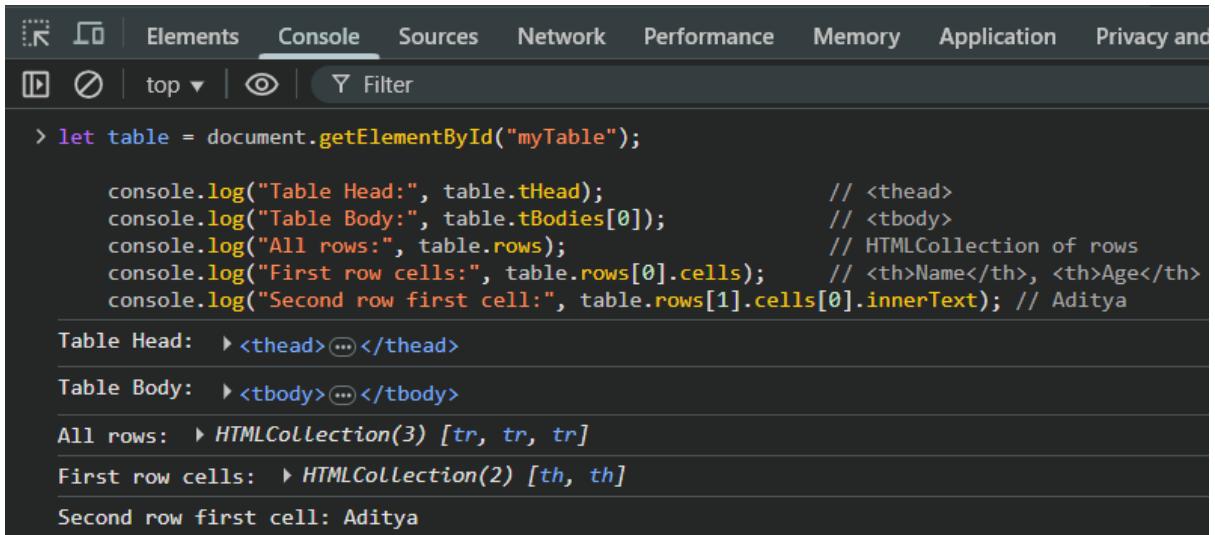
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 | <title>Table Navigation Example</title>
5 </head>
6 <body>
7 <table id="myTable" border="1">
8   <thead>
9     <tr>
10    <th>Name</th>
11    <th>Age</th>
12  </tr>
13 </thead>
14 <tbody>
15   <tr>
16    <td>Aditya</td>
17    <td>21</td>
18  </tr>
19   <tr>
20    <td>Riya</td>
21    <td>20</td>
22  </tr>
23 </tbody>
24 </table>
25 </body>
26 </html>
```

Output:



Name	Age
Aditya	21
Riya	20

Now, on console:



```
> let table = document.getElementById("myTable");
  console.log("Table Head:", table.tHead); // <thead>
  console.log("Table Body:", table.tBodies[0]); // <tbody>
  console.log("All rows:", table.rows); // HTMLCollection of rows
  console.log("First row cells:", table.rows[0].cells); // <th>Name</th>, <th>Age</th>
  console.log("Second row first cell:", table.rows[1].cells[0].innerText); // Aditya
Table Head: > <thead>...</thead>
Table Body: > <tbody>...</tbody>
All rows: > HTMLCollection(3) [tr, tr, tr]
First row cells: > HTMLCollection(2) [th, th]
Second row first cell: Aditya
```

## **Searching the DOM:**

### **What is “Searching the DOM”?**

In JavaScript, searching the DOM means finding elements inside an HTML page using selectors or element properties. The DOM (Document Object Model) represents your web page as a tree of elements, and JavaScript can search through it to access or modify those elements.

### **Why We Search the DOM**

We use DOM searching to:

- Get access to specific HTML elements
- Change text, style, or attributes
- Add interactivity (buttons, inputs, etc.)

### **Common DOM Searching Methods**

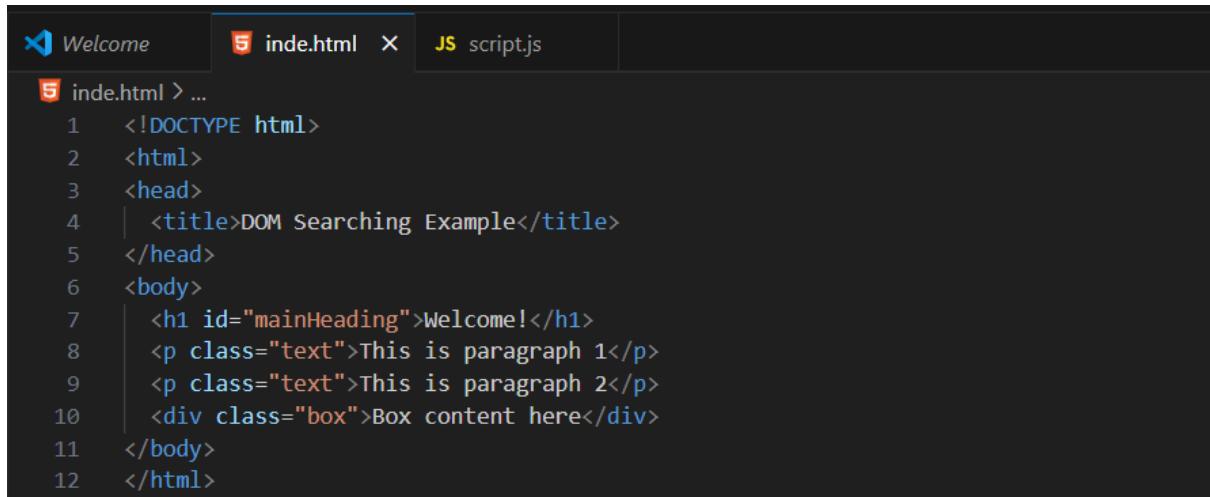
Method	Description	Example
getElementById()	Finds element by its id	document.getElementById("header")
getElementsByClassName()	Finds elements by class name	document.getElementsByClassName("item")
getElementsByTagName()	Finds elements by tag name	document.getElementsByTagName("p")
querySelector()	Finds the first element that matches a CSS selector	document.querySelector(".item")
querySelectorAll()	Finds all elements that match a CSS selector	document.querySelectorAll("p.item")

### **Difference Between Methods**

Method	Returns	Live or Static
getElementById()	Single element	—
getElementsByClassName()	HTMLCollection	Live
getElementsByTagName()	HTMLCollection	Live
querySelector()	First match	Static
querySelectorAll()	NodeList	Static

To understand this, take a very basic example:

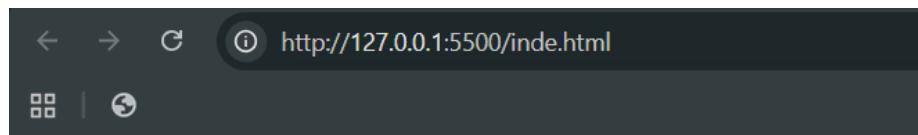
Index.html:



The screenshot shows a code editor with three tabs: "Welcome", "inde.html", and "script.js". The "inde.html" tab is active, displaying the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 | <title>DOM Searching Example</title>
5 </head>
6 <body>
7 | <h1 id="mainHeading">Welcome!</h1>
8 | <p class="text">This is paragraph 1</p>
9 | <p class="text">This is paragraph 2</p>
10 | <div class="box">Box content here</div>
11 </body>
12 </html>
```

Output:



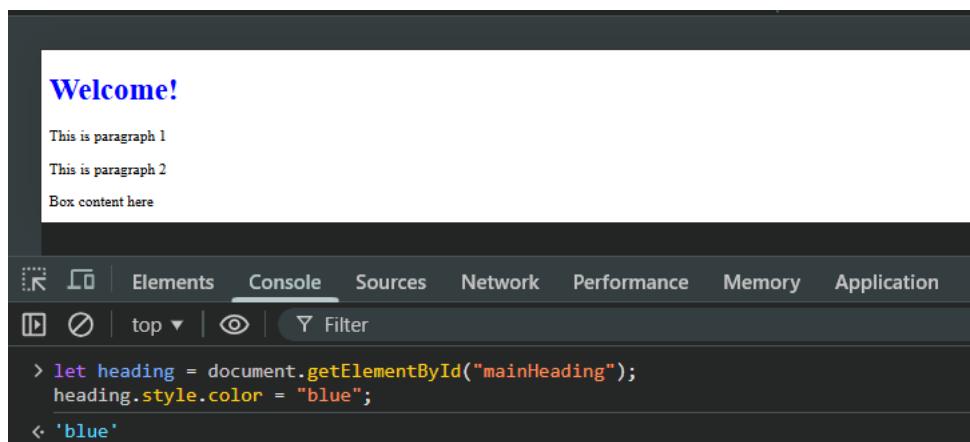
# Welcome!

This is paragraph 1

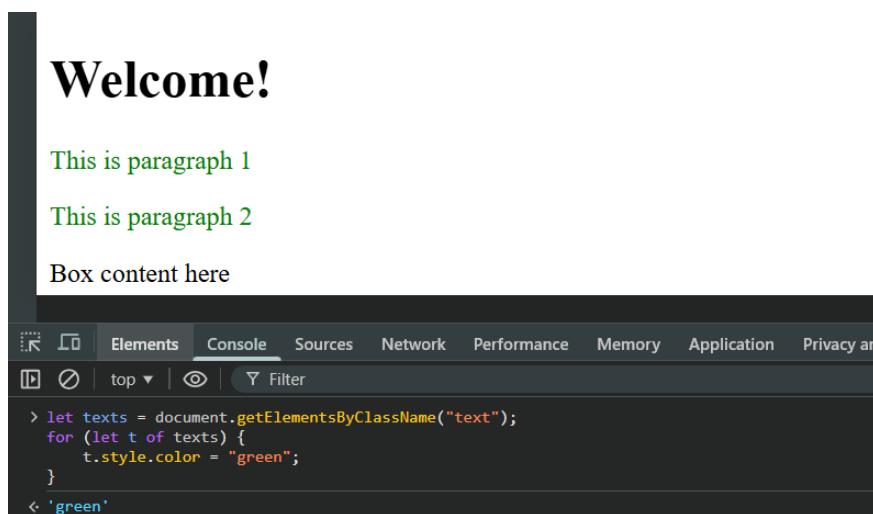
This is paragraph 2

Box content here

Now, applying the DOM searching and changing the properties: `.getElementById`



Now, using the `.getElementsByClassName`:



The screenshot shows a browser's developer tools with the "Console" tab selected. The output pane displays the following JavaScript code and its execution results:

```
> let texts = document.getElementsByClassName("text");
  for (let t of texts) {
    t.style.color = "green";
  }
< 'green'
```

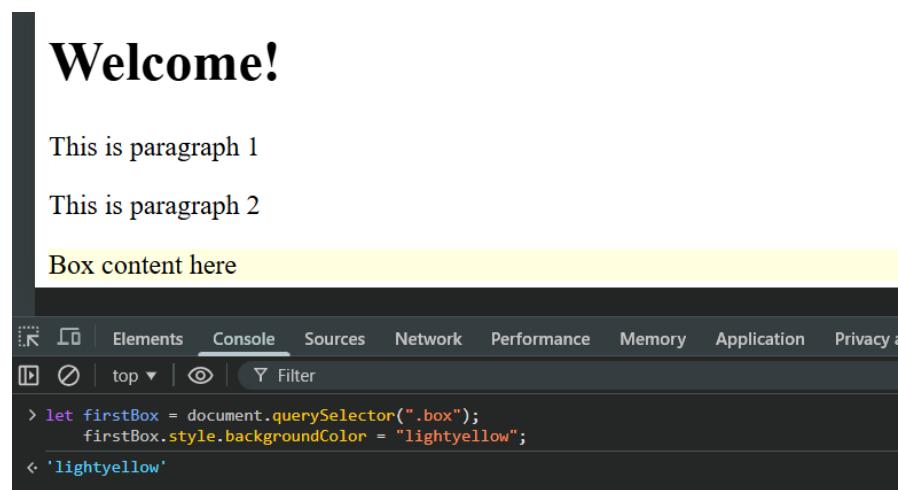
Also, using the `.getElementsByTagName`:



The screenshot shows a browser's developer tools with the "Console" tab selected. The output pane displays the following JavaScript code and its execution results:

```
> let paragraphs = document.getElementsByTagName("p");
  for (let p of paragraphs) {
    p.style.fontWeight = "bold";
  }
< 'bold'
```

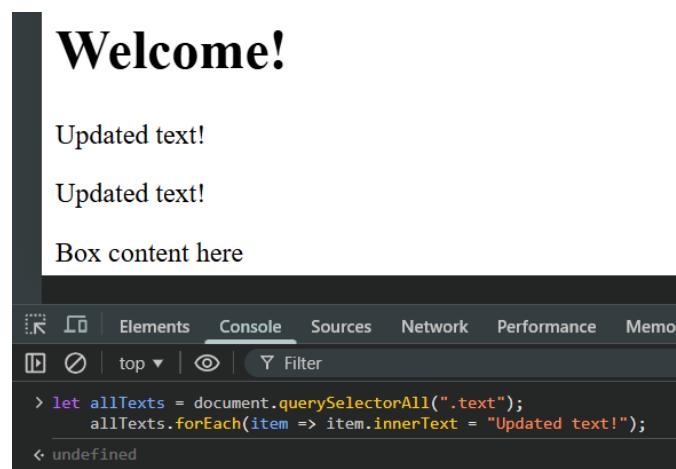
Also, using the `.querySelector`



The screenshot shows a browser's developer tools with the "Console" tab selected. The output pane displays the following JavaScript code and its execution results:

```
> let firstBox = document.querySelector(".box");
  firstBox.style.backgroundColor = "lightyellow";
< 'lightyellow'
```

Also, using .querySelectorAll:



--The End--