Document Object Model (DOM)

- **Definition**: The Document Object Model (DOM) is a programming interface for web documents. It represents the structure of a document as a tree of nodes, allowing programming languages (like JavaScript) to interact with the document.
- **Tree Structure**: The DOM represents the document as a tree of objects. Each object corresponds to a part of the document (e.g., elements, attributes, text).

Key Concepts

1. **Nodes**:
    - **Element Nodes**: Represent HTML tags (e.g., , <p>, <a>).
    - **Text Nodes**: Represent the text inside elements.
    - **Attribute Nodes**: Represent the attributes of elements (e.g., class, id).
    - **Document Nodes**: Represent the entire document.
2. **Accessing Elements**:
    - document.getElementById(id): Selects an element by its ID.
    - document.getElementsByClassName(className): Selects elements by their class name.
    - document.getElementsByTagName(tagName): Selects elements by their tag name.
    - document.querySelector(selector): Selects the first element that matches the CSS selector.
    - document.querySelectorAll(selector): Selects all elements that match the CSS selector.
3. **Manipulating Elements**:
    - **Content**: element.innerHTML and element.textContent.
    - **Attributes**: element.setAttribute(name, value), element.getAttribute(name), element.removeAttribute(name).
    - **Styles**: element.style.propertyName = value.
4. **Creating and Removing Elements**:
    - document.createElement(tagName): Creates a new element.
    - document.createTextNode(text): Creates a new text node.
    - parentNode.appendChild(childNode): Appends a child node to a parent node.
    - parentNode.removeChild(childNode): Removes a child node from a parent node.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <style>
      header {
        background-color: rgb(120, 116, 116);
```

```
      padding-top: 20px;
      height: 100px;
      text-align: center;
    }
  </style>
 </head>
 <body>
  <header>
   <h1>Front End Development</h1>
  </header>
  <main>
   <h3></h3>
   <h2 id="title"></h2>
   <p class="info"></p>
   <h2 class="info"></h2>
   <!-- <div class="queryT"></div> -->
   <p class="queryT">Para</p>
   <h2 class="queryT">H2</h2>
   <ul id="course">
    <li>HTML</li>
   </ul>

   <div id="parent">
    <div id="child">This is Child Div</div>
   </div>
  </main>
  <script>
   var parentTag =   document.getElementById("parent");
   var childTag = document.getElementById("child");
   parentTag.removeChild(childTag)

   var titleTag = document.getElementById("title");
   titleTag.textContent = "About: Front End Classes";
   var infoTag = document.getElementsByClassName("info");
   infoTag[0].textContent =
    "Hema Coding School is dedicated to providing high-quality coding education. Our
experienced instructors are committed to helping you build a strong foundation in
programming.";
    var h3Tag = document.getElementsByTagName('h3');
    h3Tag[0].textContent = "Detail view of FE";

   var queryTag = document.querySelector(".queryT");
   queryTag.innerHTML = "<h1>High Light Content</h1>";

   var queryAllTag = document.querySelectorAll("p");
   for (i = 0; i < queryAllTag.length; i++) {
    queryAllTag[i].style.color = "red";
   }
```

```
    var ulTag = document.getElementById("course");
    var createLiTag = document.createElement("li");
    createLiTag.textContent = "CSS";
    ulTag.appendChild(createLiTag);
  </script>
 </body>
</html>
```

Interview Questions

What is the DOM?

- **Answer**: The DOM (Document Object Model) is a programming interface for web documents. It represents the page structure as a tree of nodes, allowing programming languages to interact with and manipulate the document.

How do you select an element by its ID in the DOM?

- **Answer**: You can select an element by its ID using the document.getElementById(id) method. For example:

```
var element = document.getElementById("myId");
```

What is the difference between innerHTML and textContent?

- **Answer**: innerHTML gets or sets the HTML content of an element, including tags and HTML markup, while textContent gets or sets the text content of an element, ignoring any HTML tags.

How do you create a new element in the DOM?

- **Answer**: You can create a new element using the document.createElement(tagName) method. For example:

```
var newDiv = document.createElement("div");
```

How do you remove a child element from its parent in the DOM?

- **Answer**: You can remove a child element using the removeChild method. For example

```
var parent = document.getElementById("parentElement");
var child = document.getElementById("childElement");
parent.removeChild(child);
```

What is the difference between getElementById and querySelector?

- **Answer**: getElementById selects an element by its ID and returns a single element, whereas querySelector can use any CSS selector and returns the first matching element.

How can you change the style of an element dynamically using JavaScript?

- **Answer**: You can change the style of an element by setting its style properties.

```
var element = document.getElementById("myElement");
element.style.backgroundColor = "blue";
```

What is the difference between querySelector and querySelectorAll?

- **Answer**: querySelector returns the first element that matches the specified CSS selector, while querySelectorAll returns a static NodeList of all matching elements.

What is a NodeList, and how is it different from an HTMLCollection?

- **Answer**: A NodeList is a collection of nodes, which can be either live or static, whereas an HTMLCollection is a live collection of HTML elements. NodeLists can contain any node type, while HTMLCollections only contain elements.