Certainly! Here are concise notes on the Document Object Model (DOM) in JavaScript:

DOM Overview

- **DOM (Document Object Model):** A programming interface for web documents. It represents the page so that programs can change the document structure, style, and content.
- **Tree Structure:** The DOM represents the document as a tree of nodes, where each node can be an element, attribute, or text.

DOM Nodes

- **Element Node:** Represents an HTML element.
- **Attribute Node:** Represents an attribute of an HTML element.
- **Text Node:** Represents the text inside an HTML element.

Accessing DOM Elements

- **`getElementById(id):`** Selects an element by its ID.

```javascript
let element = document.getElementById('myId');
```

- **`getElementsByClassName(className):`** Selects all elements with a specific class name.

```javascript
let elements = document.getElementsByClassName('myClass');
```

- **`getElementsByTagName(tagName):`** Selects all elements with a specific tag name.

```javascript
let elements = document.getElementsByTagName('div');
```

- **`querySelector(selector):`** Selects the first element that matches a CSS selector.

```javascript
let element = document.querySelector('.myClass');
```

- **`querySelectorAll(selector):`** Selects all elements that match a CSS selector.

```javascript
let elements = document.querySelectorAll('.myClass');
```

Manipulating DOM Elements

- **Changing Content:**
  - `innerHTML`: Gets or sets the HTML content inside an element.

```javascript
element.innerHTML = '<p>New content</p>';
```

  - `textContent`: Gets or sets the text content inside an element.

```javascript
element.textContent = 'New text';
```

- **Changing Attributes:**
  - **`setAttribute(attribute, value)`**: Sets the value of an attribute.

```javascript
element.setAttribute('src', 'image.jpg');
```

  - **`getAttribute(attribute)`**: Gets the value of an attribute.

```javascript
let src = element.getAttribute('src');
```

  - **`removeAttribute(attribute)`**: Removes an attribute.

```javascript
element.removeAttribute('src');
```

- **Changing Styles:**
  - **`style.property = value`**: Sets a CSS property.

```javascript
element.style.color = 'blue';
```

Creating and Inserting Elements

- **Create Element:**
  - `createElement(tagName)`: Creates a new element.

```javascript
let newElement = document.createElement('div');
```

- **Append Element:**
  - `appendChild(node)`: Adds a node as the last child.

```javascript
parentElement.appendChild(newElement);
```

  - `insertBefore(newNode, existingNode)`: Inserts a new node before an existing node.

```javascript
parentElement.insertBefore(newElement, referenceElement);
```

Removing Elements

- **Remove Child:**
  - `removeChild(node)`: Removes a child node.

```javascript
parentElement.removeChild(childElement);
```

Event Handling

- **Adding Event Listeners:**
  - **`addEventListener(event, function)`**: Attaches an event handler.

javascript Copy code

```javascript
element.addEventListener('click', function() {
  alert('Element clicked!');
});
```

- **Removing Event Listeners:**
  - **`removeEventListener(event, function)`**: Removes an event handler.

javascript Copy code

```javascript
element.removeEventListener('click', function);
```

Traversal

- **Parent Node:**
  - `parentNode`: Gets the parent of the current node.

```javascript
let parent = element.parentNode;
```

- **Child Nodes:**
  - `childNodes`: Returns a live NodeList of child nodes.

```javascript
let children = element.childNodes;
```

- **First and Last Child:**
  - `**firstChild**`: Gets the first child node.
  - `**lastChild**`: Gets the last child node.

```javascript
let first = element.firstChild;
let last = element.lastChild;
```

- **Sibling Nodes:**
  - `**nextSibling**`: Gets the next sibling node.
  - `**previousSibling**`: Gets the previous sibling node.

```javascript
let next = element.nextSibling;
let prev = element.previousSibling;
```

These notes provide a concise overview of essential DOM concepts and methods in JavaScript.