

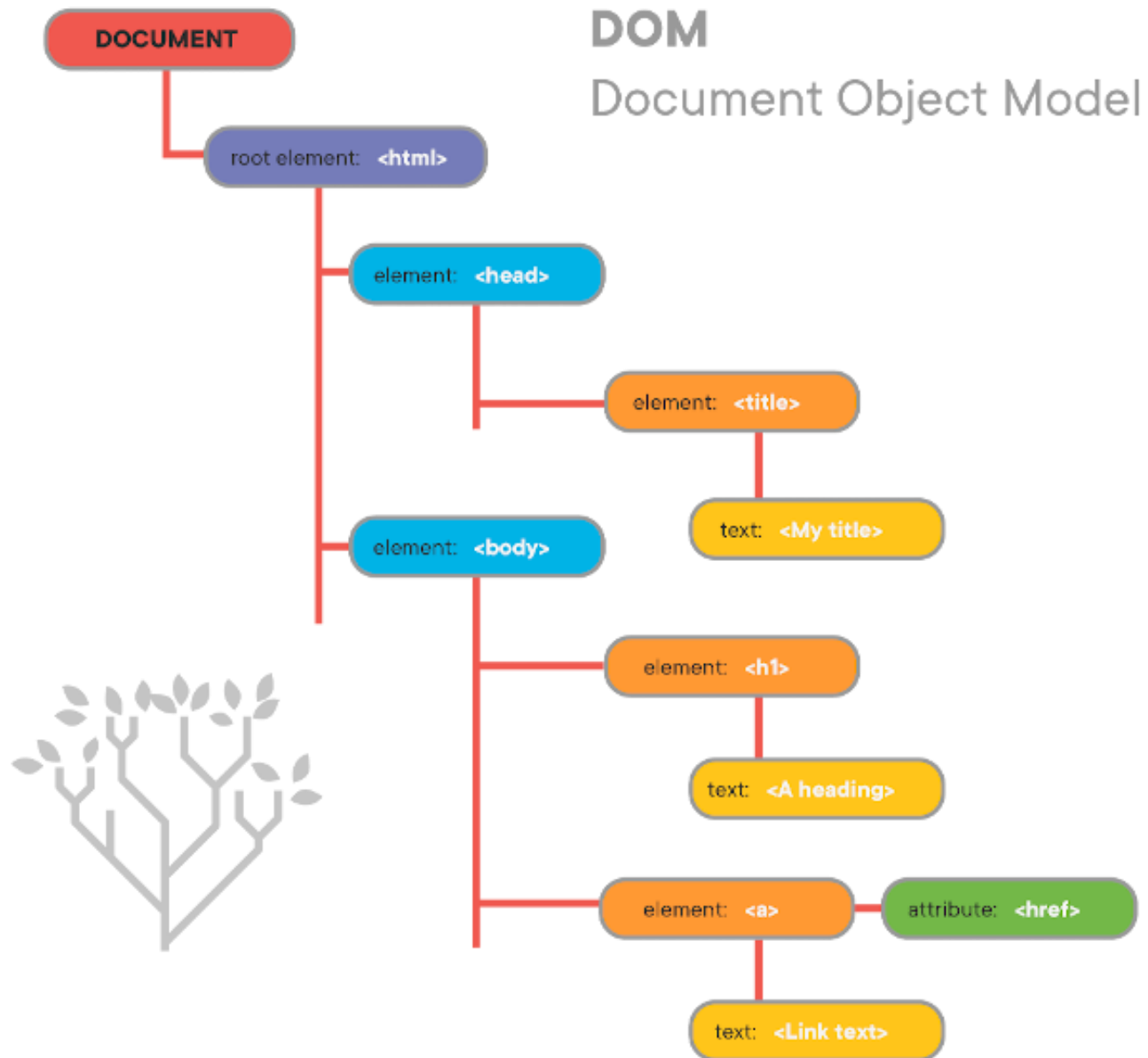
DOM Manipulation in JavaScript

A Beginner's Guide

What is the DOM?

- DOM stands for Document Object Model.
- It's a programming interface for web documents.
- Represents the page as nodes and objects.
- Allows programming languages like JavaScript to interact with the web page.

Structure of the DOM



Why Manipulate the DOM?

- Update content based on user actions.
- Show/hide elements.
- Change styles and classes.
- Create interactive web applications.

Selecting Elements

- `getElementById`: Selects an element by its id.

```
const element = document.getElementById('myElement');
```

- `getElementsByClassName`: Selects elements by their class name.

```
const elements = document.getElementsByClassName('myClass');
```

- **getElementsByTagName**: Selects elements by their tag name.

```
const elements = document.getElementsByTagName('div');
```

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

```
const element = document.querySelector('.myClass');
```

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

```
const elements = document.querySelectorAll('.myClass');
```

Changing Content

- **textContent**: Sets or gets the text content of an element.

```
element.textContent = 'New Text';
```

- **innerHTML**: Sets or gets the HTML content of an element.

```
element.innerHTML = '<p>New HTML Content</p>';
```


Changing Attributes

- **setAttribute**: Sets an attribute on an element.

```
element.setAttribute('src', 'newImage.jpg');
```

- **getAttribute**: Gets an attribute from an element.

```
const src = element.getAttribute('src');
```

- `removeAttribute`: Removes an attribute from an element.

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

Changing Styles

- **Direct Style Changes:** Change styles directly using the style property.

```
element.style.color = 'red';  
element.style.backgroundColor = 'blue';
```

- **Class List:** Add, remove, or toggle classes.

```
element.classList.add('newClass');  
element.classList.remove('oldClass');  
element.classList.toggle('activeClass');
```

Adding and Removing Elements

- Creating a New Element:

```
const newElement = document.createElement('div');  
newElement.textContent = 'Hello World';  
document.body.appendChild(newElement); // Adds the new element to the body
```

- Removing an Element:

```
const elementToRemove = document.getElementById('removeMe');  
elementToRemove.parentNode.removeChild(elementToRemove);
```

Conclusion

- DOM manipulation is a powerful way to create dynamic and interactive web pages.
- Understanding how to select elements, change their content and styles, and respond to user events is essential.
- Practice these concepts to become comfortable with manipulating the DOM in your JavaScript projects.