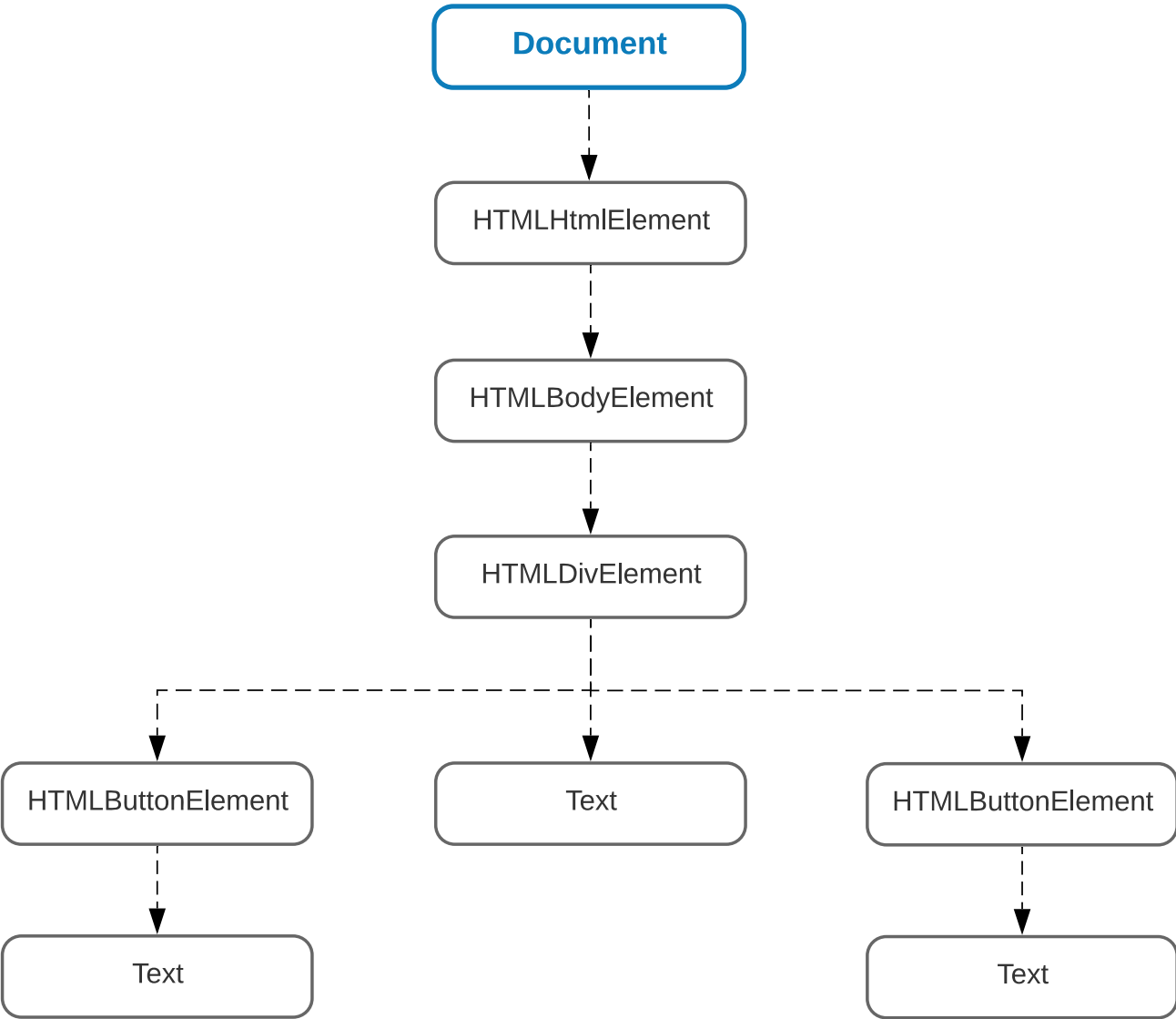


DOM Tree

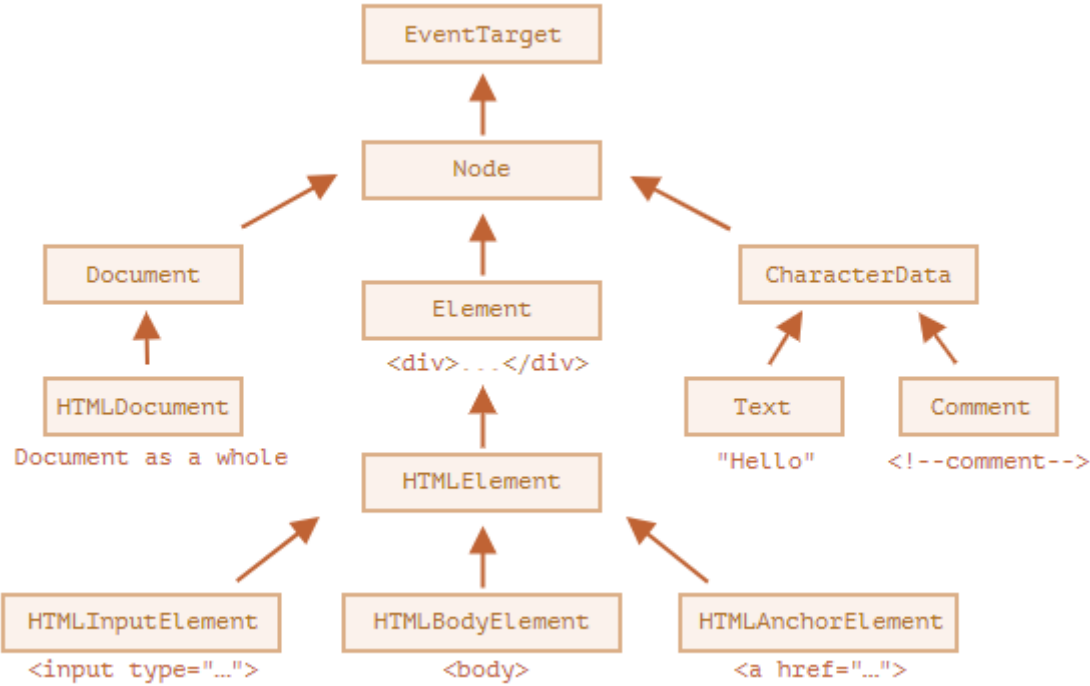


Source: <https://elmprogramming.com/virtual-dom.html>

Nội dung bài học:

- 1. Node data type
- 2. EventTarget interface
- 3. Node interface

1. Node data type



Source: <https://javascript.info/basic-dom-node-properties>

- EventTarget: root "abstract" class.
- Node: "abstract" class, a base for DOM nodes.
- Element: base class for DOM elements.
- HTMLElement: basic class for all HTML elements
 - HTMLInputElement - the class for `<input>` elements
 - HTMLBodyElement - the class for `<body>` elements
 - HTMLAnchorElement - the class for `<a>` elements
 - ...

2. EventTarget interface

#	Name	Desc
1	EventTarget.addEventListener()	Register an event handler of a specific event type
2	EventTarget.removeEventListener()	Remove an event listener

Source: <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget>

3. Node interface

Properties

# Method	Description
<code>Node.childNodes</code>	Trả về một NodeList trực tiếp chứa tất cả các con (phần tử, văn bản, chú thích).
<code>Node.firstChild</code>	Trả về đứa con trực tiếp đầu tiên, nếu không có thì trả về null.
<code>Node.lastChild</code>	Trả về đứa con trực tiếp cuối cùng, nếu không có thì trả về null.

# Method	Description
<code>Node.previousSibling</code>	Trả về node liền kề trước đó trong cây.
<code>Node.nextSibling</code>	Trả về node liền kề tiếp theo trong cây.
<code>Node.nodeName</code>	Tên của node (BODY, DIV, #text, #document).
<code>Node.nodeType</code>	Trả về một giá trị số không dấu đại diện cho loại của node.
<code>Node.nodeValue</code>	Lấy/đặt giá trị của node.
<code>Node.parentNode</code>	Trả về node cha.
<code>Node.parentElement</code>	Trả về một phần tử là cha của node này.
<code>Node.textContent</code>	Lấy/đặt nội dung văn bản của một phần tử và tất cả con cháu của nó.

```
<div id="container">
  <p>This is a paragraph</p>
  <span>This is a span</span>
  <!-- Comment node -->
  <!-- This is a comment -->
</div>
```

```
// Lấy đối tượng container từ DOM
const container = document.getElementById('container');

// Lấy NodeList chứa tất cả các con của container
const childNodes = container.childNodes;
console.log('All child nodes:', childNodes);

// Lấy đứa con trực tiếp đầu tiên của container
const firstChild = container.firstChild;
console.log('First child:', firstChild);

// Lấy đứa con trực tiếp cuối cùng của container
const lastChild = container.lastChild;
console.log('Last child:', lastChild);

// Lấy node liền kề trước đó của container
const previousSibling = container.firstChild.nextSibling;
console.log('Previous sibling:', previousSibling);

// Lấy node liền kề tiếp theo của container
const nextSibling = container.firstChild.nextSibling.nextSibling;
console.log('Next sibling:', nextSibling);

// Lấy tên của node container
const nodeName = container.nodeName;
console.log('Node name:', nodeName);
```

```
// Lấy loại của node container
const nodeType = container.nodeType;
console.log('Node type:', nodeType);

// Lấy giá trị của node container
const nodeValue = container.nodeValue;
console.log('Node value:', nodeValue);

// Lấy node cha của container
const parentNode = container.parentNode;
console.log('Parent node:', parentNode);

// Lấy phần tử cha của container
const parentElement = container.parentElement;
console.log('Parent element:', parentElement);

// Lấy và đặt nội dung văn bản của container và tất cả các con cháu của nó
const textContent = container.textContent;
console.log('Text content:', textContent);
container.textContent = 'New text content';
```

Methods

# Name	Description
Node.insertBefore(newNode, referenceNode)	Chèn newNode trước referenceNode .
Node.appendChild(childNode)	Thêm childNode làm con cuối cùng của node hiện tại.
Node.cloneNode(deep) - deep = true / false	Sao chép node hiện tại, deep = true sẽ sao chép toàn bộ cây con.
Node.contains(otherNode)	Kiểm tra xem node hiện tại có chứa otherNode hay không.
Node.hasChildNodes()	Kiểm tra xem node hiện tại có node con nào không.
Node.removeChild(childNode)	Xóa một node con từ node hiện tại.
Node.replaceChild(newChild, oldChild)	Thay thế oldChild bằng newChild .

Ví dụ

```
<div id="existingDiv">
  This is an existing div element.
</div>
```

```
// Tạo một phần tử div mới
const newDiv = document.createElement('div');
newDiv.textContent = 'New div element';
```

```
// Lấy thẻ div hiện tại từ DOM
const currentDiv = document.getElementById('existingDiv');

// Chèn newDiv trước currentDiv
currentDiv.parentNode.insertBefore(newDiv, currentDiv);

// Sao chép newDiv và các nút con của nó
const clonedDiv = newDiv.cloneNode(true);

// Kiểm tra xem currentDiv có chứa newDiv không
console.log('currentDiv contains newDiv:', currentDiv.contains(newDiv));

// Kiểm tra xem newDiv có node con nào không
console.log('newDiv has child nodes:', newDiv.hasChildNodes());

// Thay thế newDiv bằng clonedDiv
currentDiv.parentNode.replaceChild(clonedDiv, newDiv);

// Xóa clonedDiv
currentDiv.parentNode.removeChild(clonedDiv);
```

Source: <https://developer.mozilla.org/en-US/docs/Web/API/Node>

nodeType

Value	Description
Node.ELEMENT_NODE	Một node phần tử như <p> hoặc <div>
Node.ATTRIBUTE_NODE	Một thuộc tính của một phần tử
Node.TEXT_NODE	Văn bản thực sự bên trong một phần tử hoặc thuộc tính
Node.CDATA_SECTION_NODE	Một phần tử CDATA, như <![CDATA[...]]>
Node.PROCESSING_INSTRUCTION_NODE	Một xử lý chỉ dẫn của một tài liệu XML, như <?xml-stYLESHEET ... ?>
Node.COMMENT_NODE	Một node chú thích, như <!-- ... -->
Node.DOCUMENT_NODE	Một node tài liệu
Node.DOCUMENT_TYPE_NODE	Một node DocumentType, như <!DOCTYPE html>
Node.DOCUMENT_FRAGMENT_NODE	Một node DocumentFragment

Source: <https://developer.mozilla.org/en-US/docs/Web/API/Node/nodeType>

```
// Tạo một phần tử div mới
const divElement = document.createElement('div');
divElement.textContent = 'This is a div element';
```

```
// Tạo một đoạn văn bản
const textNode = document.createTextNode('This is a text node');

// Tạo một chú thích
const commentNode = document.createComment('This is a comment node');

// In ra loại của từng node
console.log('divElement.nodeType:', divElement.nodeType);
console.log('textNode.nodeType:', textNode.nodeType);
console.log('commentNode.nodeType:', commentNode.nodeType);
```

kết quả:

```
divElement.nodeType: 1 // divElement là một phần tử (ELEMENT_NODE) với nodeType là 1.
textNode.nodeType: 3 // textNode là một node văn bản (TEXT_NODE) với nodeType là 3.
commentNode.nodeType: 8 // commentNode là một node chú thích (COMMENT_NODE) với nodeType là 8.
```

Tham khảo

- <https://javascript.info/basic-dom-node-properties>
- <https://developer.mozilla.org/en-US/docs/Web/API/Element>