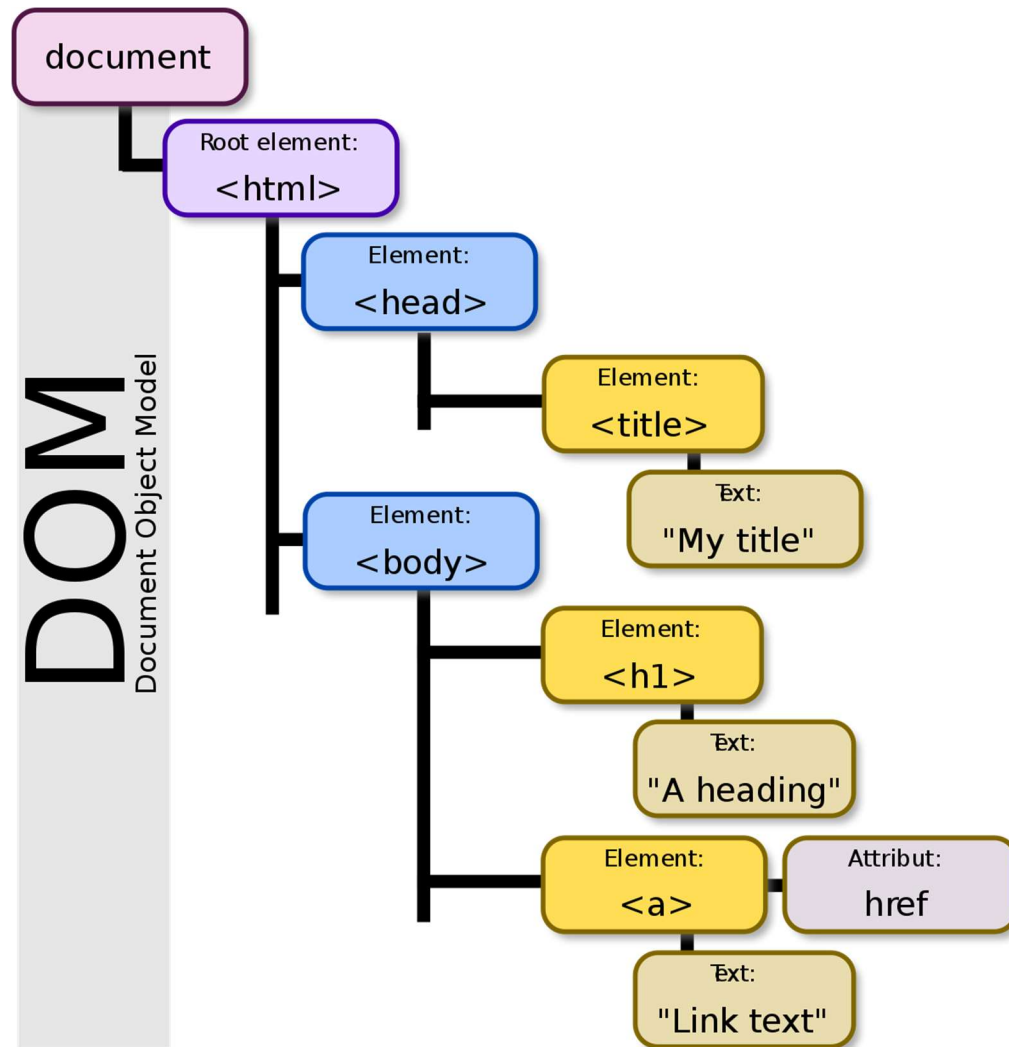


Khái niệm cơ bản Dom - JavaScript

Tìm hiểu về đối tượng DOM document trong JavaScript, lựa chọn phần tử thay đổi các phần tử với các hàm như getElementById, getElementsByTagName

DOM cơ bản

Khi bạn mở ra một trang web trong trình duyệt, nội dung HTML của trang được tải và được dựng(vẽ) trực quan trên màn hình. Để làm việc đó, trình duyệt tạo ra một đối tượng có tên **Document Object Model** gọi tắt là **DOM**, đối tượng này là mô hình logic biểu diễn cấu trúc của trang HTML. **DOM** của văn bản HTML biểu diễn bởi cây là tập hợp các hộp phần tử lồng nhau.



Hình ảnh đối tượng DOM

JavaScript có thể sử dụng đối tượng DOM để tương tác với cấu trúc trang với các hành động như thêm, xóa, chỉnh sửa các phần tử của trang.

Cây đối tượng DOM

Như hình ảnh trên, **DOM** biểu diễn ở dạng cây. Các phần tử HTML trở thành các nút trong cây. Như cấu trúc dữ liệu dạng cây, các nút có chứa các nút con. Có một số nút cùng cấp trong cây.

Xem lại hình trên thì:

- **<html>** có 2 nút con là **<head>** và **<body>**
- **<head>** có 1 phần tử con là **<title>** và phần tử cha là **<html>**
- **<title>** có 1 phần tử cha là **<head>** và không có phần tử con nào
- **<body>** có phần tử cha là **<html>** và phần tử con là **<h1>**, **<a>** ...

Hiểu rõ cấu trúc quan hệ cha con giữa các phần tử là kiến thức quan trọng giúp JavaScript tương tác với HTML

Đối tượng document

Có một đối tượng đã được định nghĩa sẵn trong JavaScript mà qua đó truy cập đến mọi phần tử của **DOM**, đó chính là đối tượng có tên **document**

Khi sử dụng JavaScript bạn sẽ sử dụng đối tượng này là gốc tương tác với cấu trúc DOM qua đó tương tác với mọi phần tử HTML, ví dụ

```
<script>
  document.body.innerHTML = "Some text";
</script>
```

Bạn thấy, **body** là một phần tử của **DOM**, bạn đã truy cập đến phần tử và thay đổi nội dung thuộc tính **innerHTML**

Thuộc tính **innerHTML** có trong mọi phần tử và bạn sử dụng nó để thay đổi nội dung trong phần tử

Chọn phần tử trong DOM

Tất cả các phần tử HTML như **body**, **a**, **p** ... đều là đối tượng trong JavaScript. Đối tượng **document** có các phương thức giúp bạn chọn ra phần tử mong muốn. Nó có ba phương thức thông dụng:

```
<script>
    //tìm 1 phần tử theo id của nó - trả về HTMLElement hoặc null nếu không thấy
    document.getElementById(id)

    //tìm các phần tử theo class - trả về HTMLCollection
    document.getElementsByClassName(name)

    //tìm các phần tử theo tên thẻ - trả về HTMLCollection
    document.getElementsByTagName(name)
</script>
```

getElementById

Ví dụ sau, tìm phần tử có id là demo, sau đó đổi nội dung phần tử thành Hello World!

```
<html>
  <head>
    <title>DEMO JAVASCRIPT</title>
  </head>
  <body>
    <div id="demo">Demo select Element</div>

    <script>
      var ele = document.getElementById('demo');
      ele.innerHTML = "Hello World!";
    </script>
  </body>
</html>
```

getElementsByTagName

Phương thức **getElementsByTagName** tìm tất cả các phần tử có tên thẻ trong tham số hàm. Nó trả về một tập hợp **HTMLCollection** trong nó chứa các phần tử **HTMLElement** tìm thấy. Bạn có thể sử dụng thuộc tính **length** để biết số lượng phần tử trả về, từ đó có thể duyệt qua tập hợp này dễ dàng

Ví dụ tìm tất cả các phần tử đoạn văn (thẻ **p**). Sau đó đổi toàn bộ nội dung các đoạn văn thành **Hello World!**

```
<html>
  <head>
```

```

    <title>DEMO JAVASCRIPT</title>
  </head>
  <body>

    <p>hi</p>
    <p>hello</p>
    <p>hi</p>

    <script>
      var arr = document.getElementsByTagName("p");
      for (var x = 0; x < arr.length; x++) {
        arr[x].innerHTML = "Hello World!";
      }
    </script>
  </body>

</html>

```

Đoạn mã trên biến đổi nội dung các thẻ p thành: Hello World!

getElementsByName

Phương thức `getElementsByName` tìm tất cả các phần tử có thuộc tính class (lớp) đưa ra. Nó cũng trả về HTMLCollection

```

var arr = document.getElementsByClassName("demo");

//truy cập phần tử thứ 2 tìm được
arr[1].innerHTML = "Hi";

```

Phần tử trong DOM - HTMLElement

Mỗi một phần tử trong **DOM** đều có các thuộc tính và phương thức cung cấp thông tin về phần tử đó trong **DOM**

Các thuộc tính của phần tử trong **DOM**:

- `element.childNodes` : trả về mảng các phần tử con của nó
- `element.firstChild` : phần tử con đầu tiên
- `element.lastChild` : phần tử con cuối cùng
- `element.hasChildNodes` : trả về `true` nếu có phần tử con, ngược lại là `false`
- `element.nextSibling` : phần tử cùng cấp tiếp theo
- `element.previousSibling` : phần tử cùng cấp trước
- `element.parentNode` : trả về nút cha

Đầy đủ các thuộc tính của phần tử HTML xem tại [HTML ELEMENT](#)

Xem ví dụ:

```
<html>
  <body>
    <div id="demo">
      <p>some text</p>
      <p>some other text</p>
    </div>

    <script>
      var a = document.getElementById("demo");
      var arr = a.childNodes;
      for(var x=0;x<arr.length;x++) {
        arr[x].innerHTML = "new text";
      }
    </script>

  </body>
</html>
```

Tổng kết

Toàn bộ cấu trúc HTML của trang được mô tả trong **DOM**, Javascript truy cập tới DOM thông qua đối tượng xây dựng sẵn là **document**

Tìm một phần tử trong DOM bằng cách sử dụng các phương

thức: **document.getElementById(id)**, **document.getElementsByClassName(name)**, **document.getElementsByTagName(name)**

Mỗi phần tử DOM lại có các thuộc tính, phương thức quá đó có thể tương tác với HTML một cách dễ dàng. Các phần sau chúng ta sẽ tìm cách tương tác với các phần tử DOM