ĐẠI NAM
UNIVERSITY

Bài giảng Thiết kế, lập trình Front-End

Bài 4. DOM

Giảng viên

Ths. Vũ Đình Thắng

LƯU Ý

KHÔNG NÓI CHUYỆN RIÊNG

KHÔNG SỬ DỤNG ĐIỆN THOẠI

KHÔNG NGỦ GẬT

GHI CHÉP ĐẦY ĐỦ

ĐẠI NAM
UNIVERSITY

Bài 4-HTML DOM

(Giảng viên: Vũ Đình Thắng

MỤC LỤC

NỘI DUNG CHÚ THÍCH

DOM LÀ GÌ?

Javascript là một ngôn ngữ được sử dụng trong các trình duyệt Browser nên nó đóng một vai trò khá quan trọng trong các ứng dụng website.

Nhiệm vụ của Javascript là thao tác với các tài liệu HTML kết hợp với các cú pháp riêng của nó để tạo nên sự ảo diệu của trang web. Để thao tác được với các thẻ HTML thì nó phải thông qua một cơ chế ta gọi là DOM.

DOM là gì?

DOM là viết tắt của chữ **D**ocument **O**bject **M**odel, dịch tạm ra là mô hình các đối tượng trong tài liệu HTML.

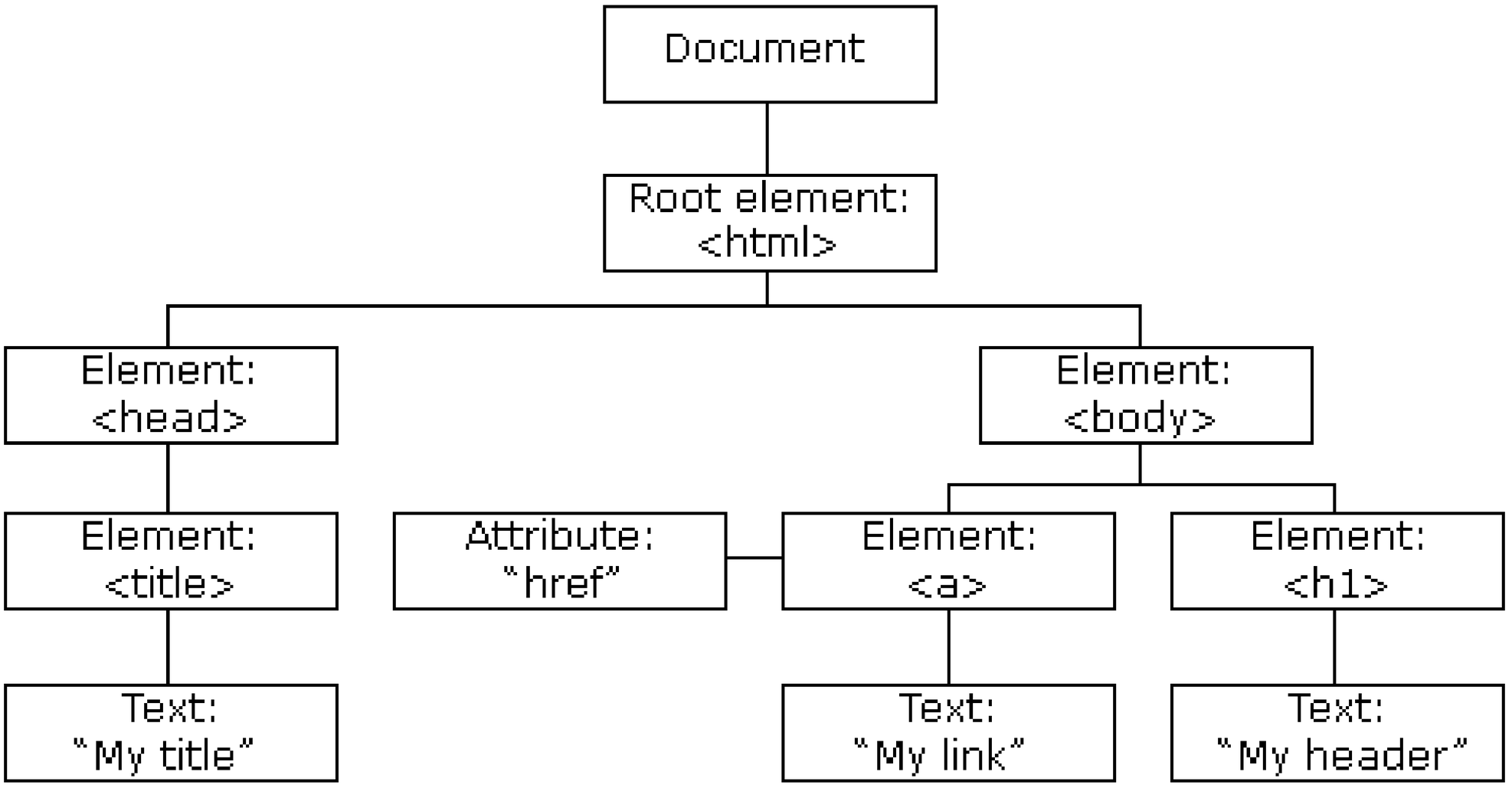
Như các bạn biết trong mỗi thẻ HTML sẽ có những thuộc tính (Properties) và có phân cấp cha - con với các thẻ HTML khác.

Sự phân cấp và các thuộc tính của thẻ HTML này ta gọi là **selector** và trong DOM sẽ có nhiệm vụ xử lý các vấn đề như đổi thuộc tính của thẻ, đổi cấu trúc HTML của thẻ,...

DOM được dùng để truy xuất các tài liệu dạng HTML và XML, có dạng một cây cấu trúc dữ liệu, và thông thường mô hình DOM độc lập với hệ điều hành và dựa theo kỹ thuật lập trình hướng đối tượng để mô tả tài liệu.

DOM là gì?

Cây đối tượng HTML DOM



DOM LÀ GÌ?


```
<html>
 <body>
  <h1 id="main-content"></h1>
  <script language="javascript">
   document.getElementById("main-content").innerHTML = ”Cấu trúc DOM"
  </script>
 </body>
</html>
```

Ta thấy sử dụng cấu trúc DOM

```
document.getElementById("main-content").innerHTML = “Cấu trúc DOM"
```

Đoạn code này có ý nghĩa rằng tìm thẻ có id="main-content"và gán nội dung HTML bên trong của thẻ này là dòng chữ ”Cấu trúc DOM".

DOM LÀ GÌ?

Với mô hình đối tượng, JavaScript có được tất cả sức mạnh cần thiết để tạo HTML động:

JavaScript có thể thay đổi tất cả các thành phần HTML trong trang

JavaScript có thể thay đổi tất cả thuộc tính HTML trong trang

JavaScript có thể thay đổi tất cả các kiểu CSS trong trang

JavaScript có thể xóa các thành phần và thuộc tính HTML hiện có

JavaScript có thể thêm các thành phần và thuộc tính HTML mới

JavaScript có thể phản ứng với tất cả các sự kiện HTML hiện có trong trang

JavaScript có thể tạo các sự kiện HTML mới trong trang

HTML DOM LÀ GÌ?

HTML DOM là một mô hình **đối tượng** và **giao diện lập trình** tiêu chuẩn cho HTML. Nó định nghĩa:

- Các phần tử HTML dưới dạng **đối tượng**
- Thuộc **tính** của tất cả các phần tử HTML
- Các **phương pháp** truy cập tất cả các phần tử HTML
- Các **sự kiện** cho tất cả các phần tử HTML

Nói cách khác: **HTML DOM là một tiêu chuẩn về cách lấy, thay đổi, thêm hoặc xóa các phần tử HTML.**

Giao diện lập trình DOM

- HTML DOM có thể được truy cập bằng JavaScript (và bằng các ngôn ngữ lập trình khác).
- Trong DOM, tất cả các phần tử HTML được định nghĩa là đối tượng .
- Giao diện lập trình là các thuộc tính và phương thức của từng đối tượng.
- Thuộc tính là một giá trị mà bạn có thể nhận hoặc đặt (như thay đổi nội dung của phần tử HTML).
- Phương thức là một hành động bạn có thể thực hiện (như thêm hoặc xóa phần tử HTML).

CÁC LOẠI DOM TRONG JAVASCRIPT

Danh sách nhóm DOM:

- DOM document: có nhiệm vụ lưu trữ toàn bộ các thành phần trong tài liệu của website
- DOM element: có nhiệm vụ truy xuất tới thẻ HTML nào đó thông qua các thuộc tính như tên class, id, name của thẻ HTML
- DOM HTML: có nhiệm vụ thay đổi giá trị nội dung và giá trị thuộc tính của các thẻ HTML
- DOM CSS: có nhiệm vụ thay đổi các định dạng CSS của thẻ HTML
- DOM Event: có nhiệm vụ gán các sự kiện như onclick(), onload() vào các thẻ HTML
- DOM Listener: có nhiệm vụ lắng nghe các sự kiện tác động lên thẻ HTML đó
- DOM Navigation dùng để quản lý, thao tác với các thẻ HTML, thể hiện mối quan hệ cha - con của các thẻ HTML
- DOM Node, Nodelist: có nhiệm vụ thao tác với HTML thông qua đối tượng (Object)

Method	Mô tả
document.getElementById(*id*)	Tìm 1 phần tử theo id
document.getElementsByTagName(*name*)	Tìm 1 phần tử theo tên thẻ
document.getElementsByClassName(*name*)	Tìm 1 phần tử theo tên lớp

Thay đổi các phần tử HTML

Thuộc tính	Mô tả
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element
Phương thức	**Mô tả**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

Thêm và xóa các phần tử

Phương thức	Mô tả
document.createElement(*element*)	Tạo 1 phần tử HTML
document.removeChild(*element*)	Xóa bỏ 1 phần tử
document.appendChild(*element*)	Nối them 1 phần tử
document.replaceChild(*new, old*)	Thay thế 1 phần tử con
document.write(*text*)	Write into the HTML output stream

Thêm trình xử lý sự kiện

Method	Description
document.getElementById(*id*).onclick = function(){*code*}	Thêm mã xử lý sự kiện vào sự kiện onClick

Document.getElementById

Phương thức này giúp ta truy xuất đến thẻ HTML thông qua ID. Nếu trong tài liệu html của bạn bị trung ID thì nó sẽ lấy thẻ html có ID trùng khớp đầu tiên

```
<body>
    <p id="p1">Tôi là javascript</p>
    <script type="text/javascript" language="javascript">
        var noidung=document.getElementById("p1");
        alert(noidung.innerHTML);
    </script>
</body>
```

Document.getElementByTagName

Phương thức này lấy tất cả các thẻ html có tên trùng với tham số truyền vào. Ví dụ dưới đây là mình lấy tất cả thẻ **h2**.

```
<body>
    <p id="p1">Tôi là javascript</p>
    <h2>Chào bạn</h2>
    <h2>Tôi là javascri
    <script type="text/
        var h=document.getElementsByTagName("h2");
        h[0].style.background="Green";
        h[0].style.color="white";
        h[1].style.background="yellow";
        h[1].style.color="red";
        h[1].style.fontStyle="italic";
        var p1=document.getElementById("p1");
        p1.style.color="blue";
        p1.innerHTML=h[0].innerHTML;
</script>
```

```
(method) Document.getElementsByTagName<"h2">(qualifiedName: "h2"):
HTMLCollectionOf<HTMLHeadingElement> (+4 overloads)
```

Chào bạn

Chào bạn

Tôi là javascript

Document.getElementByClassName

Phương thức này lấy tất cả thẻ html có tên class trùng với tên của tham số truyền vào.

```
 8    <body>
 9        <p id="p1">Tôi là javascript</p>
10        <h2 class="h">Chào bạn</h2>
11        <script type="text/javascript" language="javascript">
12            var h=document.getElementsByClassName("h");
13            var p1=document.getElementById("p1");
14            p1.style.color="blue";
15            p1.innerHTML=h[0].innerHTML;
16        </script>
```

Chào bạn

Chào bạn

Document.getElementByName

Phương thức này lấy tất cả thẻ html có tên **name** trùng với tên của tham số truyền vào.

```
<body>
    <p id="p1">Tôi là javascript</p>
    <h2 name="h">Chào bạn</h2>
    <script type="text/javascript" language="javascript">
        var h=document.getElementsByName("h");
        var p1=document.getElementById("p1");
        p1.style.color="blue";
        p1.innerHTML=h[0].innerHTML;
    </script>
</body>
```

Chào bạn

Chào bạn

CÁC PHẦN TỬ CSS SELECTOR

Tìm tất cả các phần tử HTML khớp với một bộ chọn CSS được chỉ định (id, tên lớp, loại, thuộc tính, giá trị của thuộc tính, v.v.), hãy sử dụng querySelectorAll()phương thức này.
Ví dụ này trả về danh sách tất cả <p>các phần tử có class="intro"..

```
<body>
    <p>Tìm các phần tử HTML bằng Query Selector</p>
    <p class="intro">Hello World!.</p>
    <p class="intro">Ví dụ này minh họa phương thức <b>querySelectorAll</b>.</p>
    <p id="demo"></p>
    <script type="text/javascript" language="javascript">
        const x = document.querySelectorAll("p.intro");
        document.getElementById("demo").innerHTM ='Đoạn đầu tiên (index 0) với class="intro" là: ' + x[0].innerHTML;
    </script>
```

Tìm các phần tử HTML bằng Query Selector

Hello World!.

Ví dụ này minh họa phương thức **querySelectorAll**.

CÁC PHẦN TỬ HTML OBJECT COLLECTIONS HỌC ĐỂ THAY ĐỔI

Tìm phần tử biểu mẫu có id="frm1", trong bộ sưu tập biểu mẫu và hiển thị tất cả các giá trị phần tử:

JavaScript HTML DOM

Finding HTML Elements Using **document.forms**.

First name: Donald
Last name: Duck

Submit

These are the values of each element in the form:

Donald
Duck
Submit

```
<body>
    <h2>JavaScript HTML DOM</h2>
    <p>Finding HTML Elements Using <b>document.forms</b>.</p>
    <form id="frm1" action="/action_page.php">
        First name: <input type="text" name="fname" value="Donald"><br>
        Last name: <input type="text" name="lname" value="Duck"><br><br>
        <input type="submit" value="Submit">
    </form>
    <p>These are the values of each element in the form:</p>
    <p id="demo"></p>
    <script>
        const x = document.forms["frm1"];
        let text = "";
        for (let i = 0; i < x.length ;i++) {
            text += x.elements[i].value + "<br>";
        }
        document.getElementById("demo").innerHTML = text;
    </script>
</body>
```

CÁC PHẦN TỬ HTML FORM

Lấy các phần tử FORM theo cú pháp:

Cú pháp:

document.tênForm.Tên_điều_khiển.Value

Ví dụ:

var user=document.dangnhap.username.value;

XỬ LÝ SỰ KIỆN

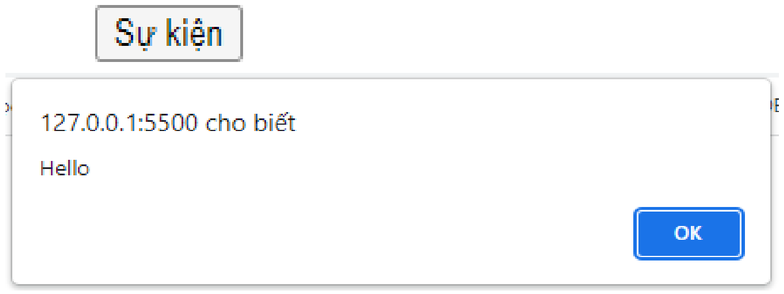
Tất cả các element (phần tử) trên trang web đều có một tập các sự kiện tương ứng

Chú ý: Cùng một element, các browser khác nhau hỗ trợ các tập sự kiện khác nhau

- **onClick:** Được kích hoạt khi nhấn chuột vào một element
- **onLoad và on Unload:** Được kích hoạt khi người dùng vào hoặc thoát khỏi trang web
- **onFocus, onBlur, onChange:** Được kích hoạt khi các trường nhận được focus, mất focus hay được thay đổi giá trị
- **onMouseOver:** Được kích hoạt khi người dùng di chuột lên một element HTML trên form
- **onMouseOut:** Được tạo ra bất cứ khi nào con trỏ chuột di chuyển ra khỏi phần tử đó.

XỬ LÝ SỰ KIỆN

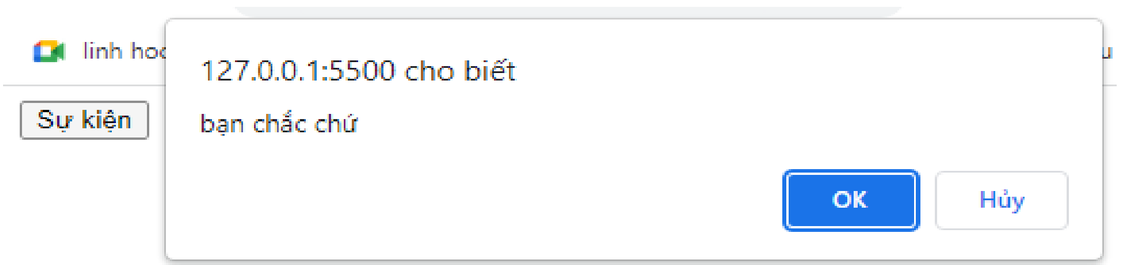
```
<body>
    <button type="button"  onclick="alert('Hello')">Sự kiện</button>
</body>
```

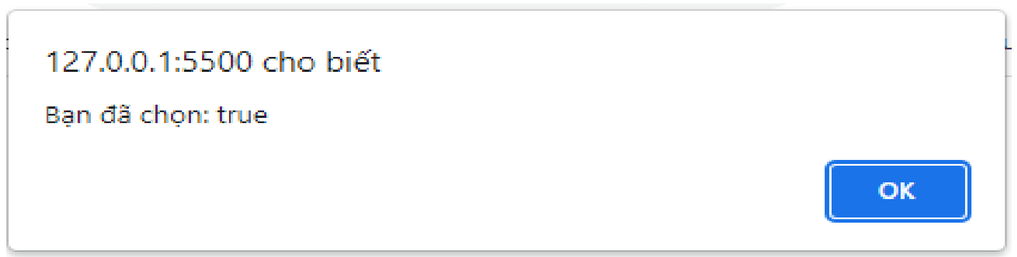


Một sự kiện có thể gọi đến nhiều hàm

Ví dụ:

```
<body>
    <button type="button" onclick="ok=confirm('bạn chắc chứ');alert('Bạn đã chọn: '+ok)">
        Sự kiện
    </button>
</body>
```





LẤY CÁC GIÁ TRỊ TỪ CÁC PHẦN TỬ TRONG FORM

1. TEXTBOX

a. document.tên_form.tên texbox.value

b. Document.getElementById(tên-ID).value

2. VỚI TẤT CẢ CÁC ĐIỀU KHIỂN

- Sử dụng phương thức getElementById(id)
- Sử dụng phương thức getElementsByName(name)
- Sử dụng phương thức getElementsByTagName(name)

3. **GÁN SỰ KIỆN**

<tên thẻ ...sự kiện="tên_hàm(Đối số)">

```
<head>
<script language="javaScript">
        function Tên_hàm(Đối số){
                Khai báo biên
                //Lấy giá trị từ form
                document.tên_form.tên texbox.value hoặc
                document.getElementById(tên textbox).value
            ....
            Tính toán
            Gán giá trị cho ô kết quả
             document.tên_form.tên texbox.value=tên biến kết quả
            }
</script>
</head>
```

```
<body>
        <form name="tên Form"...>
                <tên thẻ ...sự kiện="tên_hàm(Đối số)">
        </Form>
</Body>
```

XỬ LÝ SỰ KIỆN

Ví dụ

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>su kien</title>
    <script type="text/javascript">
        var num =0
        function showLink(num){
            var hienthi=document.getElementById("hienthi");
            if (num==1) {
                hienthi.value= "Bạn đã chọn CĐNTH FPT";
            }
            if (num==2) {
                hienthi.value = "Bạn đã chọn ĐẠI HỌC FPT";
            }
            if (num==3) {
                hienthi.value = "Bạn đã chọn ĐẠI HỌC CÔNG NGHIỆP";
            }
        }
        function thongbao(){
            hienthi.value = "ban phải đưa chuột vào 1 trong các liên kết";
        }
    </script>
</head>
<body>
    <form>
        <input type=text size=60 id="hienthi" >
    </form>
        <a href="#" onMouseOver="showLink(1)"document.bgcolor= "green" onmouseout="thongbao()">CĐN BCVT</a><br>
        <a href="#" onMouseOver="showLink(2)" onmouseout="thongbao()">CĐCQ BCVT</a><br>
        <a href="#" onMouseOver="showLink(3)" onmouseout="thongbao()">CĐ Điện Lực</a><br>
</body>
</html>
```

Bạn đã chọn CĐTH FPT

CĐTH FPT
ĐẠI HỌC FPT
ĐẠI HỌC CÔNG NGHIỆP

ban phải đưa chuột vào 1 trong các liên kết

CĐTH FPT
ĐẠI HỌC FPT
ĐẠI HỌC CÔNG NGHIỆP

KHÁC BIỆT GIỮA INNERTEXT, INNERHTML, TEXTCONTENT

Để lấy nội dung trong thẻ ta thường thấy người ta dùng innerText, innerHTML hoặc có khi dùng textContent. vậy mấy cái này khác nhau như thế nào.

Trong ví dụ dưới đây ta thấy

Ta thấy **innerHTML** sẽ in ra nội dung text và những thẻ trong div này luôn.

Còn **textContent** sẽ in ra nội dung text và nội dung các thẻ bên trong div này luôn.

Còn **innerText** sẽ in ra nội dung text, không in ra nội dung của thẻ ẩn(do thẻ span có display: none, cái này dùng để ẩn thẻ span này).

LẤY DỮ LIỆU TỪ CÁC ĐIỀU KHIÊN

ví dụ:

```
 <div id='hello'>Hello Javascript <span style="display:none;"> 2020</span></div>


  <script>
     var mydiv = document.getElementById('hello');
     console.log(mydiv.innerHTML);
     // Hello Javascript <span style="display:none;"> 2020</span>
     console.log(mydiv.textContent);
     // Hello Javascript  2020
     console.log(mydiv.innerText);
     // Hello Javascript
</script>
```

```
Hello Javascript <span style="display:none;"> 2020</span>
Hello Javascript 2020
Hello Javascript``
```

HTML DOM là gì?

HTML DOM là một chuẩn mô hình object và programming interface cho HTML. nó định nghĩa:

- HTML elements như là objects
- properties của tất cả HTML elements
- methods để truy cập đến tất cả HTML elements
- events cho tất cả HTML elements

HTML DOM là một tiêu chuẩn cho phép bạn thực hiện những công việc thao tác với bất kì một trang web: get, change, add, or delete các thành phần của HTML.

Nút (Node)

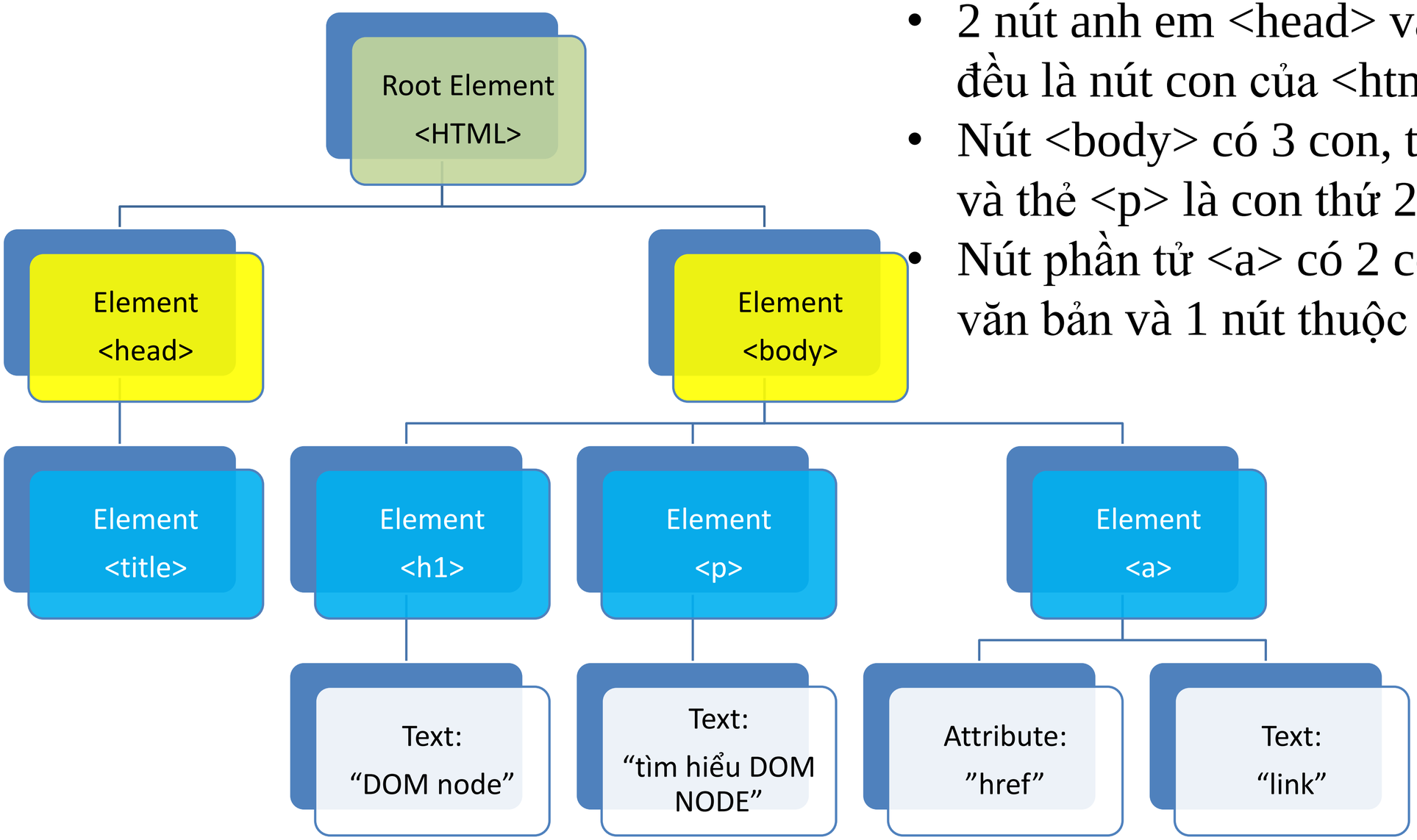
Đối với HTML DOM, cấu trúc dạng cây gọi là DOM Tree có nghĩa là mọi thành phần đều được xem là 1 nút (node), được biểu diễn trên 1 cây . Các phần tử khác nhau sẽ được phân loại nút khác nhau nhưng quan trọng nhất là 3 loại: nút gốc (document node), nút phần tử (element node), nút văn bản (text node).

- **Nút gốc**: chính là tài liệu HTML, thường được biểu diễn bởi thẻ <html>.
- **Nút phần tử (Element)**: biểu diễn cho 1 phần tử HTML.
- **Nút văn bản**: mỗi đoạn kí tự trong tài liệu HTML, bên trong 1 thẻ HTML đều là 1 nút văn bản. Đó có thể là tên trang web trong thẻ <title>, tên đề mục trong thẻ <h1>, hay một đoạn văn trong thẻ <p>.
- Ngoài ra còn có **nút thuộc tính** (attribute node) và **nút chú thích** (comment node).

Quan hệ giữa các nút

- Nút gốc (document) luôn là nút đầu tiên.
- Tất cả các nút không phải là nút gốc đều chỉ có 1 nút cha (parent).
- Một nút có thể có một hoặc nhiều con, nhưng cũng có thể không có con nào.
- Những nút có cùng nút cha được gọi là các nút anh em (siblings).
- Trong các nút anh em, nút đầu tiên được gọi là con cả (firstChild) và nút cuối cùng là con út (lastChild).
- Node lá (leaf) là node không có node con

DOM NODE-Cây cấu trúc DOM



- Nút gốc là <html>.
- 2 nút anh em <head> và <body> là anh em vì đều là nút con của <html>.
- Nút <body> có 3 con, trong đó <h1> là con cả và thẻ <p> là con thứ 2 và thẻ <a> là con út.
- Nút phần tử <a> có 2 con, trong đó có 1 nút văn bản và 1 nút thuộc tính.

DOM NODE-Cây cấu trúc DOM


```
<html lang="en">
<head>
    <title>javascript node</title>
</head>
<body>
    <h1>DOM node</h1>
    <p>tìm hiểu DOM NODE</p>
    <a href="https://codelearn.io">Trang codelean</a>
</body>
</html>
```

Thuộc tính quan hệ:

- **parentNode:** Nút cha
- **childNodes:** Các nút con
- **firstChild:** Nút con đầu tiên
- **lastChild:** Nút con cuối cùng
- **nextSibling:** Nút anh em liền kề sau
- **previousSibling:** Nút anh em liền kề trước

DOM NODE-Cây cấu trúc DOM

- <html> không có node cha, <html> là node gốc
- <html> có hai node con là <head> và <body>,
- <head> và <body> là anh em
- <head> có một node con là <title>
- <title> có một node con là text node “javascript node”
- <body> chứa ba node con là <h1>, <p> và <a>. <h1>, <p> và <a> là anh em
- Node cha của node <head> và <body> là node <html>
- Node cha của text node “tìm hiểu DOM NODE” là node <p>
- Text node “DOM node”, “tìm hiểu DOM NODE” , “Trang codelean”, attribute node “href” đều là node lá
- <head> và <body> là anh em, trong đó <head> là con cả của <html> còn <body> là con út của <html>
- <p> và <a> là anh em, trong đó <p> là con cả của <body> còn <a> là con út của <body>

Thuộc tính và phương thức của NODE

- **DOM** định nghĩa các **thuộc tính** và các **phương thức** cho các node để hỗ trợ cho việc lập trình
- **Thuộc tính** định nghĩa các đặc tính cho node
- **Phương thức** để thực hiện các thao tác với node
  - Truy cập đến node
  - Thêm node con cho node
  - Xóa node c

Các phương thức với NODE:

appendChild(node): Thêm 1 nút con vào nút hiện tại.

removeChild(node): Xóa 1 nút con khỏi nút hiện tại.

Các thuộc tính của node

Thuộc tính	Giải thích
x là đối tượng node	
Node.innerHTML	Giá trị văn bản của x
Node.nodeName	Tên của x
Node.nodeValue	Giá trị của x
Node.nodeType	Kiểu của Node
Node.parentNode	Node cha của x
Node.childNodes	Các node con của x
Node. firstChild	Tham chiếu đến đến nút con đầu tiên
Node.lastChild	Tham chiếu đến đến nút con cuối cùng
Node.nextSibling	Tham chiếu đến nút anh em nằm liền kề sau với nút hiện tại.
Node.previousSibling	Tham chiếu đến nút anh em nằm liền kề trước với nút hiện tại.
Node.attributes	Các node thuộc tính của x

DOM NODE-Cây cấu trúc DOM

```html
<body>
    <div id="domNode">
        <h1>DOM node</h1>
        <p>tìm hiểu DOM NODE HTML</p>
        <a href="https://codelearn.io">Trang codelean</a>
    </div>
    <p id="ketqua"></p>
    <script type="text/javascript" language="javascript">
        var dNode=document.getElementById("domNode");
        var ketqua="Tên node: "+dNode.nodeName+"<br />";
        ketqua+="Nội dung của node: "+dNode.nodeValue+"<br />";
        ketqua+="Node cha: "+dNode.parentNode.nodeName+"<br />";
        ketqua+="Tên Node con đầu tiên: "+dNode.firstElementChild.nodeName+"<br />";
        ketqua+="Nội dung Node con đầu tiên: "+dNode.firstElementChild.innerHTML+"<br />";
        ketqua+="Tên Node con thứ 2: "+dNode.childNodes[3].nodeName+"<br />";
        ketqua+="Nội dung Node con thứ 2: "+dNode.childNodes[3].childNodes[0].nodeValue+"<br />";
        document.getElementById("ketqua").innerHTML=ketqua;
    </script>
</body>
```

DOM node

tìm hiểu DOM NODE HTML

Trang codelean

Tên node: DIV
Nội dung của node: null
Node cha: BODY
Tên Node con đầu tiên: H1
Nội dung Node con đầu tiên: DOM node
Tên Node con thứ 2: P
Nội dung Node con thứ 2: tìm hiểu DOM NODE HTML

DOM NODE-Thêm phần tử mới

Trước tiên tạo phần tử (node), sau đó nối nó vào phần tử hiện có bằng **element.appendChild(Nút cần thêm)**;.

```
<body>
    <div id="domNode">
        <h1 id="h">< h1 >DOM node< /h1 ></h1>
        <p id="p1">tìm hiểu DOM NODE HTML</p>
        <a href="https://codelearn.io">Trang codelean</a>
    </div>
    <br />
    <button type="button" onclick="themNodeMoi();">Thêm node</button>
    <script type="text/javascript" language="javascript">
        /*
            Tạo các phần tử HTML mới (Nút) - Creating New HTML Elements (Nodes)
            Để thêm một phần tử mới vào HTML DOM, trước tiên bạn phải tạo phần tử (nút phần tử),
            sau đó nối nó vào phần tử hiện có.
        */
        function themNodeMoi(){
            var themNode=document.createElement("h2");
            var txtNode=document.createTextNode("Đây là thẻ h2 mới thêm vào bên trong thẻ DIV");
            themNode.appendChild(txtNode);
            var nodeDiv=document.getElementById("domNode");
            nodeDiv.appendChild(themNode);
        }
```

DOM NODE-Chèn phần tử mới

Sử dụng phương insertBefore() để chèn 1 nút vào trước 1 nút có sẵn

```
<div id="domNode">
    <h1 id="h">< h1 >DOM node< /h1 ></h1>
    <a href="https://codelearn.io">Trang codelean</a>
</div>
<br />
<button type="button" onclick="chenNodeMoi();">Chèn node moi</button>

<script type="text/javascript" language="javascript">
    /*
        Tạo các phần tử HTML mới - InsertBefore()
        Sử dụng phương pháp: insertBefore(Node thay thế , node bị thay thế)
    */
    function chenNodeMoi(){
        var themNode=document.createElement("h2");
        var txtNode=document.createTextNode("Đây là thẻ p mới chèn vào trước thẻ p đã có");
        themNode.style.color="blue";
        themNode.appendChild(txtNode);
        var nodeDiv=document.getElementById("domNode");
        var nodeBiChen=document.getElementById("p1");
        nodeDiv.insertBefore(themNode,nodeBiChen);
    }
```

< h1 >DOM node< /h1 >

tìm hiểu DOM NODE HTML

Trang codelean

Chèn node moi

< h1 >DOM node< /h1 >

Đây là thẻ p mới chèn vào trước thẻ p đã có

tìm hiểu DOM NODE HTML

Trang codelean

Chèn node moi

DOM NODE-Xóa phần tử

Sử dụng phương node.remove() để xóa 1 có sẵn

```
<div id="domNode">
    <h1 id="h">< h1 >DOM node< /h1 ></h1>
    <p id="p1">tìm hiểu DOM NODE HTML</p>
    <a href="https://codelearn.io">Trang codelean</a>
</div>
<br />
<button type="button" onclick="xoaNode1();">Xóa Node 1</button>

<script type="text/javascript" language="javascript">
    /*
        Xóa các phần tử HTML hiện có
        Để xóa phần tử HTML, sử dụng phương thức remove() trong các trình duyệt mới
        Ví dụ xóa thẻ p
    */
    function xoaNode1(){
        var nodeBiXoa1=document.getElementById("p1");
        nodeBiXoa1.remove();
    }
```

< h1 >DOM node< /h1 >

tìm hiểu DOM NODE HTML

Trang codelean

Xóa Node 1

< h1 >DOM node< /h1 >

Trang codelean

Xóa Node 1

DOM NODE-Xóa Node con

Đối với các trình duyệt không hỗ trợ phương thức remove(), phải tìm nút cha để xóa một phần tử: Node_cha.removeChild(Tên Node con)

```
<div id="domNode">
    <h1 id="h">< h1 >DOM node< /h1 ></h1>
    <p id="p1">tìm hiểu DOM NODE HTML</p>
    <a href="https://vudinhthang.io">Trang Vu Dinh Thang</a>
</div>
<br />
<button type="button" onclick="xoaNode2();">Xóa Node 2</button>
<script type="text/javascript" language="javascript">
    /*
        Xóa các phần tử HTML hiện có
        Sử dụng phương thức removeChild() trong các trình duyệt mới
        Ví dụ xóa thẻ h1
    */
    function xoaNode2(){
        var nodeCha=document.getElementById("domNode");
        var xoaNodeH1=document.getElementById("h");
        nodeCha.removeChild(xoaNodeH1);
    }
```

< h1 >DOM node< /h1 >

tìm hiểu DOM NODE HTML

Trang Vu Dinh Thang

Xóa Node 2

tìm hiểu DOM NODE HTML

Trang Vu Dinh Thang

Xóa Node 2

DOM NODE-Thay thế 1 Node

```html
<div id="domNode">
    <h1 id="h">< h1 >DOM node< /h1 ></h1>
    <p id="p1">tìm hiểu DOM NODE HTML</p>
    <a href="https://vudinhthang.io">Trang Vu Dinh Thang</a>
</div>
<br />
<button type="button" onclick="thayTheNode();">Thay Thế Node</button>
<script type="text/javascript" language="javascript">
    /*
        Thay thế 1 phần tử HTML cũ bằng 1 phần tử HTML mới. replaceChild(Node cũ, Node mới);
        Để thêm một phần tử mới vào HTML DOM, trước tiên bạn phải tạo phần tử (nút phần tử),
        sau đó thay thế nó cho phần tử hiện có.
    */
    function thayTheNode(){
        var nodeMoi=document.createElement("h2");
        var txtNode=document.createTextNode("Đây là thẻ h2 mới để thay thể thẻ p đã có");
        nodeMoi.style.color="red";
        nodeMoi.appendChild(txtNode);
        var nodeDiv=document.getElementById("domNode");
        var nodePCu=document.getElementById("p1");
        nodeDiv.replaceChild(nodeMoi,nodePCu);
    }
```

< h1 >DOM node< /h1 >

tìm hiểu DOM NODE HTML

Trang Vu Dinh Thang

Thay Thế Node

< h1 >DOM node< /h1 >

Đây là thẻ h2 mới để thay thể thẻ p đã có

Trang Vu Dinh Thang

Thay Thế Node

DOM NODE-Thay thế 1 Node

Đối với các trình duyệt không hỗ trợ phương thức remove(), phải tìm nút cha để xóa một phần tử: Node_cha.removeChild(Tên Node con)

ĐẠI NAM
UNIVERSITY

Lời cảm ơn!

HỌ VÀ TÊN NGƯỜI CẢM ƠN

Chức danh

Thông tin liên hệ (nếu cần)