

Dom HTML với Javascript

Một số đối tượng trong Javascript

- 1. Giới thiệu về đối tượng
- 2. Đối tượng của Browser
- 3. Đối tượng của JavaScript
- 4. Đối tượng của HTML

Giới thiệu về đối tượng

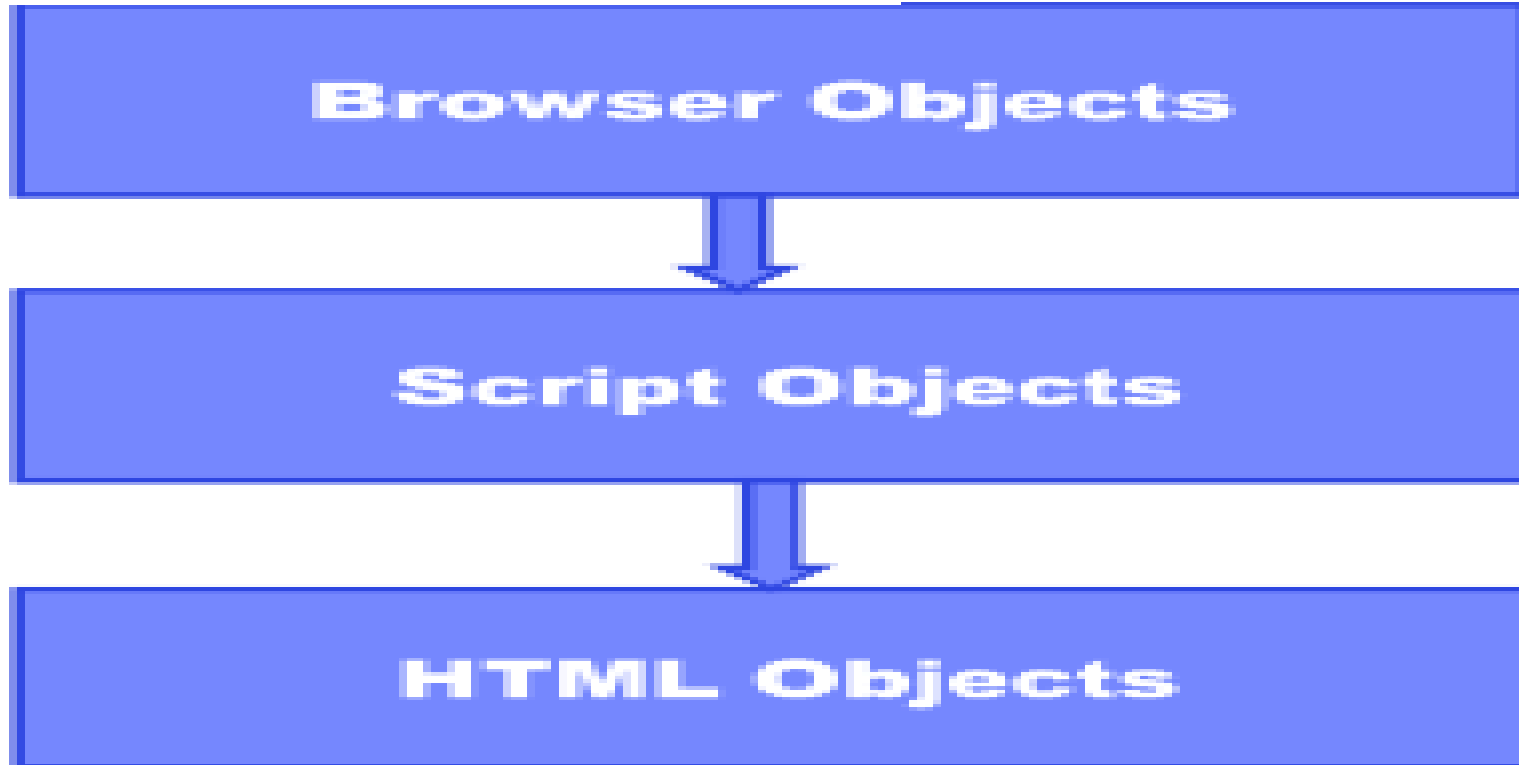
- Đối tượng là một thực thể trong thế giới thực được mô hình hóa
- Đối tượng = Dữ liệu + Phương thức



Giới thiệu về đối tượng

- JavaScript là ngôn ngữ lập trình hướng đối tượng
- Khi tạo trang Web, chúng ta có thể sử dụng các đối tượng được cung cấp bởi browser, JavaScript, HTML
- Truy nhập
 - Thuộc tính: *tên_đối_tượng . tên_thuộc_tính*
 - Phương thức: *tên_đối_tượng . tên_phương_thức()*

Giới thiệu về đối tượng



Phân tầng các đối tượng

Giới thiệu về đối tượng

- Khởi tạo đối tượng

```
tên_đối_tượng = new  
    đối_tượng(danh_sách_tham_số)
```

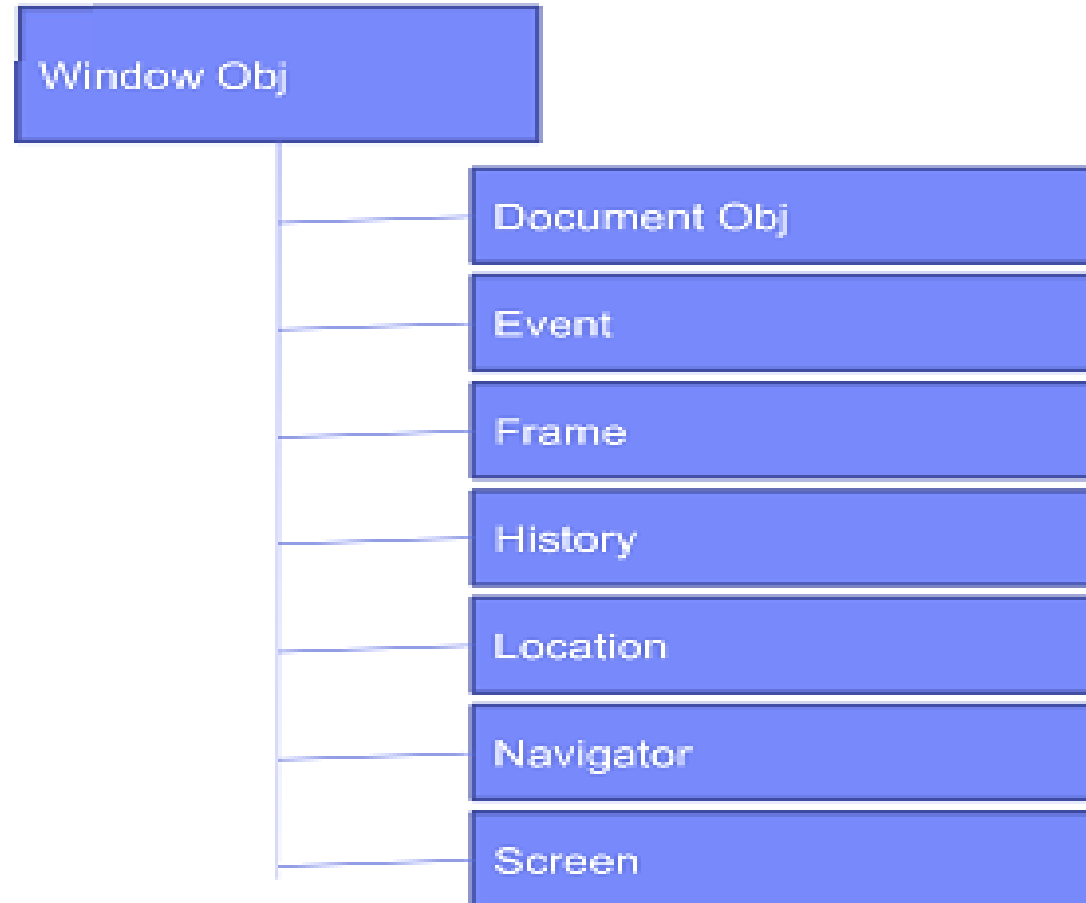
- Cấu trúc for..in và with

- Cú pháp

```
for (variable in object)  
    lệnh;
```

Đối tượng của browser

- IE Objects



Đối tượng của window

- Mỗi thành phần trên trang web được xem như một đối tượng,
- Tất cả các đối tượng trong Javascript được tổ chức phân cấp dạng hình cây gọi là mô hình DOM (Document Object Model),
- Ở mức trên cùng là đối tượng **window** biểu thị cho khung hay cửa sổ của trình duyệt, các phần tử còn lại là đối tượng con của window
- Để truy xuất đến một đối tượng trong IE và thay đổi dữ liệu cho chính phần tử đó dùng thuộc tính ID , tất cả các đối tượng trên trang đều có một ID duy nhất.

Đối tượng của window

- Đối tượng Window
 - Thể hiện cửa sổ của trình duyệt
 - Được sử dụng để
 - nhận các thông tin về trạng thái của cửa sổ
 - hiển thị các đối tượng khác
 - truy cập các sự kiện

Đối tượng của window

- document Obj: biểu diễn tài liệu HTML
- event: biểu diễn trạng thái của sự kiện
- frame: biểu diễn các frame (child windows)
- history: thông tin về URLs đã mở
- location: thông tin về URL hiện tại
- navigator: thông tin về Web browser
- screen: thông tin về màn hình client, khả năng render

Đối tượng của window

Ví dụ:

```
window.defaultStatus="String";
```

```
window.status="String";
```

```
window.location="URL";
```

Ta cũng có thể mở một trang web mới bằng lệnh:

```
window.location.href= "URL";
```

Đối tượng của window

Phương Thức	Mô tả
<code>alert("msg")</code>	Hiển Thị hộp thoại thông báo
<code>blur()</code>	Di chuyển con trỏ đến cửa sổ hiện hành
<code>clearInterval(ID)</code>	Hủy thời gian đã thiết lập bằng <code>setInterval()</code>
<code>clearTimeout(ID)</code>	Hủy thời gian đã thiết lập bằng <code>setTimeout()</code>
<code>close()</code>	Đóng cửa sổ hiện hành
<code>confirm("msg")</code>	Hiển thị hộp thoại xác nhận với hai nút Cancel và OK
<code>focus()</code>	Đưa con trỏ về cửa sổ hiện hành
<code>MoveBy(x,y)</code>	Di chuyển cửa sổ đến một vị trí mới một đoạn pixel so với cửa sổ hiện hành
<code>MoveTo(x,y)</code>	Di chuyển cửa sổ qua trái và lên trên một đoạn pixel cụ thể so với cửa sổ hiện hành.

Đối tượng của window

<code>open(URL,"windowname", "FeatureList")</code>	URL : đc trang web nạp vào cửa sổ. WindowName: là tên cửa sổ . FeatureList : danh sách các thuộc tính của cửa sổ: toolbars, menu, status
<code>print()</code>	in nội dung trong cửa sổ hiện hành.
<code>prompt("msg","reply")</code>	Hiển thị hộp thoại nhập liệu
<code>setTimeout(func,millisec)</code>	Thiết lập thời gian mili giây để gọi 1 hàm
<code>stop()</code>	Hủy việc download một cửa sổ. Tương tự như việc đóng một cửa sổ trình duyệt.
<code>resizeBy(dx,dy)</code>	Thay đổi kích thước cửa sổ sang phải dx, dưới dy pixel
<code>resizeTo(x,y)</code>	Thay đổi kích thước x, y pixel
<code>scrollBy(dx,dy)</code>	cuộn nội dung sang phải dx, xuống dưới dy pixel

Đối tượng của window

Ví dụ: `Objwindow.close()`

❖ **Từ khoá self**: trong trường hợp muốn thao tác trên cửa sổ hiện hành ta dùng từ khoá **self** thay thế cho đối tượng **window**

Ví dụ : đóng cửa sổ hiện hành:

Self.close() hoặc **window.close()**

Đối tượng của Document

- Đối tượng document cung cấp các thuộc tính và phương thức để làm việc với toàn bộ tài liệu hiện hành gồm: form, liên kết, hình ảnh, tiêu đề, vị trí hiện hành, màu hiện hành...
- Đối tượng document được định nghĩa khi tag body được xử lý trong trang HTML và nó vẫn tồn tại nếu trang được nạp.
- Các thuộc tính của document phản ánh thuộc tính của tag body.
 - Trong body có 2 sự kiện OnLoad và Unload

Đối tượng của Document

■ Thuộc tính

Thuộc tính	Mô tả
alinkcolor	Thiết lập hoặc trả về giá trị màu của liên kết đang xem của tài liệu
bgcolor	Thiết lập hoặc trả về giá trị màu nền của tài liệu
cookie	Chứa giá trị các cookies dành cho tài liệu hiện hành
domain	Trả về giá trị tên miền máy chủ chứa document
fgcolor	Thiết lập hoặc trả về giá trị màu chữ của tài liệu
lastmodified	Trả về giá trị ngày giờ cuối cùng mà tài liệu được cập nhật

Đối tượng của Document

■ Thuộc tính

linkcolor	Thiết lập hoặc trả về giá trị màu của liên kết trong tài liệu
location	mở một trang web mới
referrer	Returns the URL of the document that loaded the current document
title	Trả về giá trị của tựa đề của tài liệu
url	Trả về đường dẫn củ tài liệu hiện hành
vlinkcolor	Thiết lập hoặc trả về giá trị màu của liên kết đã

Đối tượng của Document

■ Phương thức

Phương thức	Mô tả
<code>clear()</code>	Xóa tài liệu
<code>close()</code>	Đóng một tài liệu
<code>focus()</code>	Đưa trở về một đối tượng trong trang
<code>open(["mimetype"])</code>	Mở một stream để thu thập dữ liệu và thực hiện các phương thức <code>write</code> và <code>writeln</code> .
<code>write("str")</code>	viết một chuỗi vào một tài liệu
<code>writeln("str")</code>	viết một chuỗi vào một tài liệu và xuống dòng

Đối tượng History

- Cung cấp các URLs được mở gần nhất
- Phương thức
 - back: quay lại URL ngay trước
 - go(number): quay lại vị trí xác định trong tài liệu
 - forward: mở URL tiếp theo

Đối tượng HTML Dom

- DOM = Document Object Model
- Là tập hợp các đối tượng HTML chuẩn được dùng để **truy xuất** và **thay đổi thành phần HTML** trong trang web (thay đổi nội dung tài liệu của trang)
- Một số đối tượng của DOM: window, document, history, link, form, frame, location, event, ...

Đối tượng window - Dom

■ **Properties**

- document
- event
- history
- location
- name
- navigator
- screen
- status

● **Methods**

- alert
- confirm
- prompt
- blur
- close
- focus
- open

Đối tượng window - Dom

■ Ví dụ:

```
<html>
  <body>
    <script type="text/javascript">
      var curURL = window.location;
      window.alert(curURL);
    </script>
  </body>
</html>
```



Đối tượng Document - Dom

- Biểu diễn cho **nội dung trang HTML** đang được hiển thị trên trình duyệt
- Dùng để lấy thông tin về tài liệu, các thành phần HTML và xử lý sự kiện



Đối tượng Document - Dom

● Properties	● Methods
<ul style="list-style-type: none">➤ aLinkColor➤ bgColor➤ body➤ fgColor➤ linkColor➤ title➤ URL➤ vlinkColor➤ forms[]➤ images[]➤ childNodes[]	<ul style="list-style-type: none">➤ close➤ open➤ createTextNode(" text ")➤ createElement("HTMLtag")➤ getElementById("id")➤ ...
<ul style="list-style-type: none">➤ documentElement➤ cookie➤	

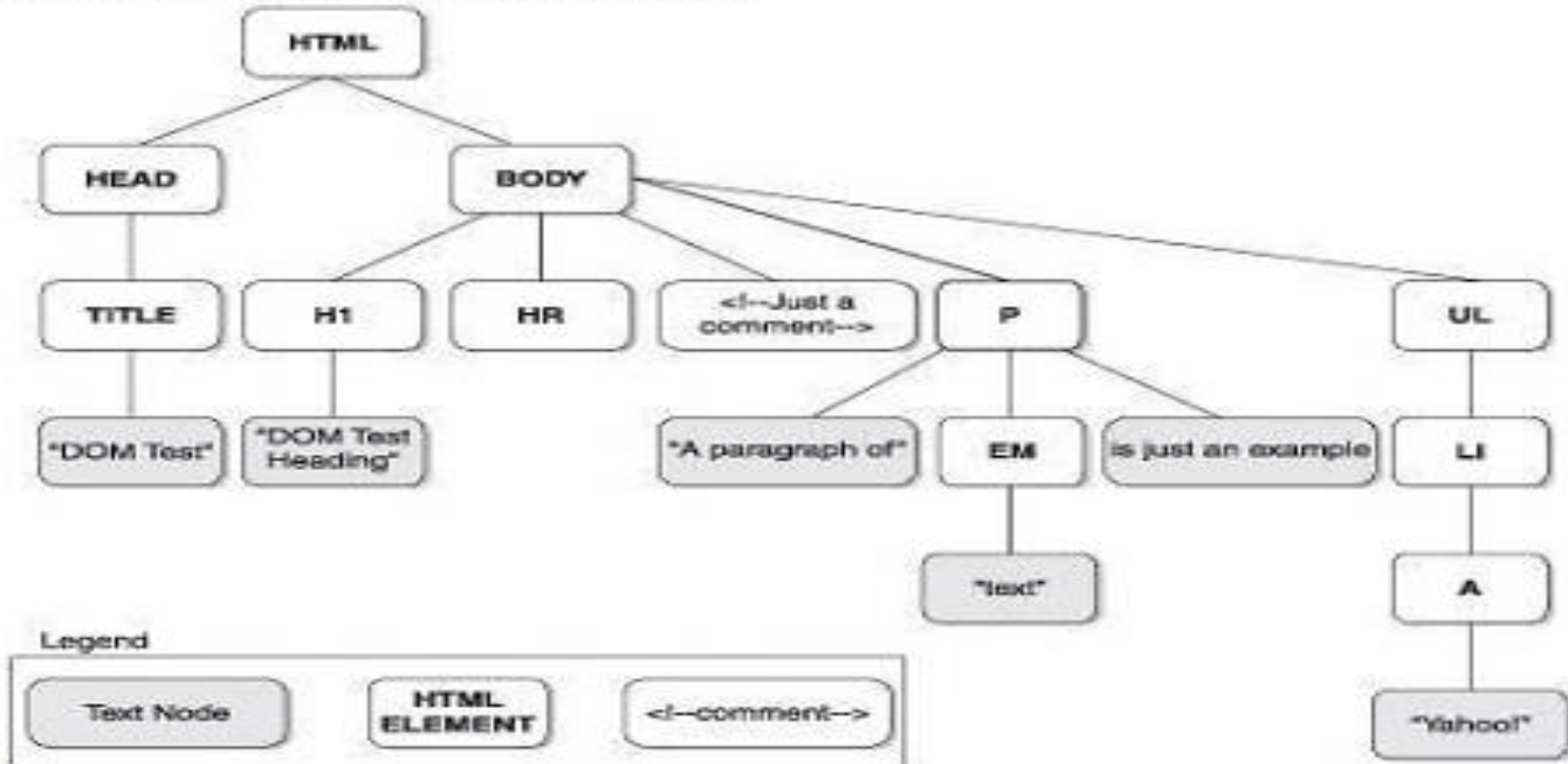
Đối tượng Document - Dom

- Biểu diễn nội dung của tài liệu theo cấu trúc cây

```
<html>
  <head>
    <title>DOM Test</title>
  </head>
  <body>
    <h1>DOM Test Heading</h1>
    <hr />
    <!-- Just a comment -->
    <p id="p1" >A paragraph of <em>text</em>
      is just an example</p>
    <ul>
      <li>
        <a href="http://www.yahoo.com" > Yahoo!
        </a>
      </li>
    </ul>
  </body>
```

Đối tượng Document - Dom

- Cấu trúc cây nội dung tài liệu



Đối tượng Document - Dom

■ Các loại DOM Node chính

Node Type Number	Loại	Mô tả	Vi dụ
1	Element	HTML or XML element	<code><p>...</p></code>
2	Attribute	Thuộc tính của HTML hay XML element	<code>align="center"</code>
3	Text	Nội dung chứa trong HTML or XML element	<code>This is a text fragment!</code>
8	Comment	HTML comment	<code><!-- This is a comment --></code>
9	Document	Đối tượng tài liệu gốc, thường là element nằm ở cấp cao nhất trong cây cấu trúc của tài liệu	<code><html></code>
10	DocumentType	Định nghĩa loại tài liệu	<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>


Đối tượng Document - Dom

- `getElementById (id1)`

Trả về node có giá trị thuộc tính `id = id1`

Ví dụ:

```
//<p id="id1" >  
//    some text  
//</p>
```



Text Node

```
var node = document.getElementById("id1");  
var nodeName = node.nodeName; // p  
var nodeType = node.nodeType; // 1  
var nodeValue = node.nodeValue; // null  
var text      = node.innerText; // some text
```

Đối tượng Document - Dom

- createElement (nodeName)

Cho phép tạo ra 1 node HTML mới tùy theo đối số nodeName đầu vào

Ví dụ:

```
var imgNode = document.createElement("img");  
imgNode.src = "images/test.gif";  
  
// 
```

Đối tượng Document - Dom

■ createTextNode (content)

Ví dụ:

```
var textNode = document.createTextNode("New  
text");  
var pNode = document.createElement("p");  
pNode.appendChild(textNode);  
  
// <p>New text</p>
```

Đối tượng Document - Dom

- appendChild (newNode)

Chèn node mới **newNode** vào cuối danh sách các node con của một node.

Ví dụ:

```
//<p id="id1" >
//    some text
//</p>
var pNode = document.getElementById("id1");
var imgNode = document.createElement("img");
imgNode.src = "images/test.gif";
pNode.appendChild(imgNode);

//<p id="id1" >
//    some text
//</p>
```

Đối tượng Document - Dom

- innerHTML

Chỉ định nội dung HTML bên trong một node.

Ví dụ:

```
//<p id="para1" >  
// some text  
//</p>
```

```
var theElement = document.getElementById("para1");  
theElement.innerHTML = "Some <b> new </b> text";
```

```
// Kết quả :  
// <p id="para1" >  
// Some <b> new <b/> text  
// </p>
```


Đối tượng Document - Dom

- innerText

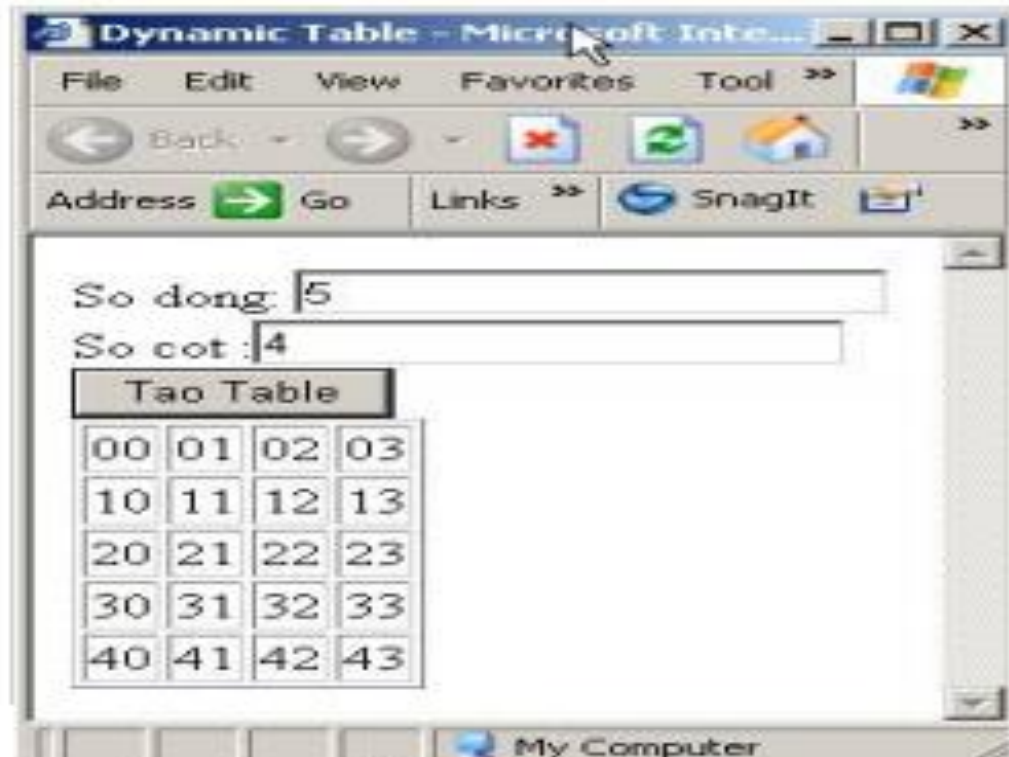
Tương tự innerHTML, tuy nhiên bất kỳ nội dung nào đưa vào cũng được xem như là text hơn là các thẻ HTML.

Ví dụ:

```
var theElement = document.getElementById("para1");  
theElement.innerText = "Some <b> new </b> text";  
// Kết quả hiển thị trên trình duyệt  
// bên trong thẻ p: "Some <b> new </b> text"
```

Ví dụ: Dynamic table

- Viết trang web cho phép tạo table có số dòng, số cột do người dùng nhập vào.



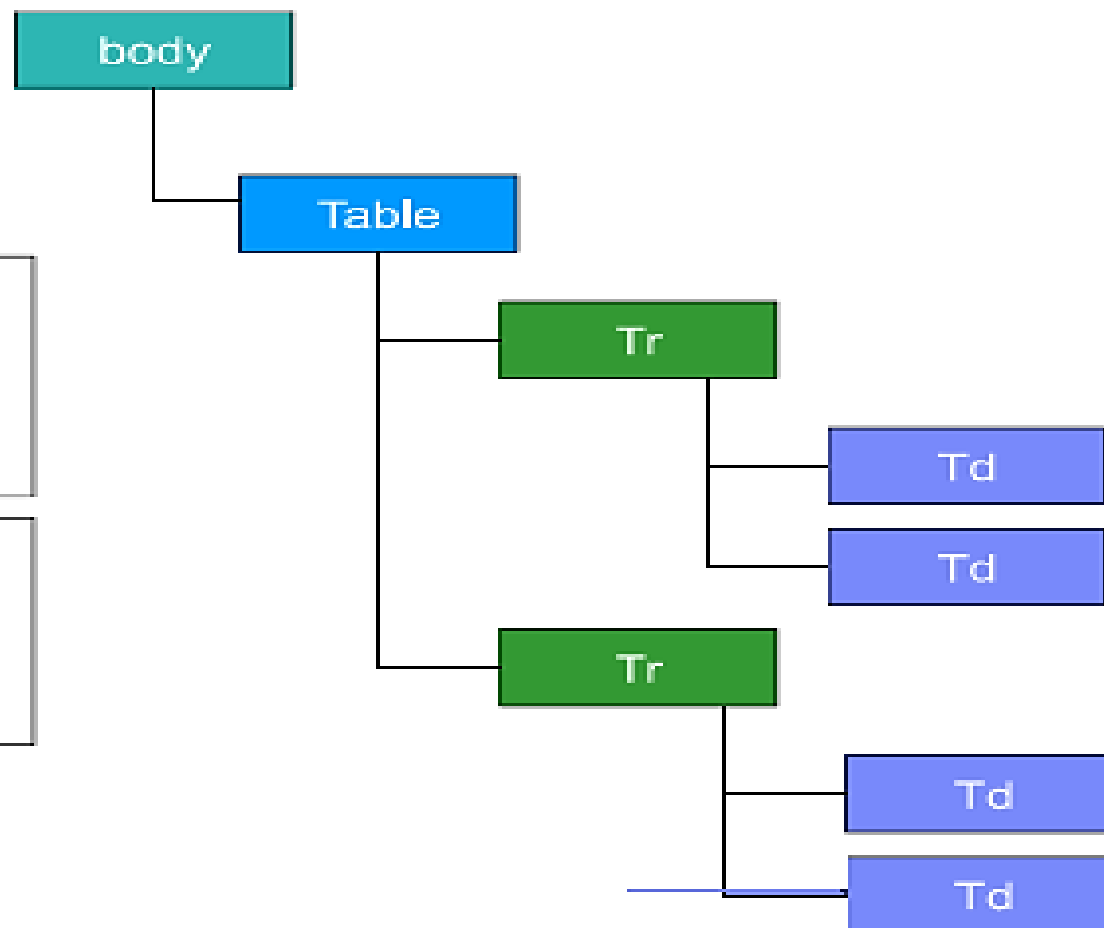
Ví dụ: Dynamic table

`<Table>`

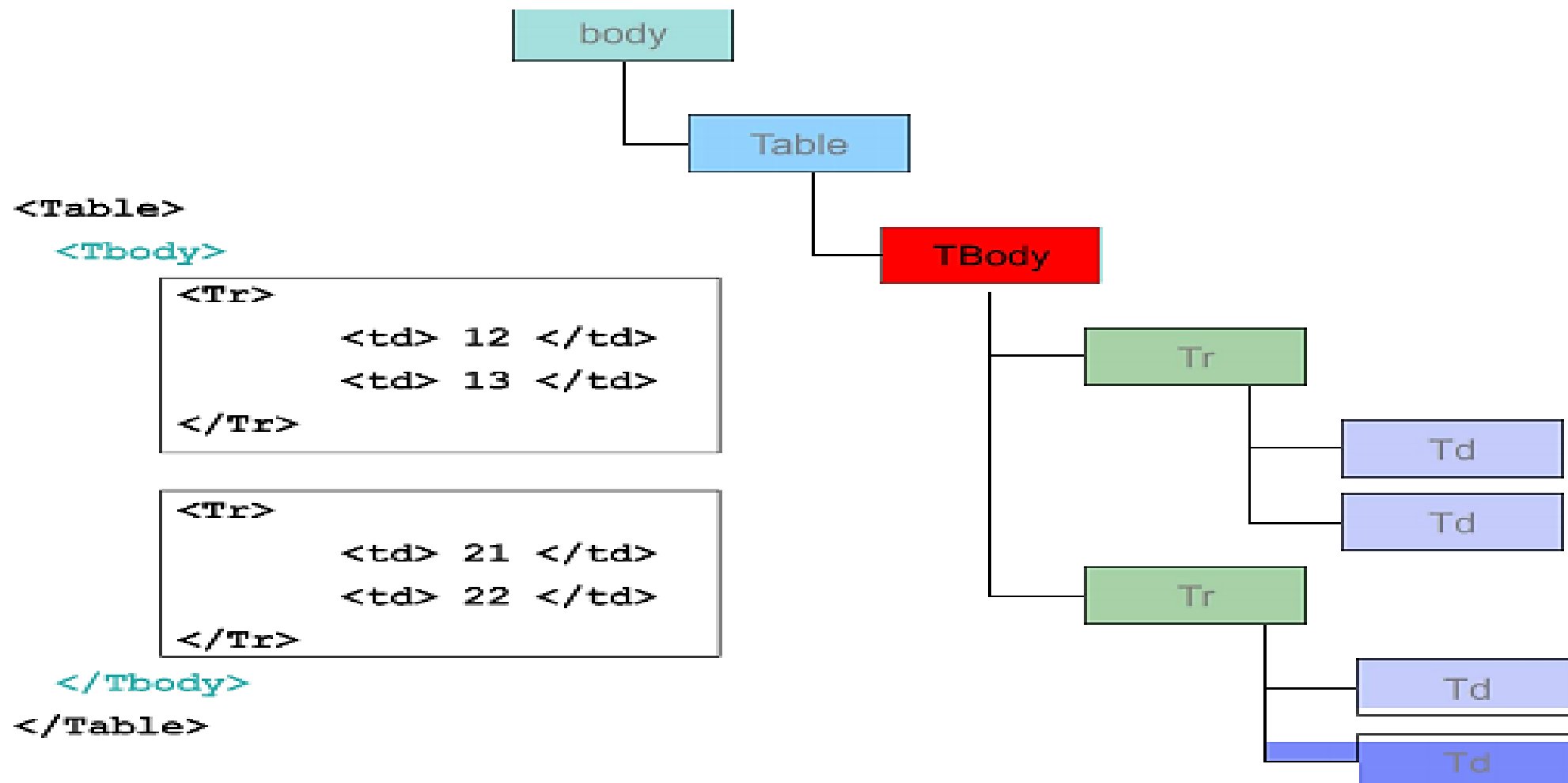
<code><tr></code>
<code><td> 12 </td></code>
<code><td> 13 </td></code>
<code></tr></code>

<code><tr></code>
<code><td> 21 </td></code>
<code><td> 22 </td></code>
<code></tr></code>

`</Table>`



Ví dụ



Ví dụ: Dynamic table

- `Document.createElement(...)` :Tạo một đối tượng thẻ DOM HTML
- `Object.appendChild(...)`: Thêm một đối tượng thẻ DOM HTML như là nút con.

```
function CreateTable(divTable)
{
    var tagTable = document.createElement("table");
    tagTable.border = 1;
    var tagTBody = document.createElement("tbody");
    tagTable.appendChild(tagTBody);

    var nDong = txtSoDong.value;
    var nCot = txtSoCot.value;

    for (i=0; i<nDong; i++)
    {
        var tagTR = document.createElement("tr");
        for (j=0; j<nCot; j++)
        {
            var tagTD = document.createElement("td");
            var textNode = document.createTextNode(i+" "+j);
            tagTD.appendChild(textNode);

            tagTR.appendChild(tagTD);
        }

        tagTBody.appendChild(tagTR);
    }

    divTable.appendChild(tagTable);
}
```

Đối tượng Javascript

- 1. Đối tượng String
- 2. Đối tượng Date
- 3. Đối tượng Math
- 4. Đối tượng Array

Đối tượng String

- Được sử dụng để làm việc với chuỗi văn bản
- Khởi tạo

```
var tên_xâu = "nội_dung_xâu";
```

```
tên_xâu="nội_dung_xâu";
```

```
var tên_xâu = String("nội_dung_xâu");
```


Đối tượng String

- Thuộc tính
 - **length**: độ dài xâu
- Phương thức
 - **big**: tăng kích thước xâu
 - **small**: giảm kích thước xâu
 - **toLowerCase**: chuyển thành chữ thường
 - **toUpperCase**: chuyển thành chữ hoa
 - **fontcolor**: xác định màu chữ
 - **charAt**: vị trí kí tự trong xâu

Đối tượng Date

- Chứa các thông tin về ngày, giờ
- Lưu trữ thời điểm thông qua số lượng ms từ ngày 1/1/1970 tại 00:00:00

- Khởi tạo

```
tên_đối_tượng = new Date(tham_số);
```

- Ví dụ

```
today    = new Date();
```

```
dateobj = new Date("Octorber  
13,2009,10:10:10");
```

Đối tượng Date

- Phương thức chia làm:

get: lấy giá trị ngày và giờ

set: thiết lập giá trị ngày và giờ

Bao gồm: Hours, Minutes, Seconds, Time, Year, Month, Date

- Giá trị

- seconds và minutes: 0-59

- hours: 0-23

- day: 0-6

- date: 0-31

- months: 0-11

Đối tượng Math

- Thuộc tính
 - PI: số Pi, xấp xỉ 3.1415
 - E: số e, xấp xỉ 2.718
- Phương thức
 - abs: lấy giá trị tuyệt đối
 - sin, cos, tan
 - min (number1, number2)
 - max (number1, number2)
 - sqrt
 - pow

Đối tượng Array

■ Khởi tạo

```
tên_mảng = new Array(kích_thuớc_mảng);  
hoặc  
tên_mảng = new Array(ds_phần_tử);
```

Ví dụ: mybook = new Array(2);
 mybook[0] = "Access 2000";
 mybook[1] = "Programming Techniques";

■ Phương thức

- reverse: đảo ngược các phần tử trong mảng
- pop: trả về phần tử cuối cùng của mảng, rồi xóa nó
- push: thêm một phần tử vào cuối mảng

Đối tượng Array

```
<head>
  <script language="javascript">
    color = new Array("red","blue","green");
    var scolor;
    for(var elem in color){
      scolor = "color " + elem + " = " +
        color[elem] + "<br>";
      document.write(scolor);
    }
  </script>
</head>
```

