JAVASCRIPT & DOM HTML

Nội dung

- Giới thiệu về Javascript
- Nhúng Javascript vào trang web
- Kiểu dữ liệu & cú pháp trong Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

Nội dung

- Giới thiệu về Javascript
- Nhúng Javascript vào trang web
- Kiểu dữ liệu & Các cú pháp Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

Giới thiệu về Javascript

- Là ngôn ngữ Client-side script hoạt động trên trình duyệt của người dùng (client)
- Chia sẻ xử lý trong ứng dụng web. Giảm các xử lý không cần thiết trên server.
- Giúp tạo các hiệu ứng, tương tác cho trang web.

Giới thiệu về Javascript

- Client-Side Script:
  - Script được thực thi tại Client-Side (trình duyệt): Thực hiện các tương tác với người dùng (tạo menu chuyển động, …) , kiểm tra dữ liệu nhập, …
- Server-Side Script:
  - Script được xử lý tại Server-Side, nhằm tạo các trang web có khả năng phát sinh nội dung động. Một số xử lý chính: kết nối CSDL, truy cập hệ thống file trên server, phát sinh nội dung html trả về người dùng…

Giới thiệu về Javascript

- Khi trình duyệt (Client browser) truy cập trang web có chứa các đoạn mã xử lý tại server-side. Server (run-time engine) sẽ thực hiện các lệnh Server-side Scipts và trả về nội dung HTML cho trình duyệt

  Nội dung html trả về chủ yếu bao gồm: mã html, client-script.

Nội dung

- Giới thiệu về Javascript
- Nhúng Javascript vào trang web
- Kiểu dữ liệu & Các cú pháp Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

Nhúng Javascript vào trang web

Định nghĩa script trực tiếp trong trang html:

```
<script type="text/javascript">
    <!--
   // Lệnh Javascript
     -->
</script>
```

Nhúng sử dụng script cài đặt từ 1 file .js khác:

```
<script src="xxx.js"></script>
```

Nhúng Javascript vào trang web

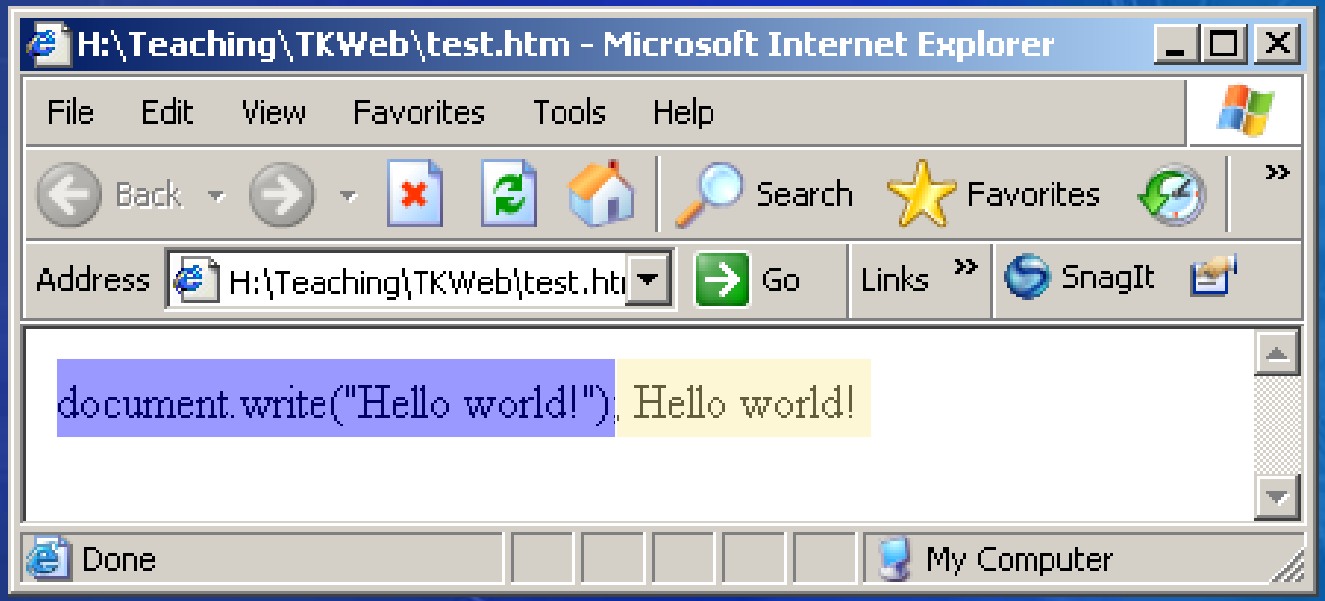
```
<html>
    <head>
      <script type="text/javascript">
        some javascript statements
      </script>
      <script type="text/javascript">
          some statements
      </script>
    </head>
    <body>
      <script type="text/javascript">
          some statements
      </script>
      <script src="Tên_file_script.js">method()</script>
      <script type="text/javascript">
      // goi thực hiện các phương thức được định nghĩa
      // trong "Tên_file_script.js"
      </script>
    </body>
</html>
```

Nhúng Javascript vào trang web

- Đặt giữa tag <head> và </head>: script sẽ thực thi ngay khi trang web được mở.
- Đặt giữa tag <body> và </body>: script trong phần body được thực thi khi trang web đang mở (sau khi thực thi các đoạn script có trong phần <head>).
- Số lượng đoạn client-script chèn vào trang không hạn chế.

VD: Nhúng Javascript vào trang web

```
<html>
    <body>
        document.write(“Hello world!”);
        <script type="text/javascript">
            document.write(“Hello world!”);
        </script>
    </body>
</html>
```



Nội dung

- Giới thiệu về Javascript
- Nhúng Javascript vào trang web
- Kiểu dữ liệu & Các cú pháp Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

Biến số trong Javascript

- Cách đặt tên biến
  - Bắt đầu bằng một chữ cái hoặc dấu _
  - A..Z,a..z,0..9,_ : phân biệt HOA, Thường
- Khai báo biến
  - Sử dụng từ khóa **var**

    Ví dụ: **var count=10,amount;**
  - Không cần khai báo biến trước khi sử dụng, biến thật sự tồn tại khi bắt đầu sử dụng lần đầu

Kiểu dữ liệu trong Javascript

Kiểu dữ liệu	Ví dụ	Mô tả
Object	var listBooks = new Array(10) ;	Trước khi sử dụng, phải cấp phát bằng từ khóa new
String	“The cow jumped over the moon.” “40”	Chứa được chuỗi unicode Chuỗi rỗng “”
Number	0.066218 12	Theo chuẩn IEEE 754
boolean	true / false	
undefined	var myVariable ;	myVariable = undefined
null	connection.Close();	connection = null

1 Biến trong javascript có thể lưu **bất kỳ kiểu dữ liệu** nào.

Đổi kiểu dữ liệu

- Biến tự đổi kiểu dữ liệu khi giá trị mà nó lưu trữ thay đổi

Ví dụ:

```
var x = 10;                 // x kiểu Number
x = “hello world !”;  // x kiểu String
```

- Có thể cộng 2 biến khác kiểu dữ liệu

Ví dụ:

```
var x;
x = “12” + 34.5;  // KQ: x = “1234.5”
```

- Hàm parseInt(…), parseFloat(…) : Đổi KDL từ chuỗi sang số.

Hàm trong Javascript

- Dạng thức khai báo chung:

```
function Tên_hàm(thamso1, thamso2,..)
{
  ……
}
```

- Hàm có giá trị trả về:

```
function Tên_hàm(thamso1, thamso2,..)
{
  ……
  return (value);
}
```

Hàm trong Javascript

- Ví dụ:

```
function Sum(x, y)
{
  tong = x + y;
  return tong;
}
```

- Gọi hàm:

```
var x = Sum(10, 20);
```

Các quy tắc chung

- Khối lệnh được bao trong dấu {}
- Mỗi lệnh nên kết thúc bằng dấu ;
- Cách ghi chú thích:
  - // Chú thích 1 dòng
  - /* Chú thích
    nhiều dòng */

Câu lệnh if

```
if (condition)
{
  statement[s] if true
}
else
{
  statement[s] if false
}
```

Ví dụ:

```
var x = 5, y = 6, z;
if (x == 5) { if (y == 6) z = 17; } else z = 20;
```

Câu lệnh switch

```
switch (expression)
{
    case label :
     statementlist
    case label :
      statementlist
        ...
  default :
     statement list
}
```

Ví dụ :

```
var diem = “G”;
switch (diem) {
        case “Y”:
                document.write(“Yếu");
                break;
        case “TB”:
                document.write(“Trung bình");
                break;
        case “K”:
                document.write(“Khá");
                break;
        case “G” :
                document.write(“Giỏi");
                break;
        default:
                document.write(“Xuất sắc")
}
```

Vòng lặp for

```
for ([initial expression]; [condition];
    [update expression]) {
    statement[s] inside loop
}
```

Ví dụ:

```
var myarray = new Array();
for (i = 0; i < 10; i++)
{
        myarray[i] = i;
}
```

Vòng lặp while

```
while (expression)
{
    statements
}
```

Ví dụ:

```
var i = 9, total = 0;
while (i < 10)
{
        total += i * 3 + 5;
        i = i +5;
}
```

Vòng lặp do.. while

```
do
    {
    statement
}while (expression);
```

Ví dụ:

```
var i = 9, total = 0;
do
{
        total += i * 3 + 5;
        i = i +5;
}  while (i > 10);
```

Nội dung

- Giới thiệu về Javascript
- Nhúng Javascript vào trang web
- Kiểu dữ liệu & Các cú pháp Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

Các sự kiện thông dụng

- Các sự kiện được hỗ trợ bởi hầu hết các đối tượng
  - onClick
  - onFocus
  - onChange
  - onBlur
  - onMouseOver
  - onMouseOut
  - onMouseDown
  - onMouseUp
  - onLoad
  - onSubmit
  - onResize
  - .....

Xử lý sự kiện cho các thẻ HTML

- Cú pháp 1:

```
<TAG eventHandler = "JavaScript Code">
```

- Ví dụ:

```
<body>
    <INPUT TYPE="button“ NAME="Button1“
    VALUE="OpenSesame!”
    onClick="window.open('mydoc.html');">
</body>
```

- Lưu ý: Dấu “…” và ‘…’

Xử lý sự kiện bằng function

```
<head>
    <script language="Javascript">
      function GreetingMessage()
      {
            window.alert("Welcome to my world");
      }
    </script>
</head>

<body onload="GreetingMessage()">
</body>
```

Xử lý sự kiện bằng thuộc tính

- Gán tên hàm xử lý cho 1 object event

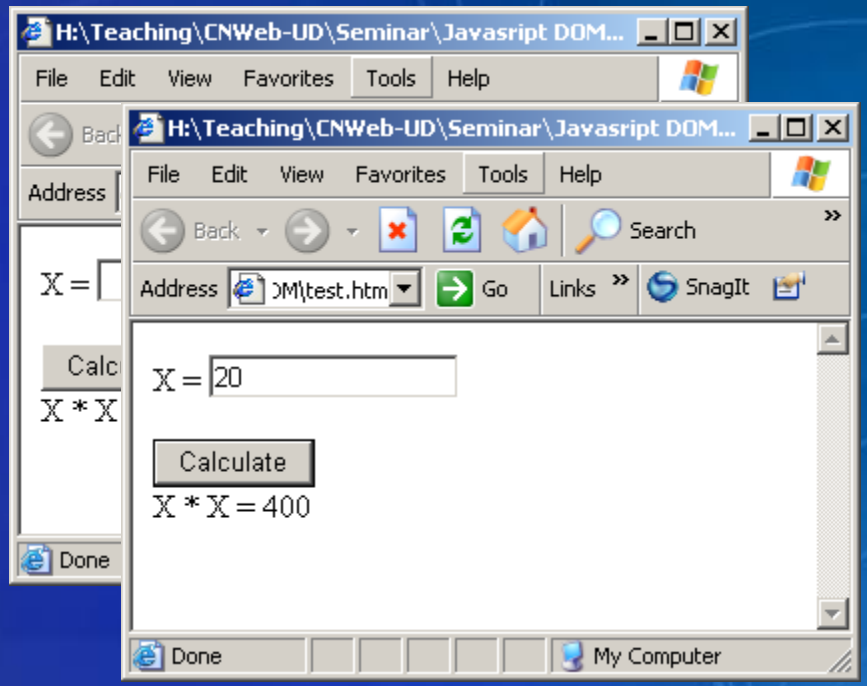
  `object.eventhandler = function_name;`

- Ví dụ:

```
<head>
    <script language="Javascript">
       function GreetingMessage()
       {
              window.alert("Welcome to my world");
       }

       window.onload = GreetingMessage ()
    </script>
</head>

<body>
</body>
```

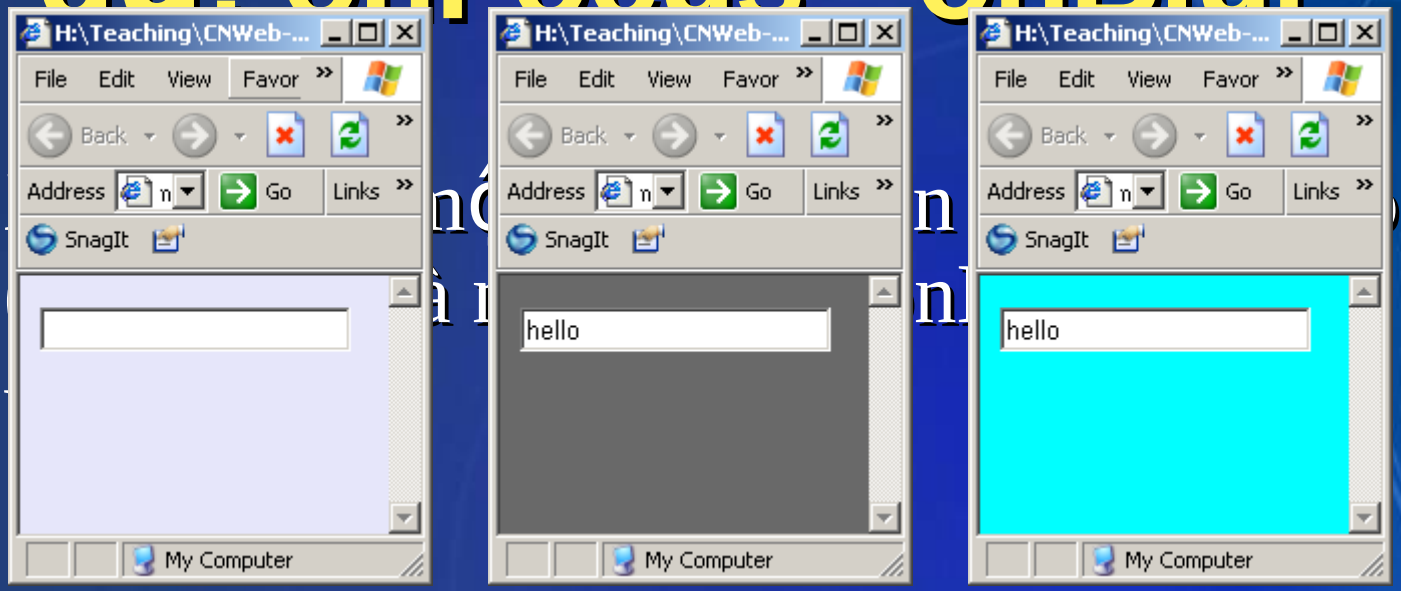
Ví dụ: onclick Event

```
<SCRIPT LANGUAGE="JavaScript">
     function compute(frm)
     {
        var x = frm.expr.value;
        result.innerHTML = x*x;
     }
</SCRIPT>

<FORM name="frm">
     X = <INPUT TYPE="text" NAME="expr" SIZE=15>
     <BR><BR>
     <INPUT TYPE="button" VALUE="Calculate"
     ONCLICK="compute(this.form)">
     <BR>
     X * X = <SPAN ID="result"></SPAN>
</FORM>
```



Ví dụ: onFocus - onBlur

-
-




```
<BODY BGCOLOR="lavender">
    <FORM>
       <INPUT type="text" name="myTextbox"
              onblur="(document.bgColor='aqua')"
              onfocus="(document.bgColor='dimgray')">
    </FORM>
</BODY>
```

Nội dung

- Giới thiệu về Javascript
- Nhúng Javascript vào trang web
- Kiểu dữ liệu & Các cú pháp Javascript
- Sử dụng các đối tượng trong Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

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

- Là thể hiện của đối tượng cửa sổ trình duyệt
- Tồn tại khi mở 1 tài liệu HTML
- Sử dụng để truy cập thông tin của các đối tượng trên cửa sổ trình duyệt (tên trình duyệt, phiên bản trình duyệt, thanh tiêu đề, thanh trạng thái …)

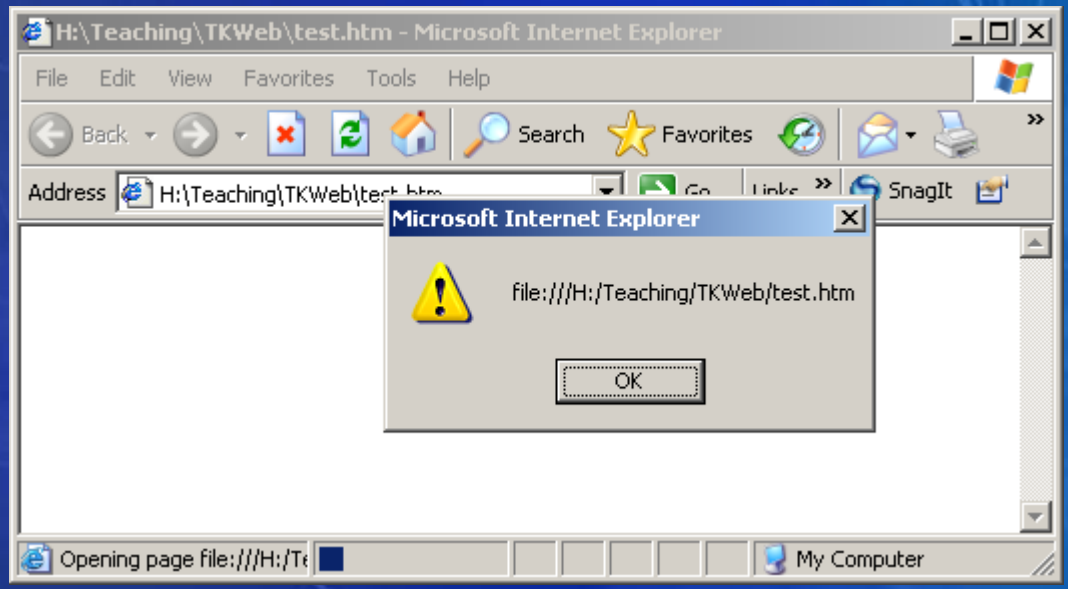
Đối tượng Window - DOM

- **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**
  - aLinkColor
  - bgColor
  - body
  - fgColor
  - linkColor
  - title
  - URL
  - vlinkColor
  - forms[]
  - images[]
  - childNodes[]
  - documentElement
  - cookie
  - ……
- **Methods**
  - close
  - open
  - createTextNode(“ text ")
  - createElement(“HTMLtag")
  - getElementById(“id”)
  - …

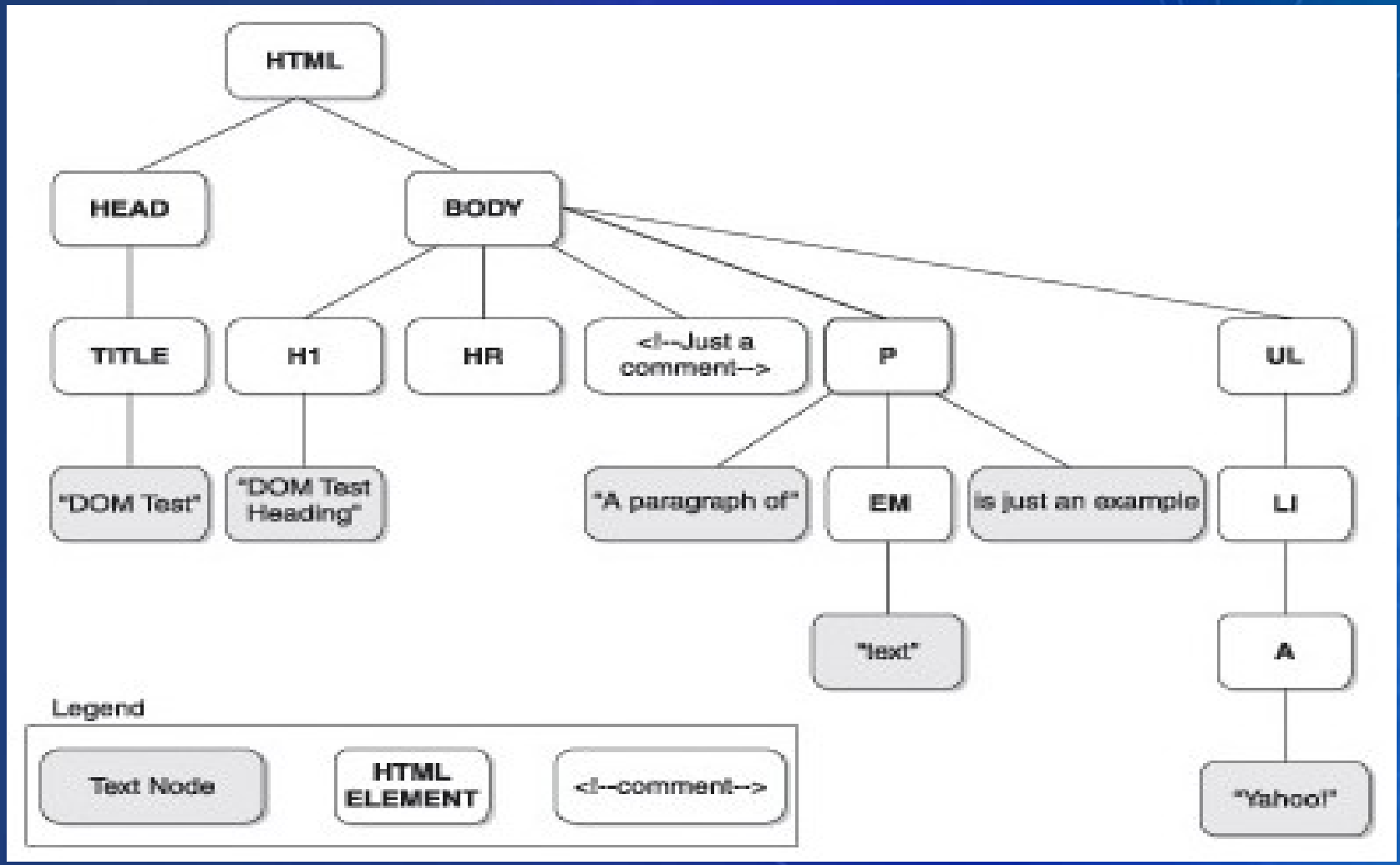
Đối tượng Document - DOM

- 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>
</html>
```

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

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”;

// <img 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<img src=“images/test.gif” />
//</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”
```

Đối tượng Document - DOM

- Thay đổi định dạng **CSS** của một node thông qua thuộc tính style

Ví dụ:

```
<p id=“myParagraph” style="color: red;">This is a text</p>
<script language=“javascript” >
      var node = document.getElementById(“myParagraph”);
      node.style.color = "green";
      node.style.fontSize = "14";
      node.style.backgroundColor = "yellow";
</script>
```

Đối tượng Document - DOM

- Thay đổi định dạng css thông qua thuộc tính className

Ví dụ:

```
<head>
<style type="text/css">
    .look1  {   color: black;  background-color:
yellow; font-style: normal; }
    .look2  {   background-color: orange; font-style:
italic; }
</style>
</head>
<body>
      <p id="p1" class="look1">
            this is my text     </p>
      <script language="javascript" >
            var pNode = document.getElementById("p1");
            pNode.className = "look2";
      </script>
```

Đối tượng Document - DOM

- Thay đổi tham chiếu đến file **CSS**

Ví dụ:

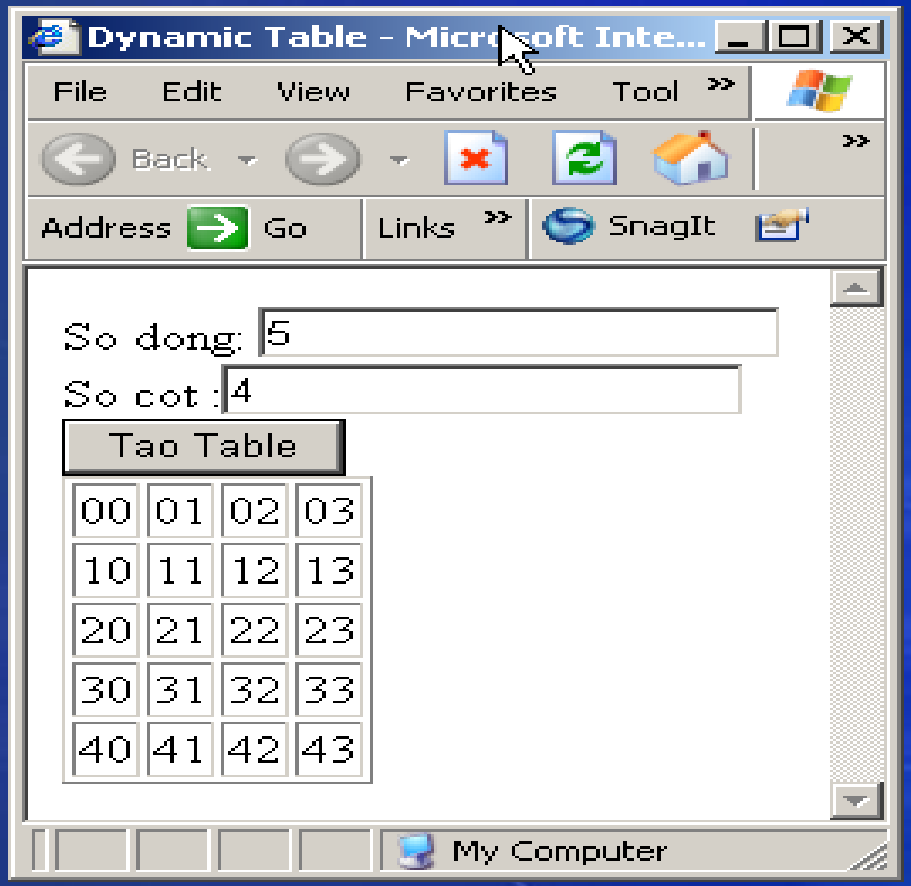
```
<head>
        <script language="javascript" >
         function changeSkin()
         {
             document.getElementById("myStyle").href =
                                        "css/style2.css";
         }

        </script>
        <link id="myStyle" rel="stylesheet"
type="text/css" href="css/style1.css" />
</head>
<body>
         <p class="style1">
               Hello world
         </p>
        <input type="button" onclick="changeSkin()"
value="change skin" />
```

Nội dung

- Giới thiệu về Javascript
- Kiểu dữ liệu & Các cú pháp Javascript
- Nhúng Javascript vào trang web
- Sử dụng các đối tượng trong Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

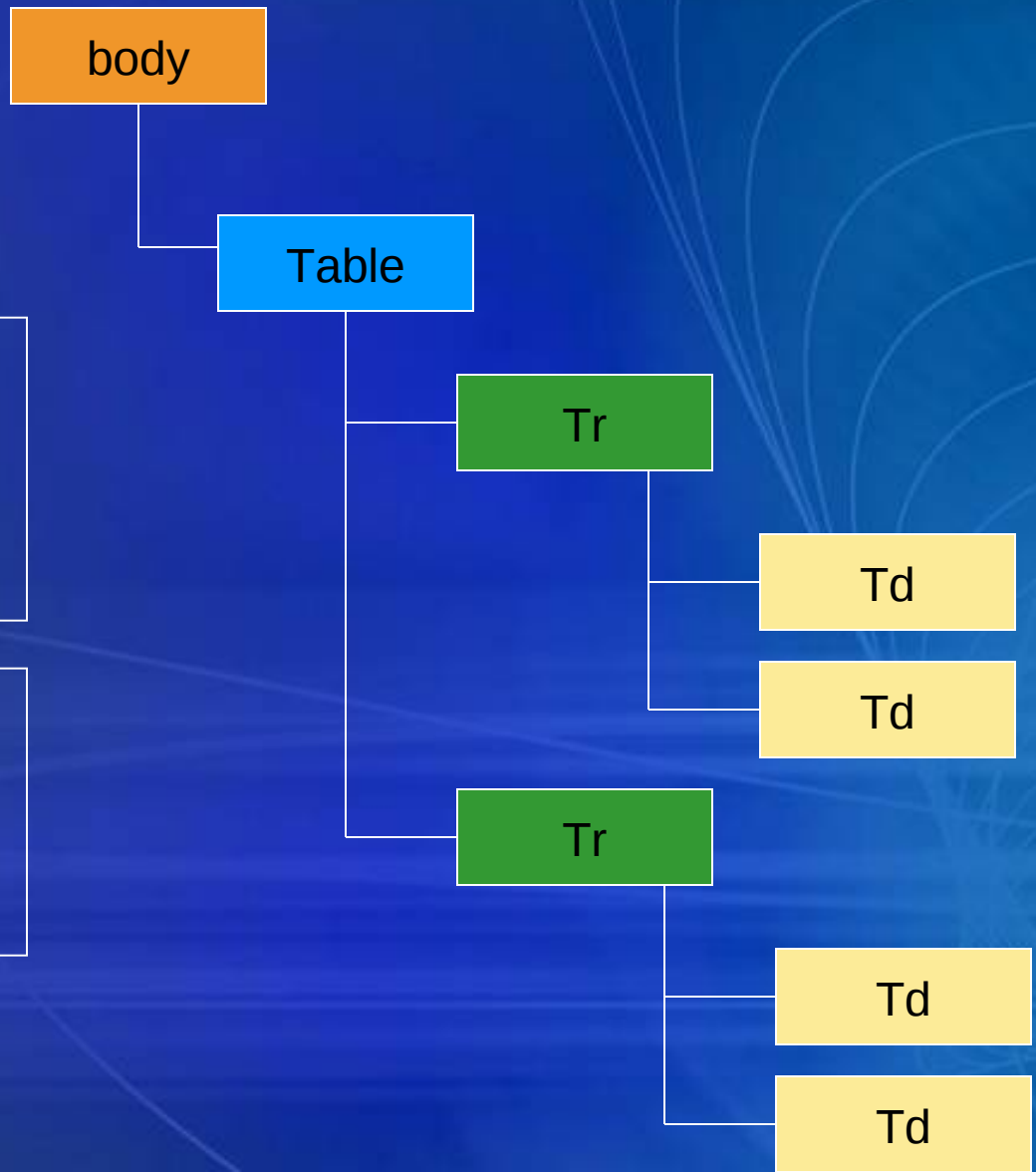
Ví dụ: Dynamic Table

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



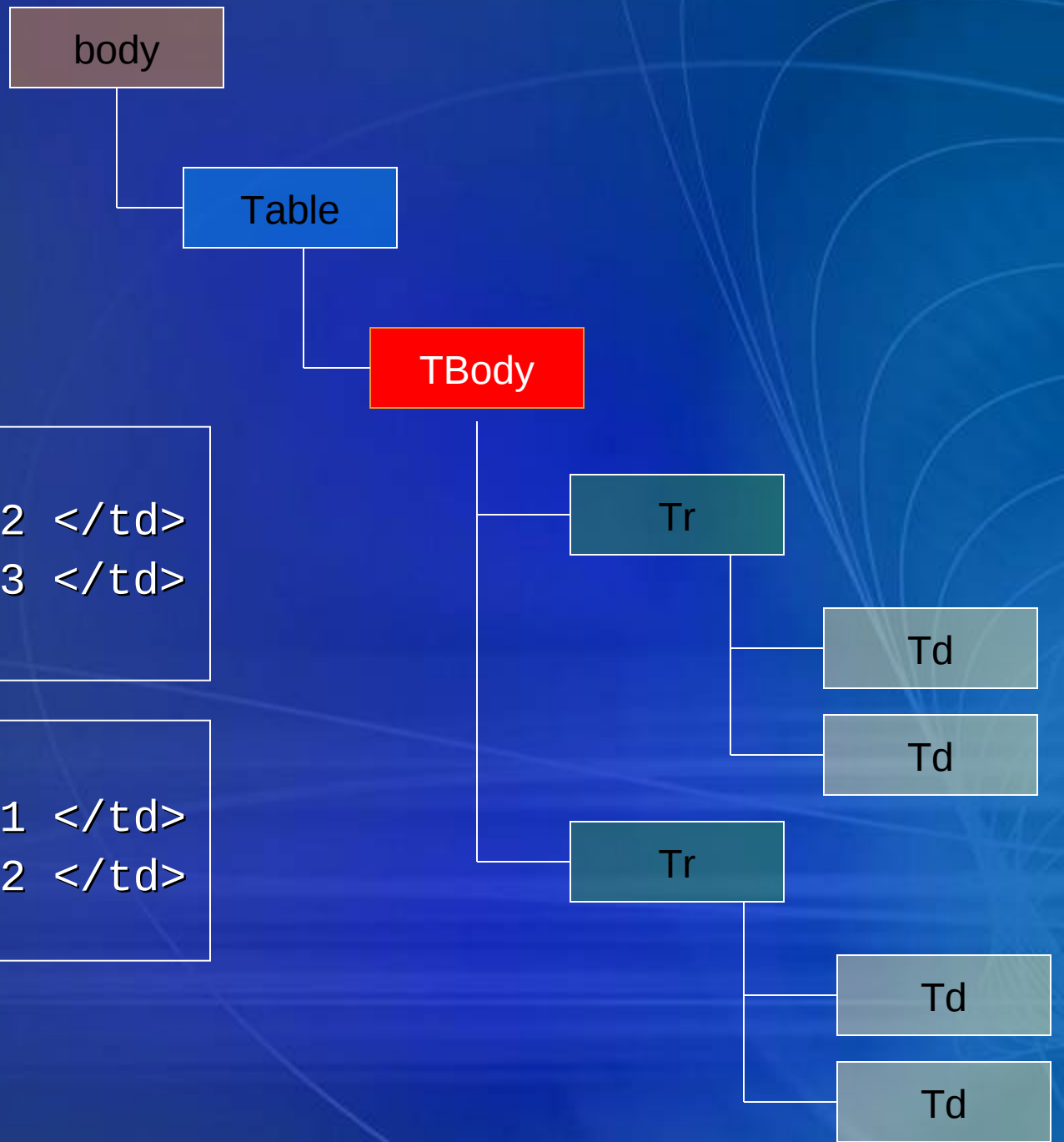
Ví dụ: Dynamic Table

```
<Table>
    <Tr>
        <td> 12 </td>
        <td> 13 </td>
    </Tr>
    <Tr>
        <td> 21 </td>
        <td> 22 </td>
    </Tr>
</Table>
```



Ví dụ:

```
<Table>
    <Tbody>
        <Tr>
            <td> 12 </td>
            <td> 13 </td>
        </Tr>

        <Tr>
            <td> 21 </td>
            <td> 22 </td>
        </Tr>
    </Tbody>
</Table>
```



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.

Ví dụ: Dynamic Table

```
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);
        }
```