

Phát triển ứng dụng web

Javascript

Nội dung

- ☐ Giới thiệu Javascript
- ☐ Cách nhúng Javascript vào HTML
- ☐ Javascript – cơ bản
- ☐ Javascript – events
- ☐ Javascript – HTML DOM

Nội dung

- ❑ **Giới thiệu Javascript**
- ❑ Cách nhúng Javascript vào HTML
- ❑ Javascript – cơ bản
- ❑ Javascript – events
- ❑ Javascript – HTML DOM

Giới thiệu về Script

❑ 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

- ❑ **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.

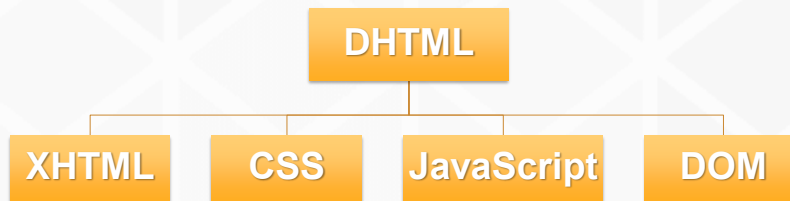


Nơi cung cấp

- ❑ 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 Scripts* và trả về nội dung HTML cho trình duyệt.
- ❑ Nội dung HTML trả về chủ yếu bao gồm: *mã html, client-script*.

Dynamic HTML (DHTML)

- ❑ Cho phép trang web có thể tương tác và thay đổi tùy theo hành động của người dùng.
- ❑ DHTML = HTML + CSS + JavaScript



7

Ý nghĩa Javascript

- ❑ HTML: xác định nội dung trang web thông qua các thẻ ngữ nghĩa (heading, paragraph, list...).
- ❑ CSS: xác định luật hay định dạng để thể hiện tài liệu HTML
 - ❑ Font chữ
 - ❑ Nền (màu, hình ảnh, ...)
 - ❑ Vị trí và kích thước
- ❑ Javascript: xác định các hành động
 - ❑ Tương tác thông qua các hành động của người dùng, xử lý sự kiện, ...

8

Nội dung

- ☐ Giới thiệu Javascript
- ☐ **Cách nhúng Javascript vào HTML**
- ☐ Javascript – cơ bản
- ☐ Javascript – events
- ☐ Javascript – HTML DOM

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="abc.js"></script>
```

Nhúng JavaScript vào trang Web

- ❑ Web Browser sẽ thực thi **<script>** khi load trang web theo thứ tự từ *trên xuống dưới*.
- ❑ Đối với **Source code JavaScript** có thể đặt trong các file **.js** sẽ được **nhúng** vào file HTML trước khi hoạt động.
- ❑ Các đoạn **code JavaScript** được Browser xử **cùng thứ tự** với các thẻ **HTML**. Trừ các **hàm (function)** chỉ được thực thi khi **có lời gọi hàm**.

Nhúng Javascript vào trang web

```

<html>
<head>
  <script type="text/javascript">
    some statements
  </script>
</head>
<body>
  <script type="text/javascript">
    some statements
  </script>

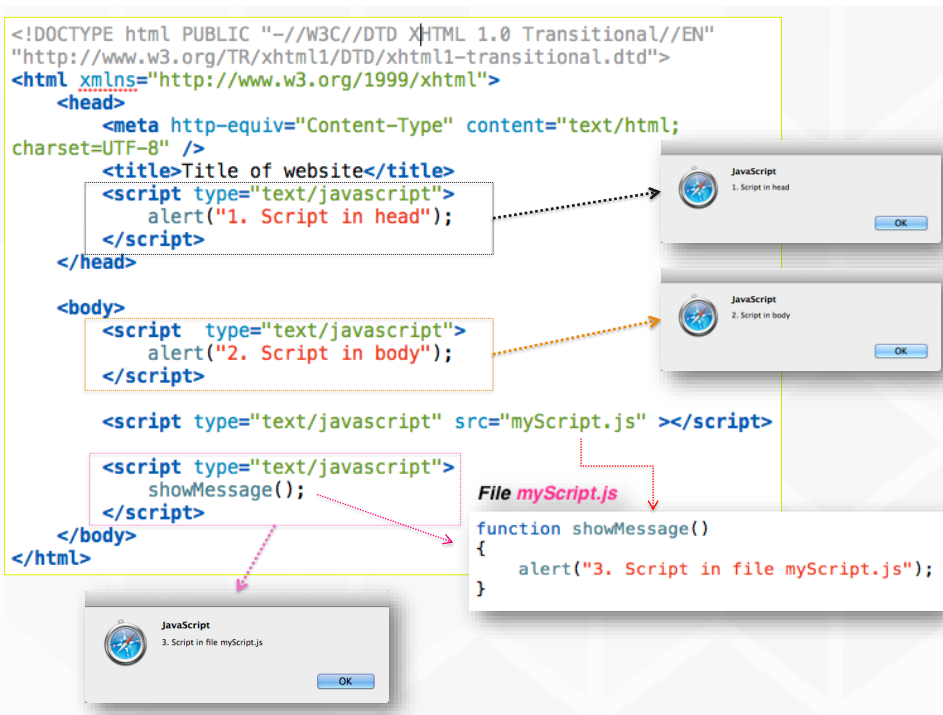
  <script src="Tên_file_script.js">method()</script>

  <script type="text/javascript">
    // gọi 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.



Nội dung

- ☐ Giới thiệu Javascript
- ☐ Cách nhúng Javascript vào HTML
- ☐ **Javascript – cơ bản**
- ☐ Javascript – events
- ☐ Javascript – HTML DOM

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 tiên.
 - ☐ Biến không cần khai báo kiểu dữ liệu vì biến trong javascript *không có kiểu dữ liệu nhất định*

Kiểu dữ liệu trong Javascript

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

Một biến trong JavaScript có thể lưu bất kỳ kiểu 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);
}
```



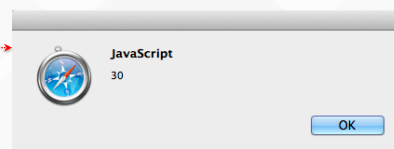
Hàm trong Javascript

- ❑ Ví dụ:

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

- ❑ Gọi hàm:

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



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 label1 :
        statementlist
    case label2 :
        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** & **do ... while**

while (expression)

```
{
    statements
}
```

Ví dụ:

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

do

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

Ví dụ:

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

Nội dung

- ☐ Giới thiệu Javascript
- ☐ Cách nhúng Javascript vào HTML
- ☐ Javascript – cơ bản
- ☐ **Javascript – events**
- ☐ Javascript – HTML DOM

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**
- ☐ ...



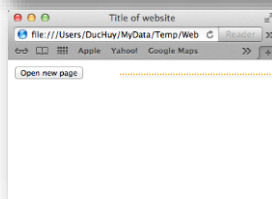
Xử lý sự kiện cho các thẻ HTML

❑ Cú pháp:

<TAG eventHandler = "JavaScript Code">

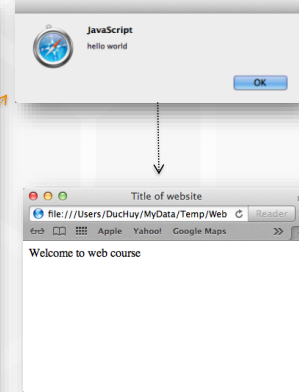
❑ Lưu ý: Dấu "..." và '...'

```
<body>
  <input type="button"
        name="btnClickMe"
        value="Open new page"
        onclick="window.open('http://www.google.com');" />
</body>
```



Xử lý sự kiện bằng function

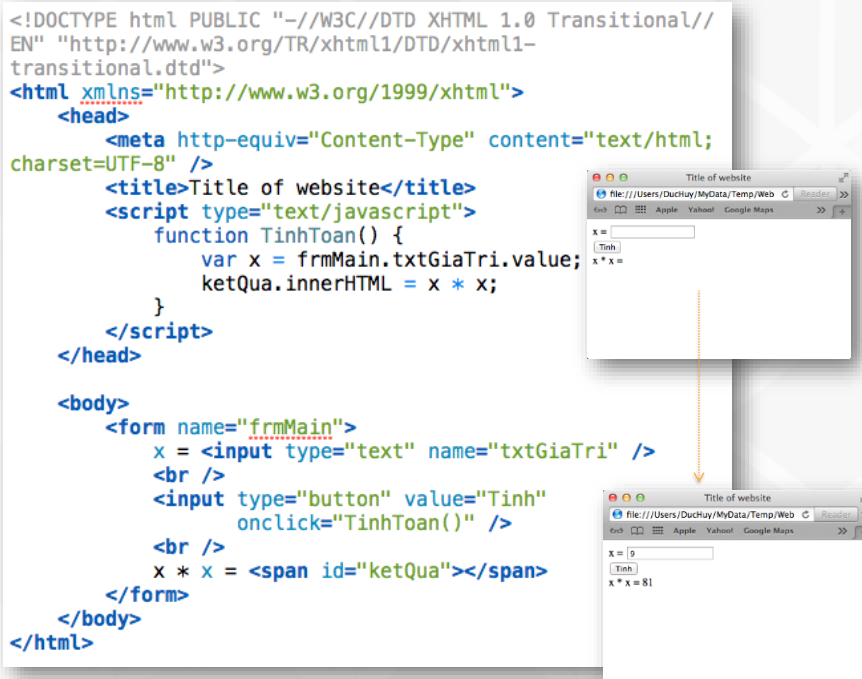
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
    <title>Title of website</title>
    <script type="text/javascript">
      function ShowMessage() {
        alert("hello world");
      }
    </script>
  </head>
  <body onload="ShowMessage()">
    Welcome to web course
  </body>
</html>
```



Xử lý sự kiện bằng thuộc tính

- ❑ Gán tên hàm xử lý cho 1 object event
object.eventhandler = function_name;

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



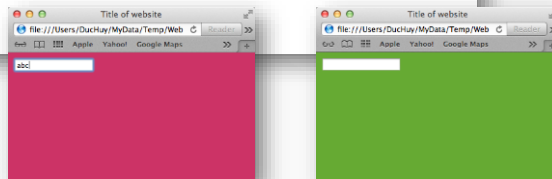
Ví dụ onFocus - onBlur

- ❑ Xảy ra khi một thành phần HTML bị focus (**onFocus**) và mất focus (**onBlur**)

```

<body>
  <form name="frmMain">
    <input type="text"
      onFocus="document.bgColor='#cc3366';"
      onBlur="document.bgColor='#66aa33';"
    />
  </form>
</body>

```



Nội dung

- ☐ Giới thiệu Javascript
- ☐ Cách nhúng Javascript vào HTML
- ☐ Javascript – cơ bản
- ☐ Javascript – events
- ☐ **Javascript – HTML DOM**

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

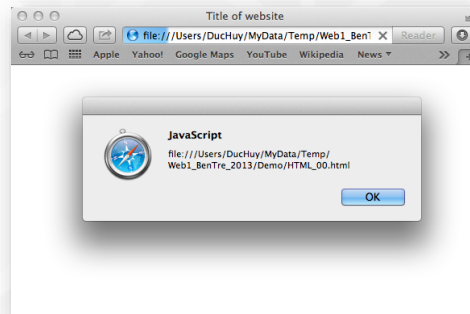
☒ Methods

- ☒ 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>
```

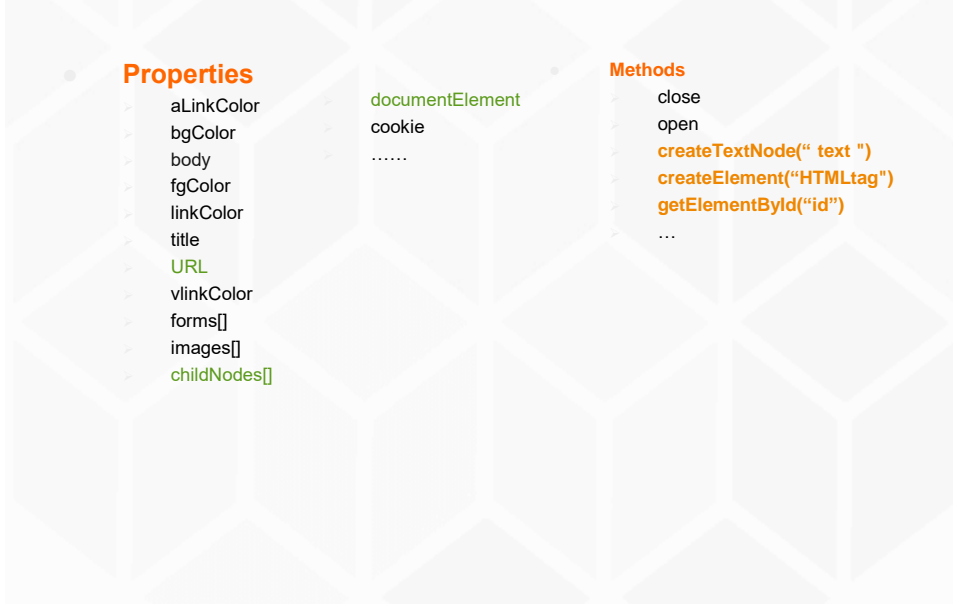


Đối tượng **Document** - DOM

- ❑ 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



Đối tượng Document - DOM

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

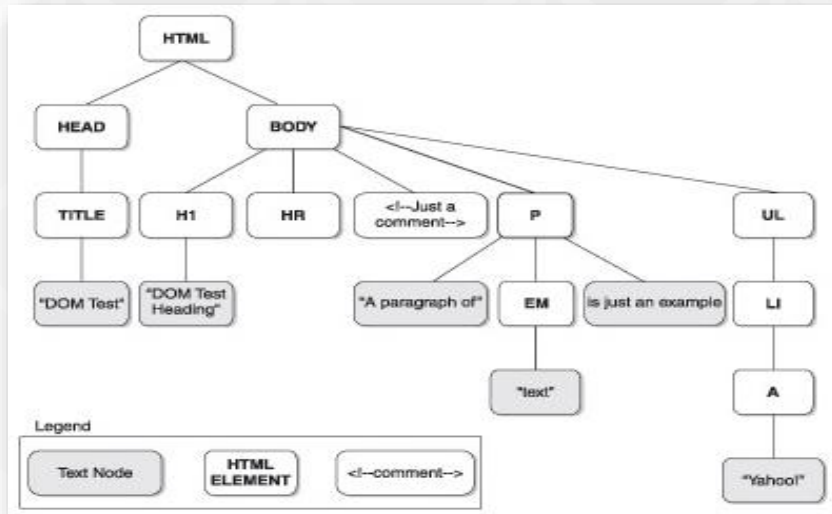
```

<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	<p>...</p>
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";

// 
```

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";

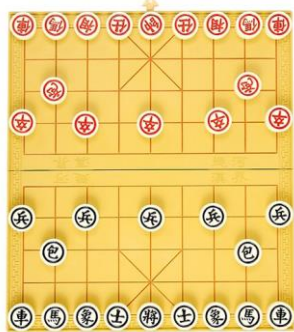
// Kết quả hiển thị trên trình duyệt
// bên trong thẻ p: "Some <b> new </b> text"
```


Nội dung

- ☐ Giới thiệu Javascript
- ☐ Cách nhúng Javascript vào HTML
- ☐ Javascript – cơ bản
- ☐ Javascript – events
- ☐ Javascript – HTML DOM
- ☐ **Bài tập**

Bài tập 1

- ☐ Dựng trang HTML là bàn cờ (cờ vua hoặc cờ tướng), cho phép sử dụng chuột để di chuyển các quân cờ (chỉ cho phép thay đổi vị trí vào ô đúng).



Bài tập 2



- ❑ Dùng HTML, CSS và Javascript để làm trang web như sau:

Bé tập tính

Số thứ nhất	<input type="text"/>	<input type="radio"/> Cộng	<input type="radio"/> Trừ
Số thứ hai	<input type="text"/>	<input type="radio"/> Nhân	<input checked="" type="radio"/> Chia
Kết quả	<input type="text"/>	<input type="button" value="Tính"/>	
Thông báo	Giá trị nhập ở ô <i>Số thứ nhất</i> không phải là số		

51

- ❑ Tạo giao diện, canh thẳng theo cột, chỉ có thể chọn một trong bốn phép tính.
- ❑ Viết mã Javascript để khi bấm vào nút tính, kết quả của phép tính đang chọn sẽ được điền vào ô kết quả.
- ❑ Vùng thông báo sẽ hiển thị thông báo lỗi khi
 - ❑ Di chuyển khỏi ô nhập *Số thứ nhất* hoặc *Số thứ hai* mà dữ liệu không phải là số thực.
 - ❑ Bấm nút *Tính* mà chưa chọn phép tính.
 - ❑ Chưa điền đủ hai số hợp lệ để thực hiện phép tính (4 điểm)