

2-2009

**GIÁO TRÌNH  
THIẾT KẾ WEB**

**Lưu Hành Nội Bộ**

## CHƯƠNG I: GIỚI THIỆU VỀ WEB

### I.1. CÁC KHÁI NIỆM CƠ BẢN:

- *Internet* là một mạng máy tính toàn cầu trong đó các máy truyền thông với nhau theo một ngôn ngữ chung là TCP/IP.
- *Intranet* đó là mạng cục bộ không nối vào Internet và cách truyền thông của chúng cũng theo ngôn ngữ chung là TCP/IP.
- *Mô hình Client-Server*: là mô hình khách-chủ. Server chứa tài nguyên dùng chung cho nhiều máy khách(Client) như các tập tin, tài liệu, máy in... Ưu điểm của mô hình này là tiết kiệm về thời gian, tài chính, dễ quản trị hệ thống... Cách hoạt động của mô hình này là máy Server ở trạng thái hoạt động(24/24) và chờ yêu cầu từ phía Client. Khi Client yêu cầu thì máy Server đáp ứng yêu cầu đó.
- *Internet Server* là các Server cung cấp các dịch vụ Internet(Web Server, Mail Server, FTP Server...)
- *Internet Service Provider(ISP)*: Là nơi cung cấp các dịch vụ Internet cho khách hàng. Mỗi ISP có nhiều khách hàng và có thể có nhiều loại dịch vụ Internet khác nhau.
- *Internet Protocol* : Các máy sử dụng trong mạng Internet liên lạc với nhau theo một tiêu chuẩn truyền thông gọi là Internet Protocol (IP). IP Address-địa chỉ IP: để việc trao đổi thông tin trong mạng Internet thực hiện được thì mỗi máy trong mạng cần phải định danh để phân biệt với các máy khác. Mỗi máy tính trong mạng được định danh bằng một nhóm các số được gọi là địa chỉ IP. Địa chỉ IP gồm 4 số thập phân có giá từ 0 đến 255 và được phân cách nhau bởi dấu chấm. Ví dụ 192.168.0.1 Địa chỉ IP này có giá trị trong toàn mạng Internet. Ủy ban phân phối địa chỉ IP của thế giới sẽ phân chia các nhóm địa chỉ IP cho các quốc gia khác nhau. Thông thường địa chỉ IP của một quốc gia do các cơ quan bưu điện quản lý và phân phối lại cho các ISP. Một máy tính khi thâm nhập vào mạng Internet cần có một địa chỉ IP. Địa chỉ IP có thể cấp tạm thời hoặc cấp vĩnh viễn. Thông thường các máy Client kết nối vào mạng Internet thông qua một ISP bằng đường điện thoại. Khi kết nối, ISP sẽ cấp tạm thời một IP cho máy Client.
- *Phương thức truyền thông tin trong Internet*: Khi một máy tính có địa chỉ IP là x(máy X) gửi tin đến máy tính có địa chỉ IP là y (máy Y) thì phương thức truyền tin cơ bản diễn ra như sau: Nếu máy X và máy Y cùng nằm trên một mạng con thì thông tin sẽ được gửi đi trực tiếp. Còn máy X và Y không cùng nằm trong mạng con thì thông tin sẽ được chuyển tới một máy trung gian có đường thông với các mạng khác rồi mới chuyển tới máy Y. Máy trung gian này gọi là Gateway.
- *World Wide Web(WWW)*: là một dịch vụ phổ biến nhất hiện nay trên Internet. Dịch vụ này đưa ra cách truy xuất các tài liệu của các máy phục vụ để dàng thông qua các giao tiếp đồ họa. Để sử dụng dịch vụ này máy Client cần có một chương trình gọi là Web Browser.
- *Web Browser(trình duyệt)*: là trình duyệt Web. Dùng để truy xuất các tài liệu trên các Web Server. Các trình duyệt hiện nay là Internet Explorer, Nestcape
- *Home page*: là trang web đầu tiên trong web site
- *Hosting provider*: là công ty hoặc tổ chức đưa các trang của chúng ta lên web
- *Hyperlink* : tên khác của hypertextlink
- *Publish*: làm cho trang web chạy được trên mạng
- *URL(Unioform resource locator)*: một địa chỉ chỉ đến một file cụ thể trong nguồn tài nguyên mạng.

- Mỗi nguồn trên web có duy nhất một địa chỉ rất khó nhớ. Vì vậy, người ta sử dụng URL là một chuỗi cung cấp địa chỉ Internet của một web site hoặc nguồn trên World Wide Web. Định dạng đặc trưng là:

www.nameofsite.typeofsite.countrycode

Ví dụ:

207.46.130.149 được biểu diễn trong URL là www.microsoft.com

- URL cũng nhận biết giao thức của site hoặc nguồn được truy cập. Giao thức thông thường nhất là “http”, một vài dạng URL khác là “gopher”, cung cấp địa chỉ Internet của một thư mục Gopher, và “ftp”, cung cấp vị trí mạng của nguồn FTP.
- Có hai dạng URL:
  - ✧ *URL tuyệt đối* – là địa chỉ Internet đầy đủ của một trang hoặc file, bao gồm giao thức, vị trí mạng, đường dẫn tùy chọn và tên file.  
Ví dụ, http:// www.microsoft.com/ms.htm.
  - ✧ *URL tương đối* - mô tả ngắn gọn địa chỉ tập tin kết nối có cùng đường dẫn với tập tin hiện hành, URL tương đối đơn giản bao gồm tên và phần mở rộng của tập tin.  
Ví dụ: index.html
- *Web server* là một chương trình đáp ứng lại các yêu cầu truy xuất tài nguyên từ trình duyệt.

## I.2. GIỚI THIỆU KHÁI QUÁT VỀ WEB

- *Web* là một ứng dụng chạy trên mạng(Client-Server), được chia sẻ khắp toàn cầu.
- *Trang web* là một file văn bản chứa những tag HTML hoặc những đoạn mã đặc biệt mà trình duyệt web (Web browser) có thể hiểu và thông dịch được, file được lưu với phần mở rộng là .html hoặc htm.
- *HTML (HyperText Markup Language)*, gồm các đoạn mã chuẩn được quy ước để thiết kế Web và được hiển thị bởi trình duyệt Web (Web Browser)
  - *Hypertext* (Hypertext link), là một từ hay một cụm từ đặc biệt dùng để tạo liên kết giữa các trang web
  - *Markup*: là cách định dạng văn bản để trình duyệt hiểu và thông dịch được.
  - *Language*: đây không là ngôn ngữ lập trình, mà chỉ là tập nhỏ những quy luật để định dạng văn bản trên trang web.
- *Trình soạn thảo trang web* :Có thể soạn thảo web trên bất kỳ trình soạn thảo văn bản nào. Các trình soạn thảo phổ biến hiện nay là: Notepad, FrontPage hoặc Dreamweaver.

## I.3. TAG HTML:

Tag HTML là những câu lệnh nằm giữa cặp tag “<” và “>”, dùng để định dạng các văn bản trên trang web. Dạng chung của một tag HTML là:

**<tagName ListProperties> Object </tagName>**

Trong đó:

- *TagName* : là tên một tag HTML, viết liền với dấu “<”, không có khoảng trắng
- *Object* : là đối tượng cần định dạng trong trang Web
- *ListProperties* là danh sách thuộc tính của Tag, là những đặc điểm bổ sung vào cho một tag, thứ tự các thuộc tính trong một tag là tùy ý. Nếu có từ 2 thuộc tính trở lên thì mỗi thuộc tính cách nhau bởi khoảng trắng.  
**<TagName property1='value1' property2='value2' ...>Object</TagName>**
- Giá trị của thuộc tính được đặt trong nháy đơn ‘ hoặc nháy đôi “. (có thể bỏ qua)

- `<TagName>`: gọi là tag mở
- `</TagName>`: gọi là tag đóng. Thông thường thì các tag đều có tag đóng. Tuy nhiên có một số tag không có tag đóng  
**Ví dụ :** `<body BGCOLOR="RED">nội dung </body>`  

↑  
TagName(mở)

↑  
Properties

↑  
TagName(đóng)
- Có thể có nhiều tag lồng vào nhau, theo nguyên tắc tag nào mở trước thì tag đó đóng sau  
**Ví dụ:**  

```
<Tag1><Tag2>Object</Tag2></Tag1>
<B>Object1<I>Object2 </I></B>
```
- Trong trang HTML, nếu một tag bị sai thì nội dung bên trong Tag đó không hiển thị trên trình duyệt

## I.4. CẤU TRÚC CƠ BẢN CỦA TRANG WEB:

### 1. Cấu trúc trang web

- Phần đầu(`<Head></Head>`): là phần chứa thông tin của trang Web.
- Phần thân (`<Body></Body>`): là phần chứa nội dung của trang Web.
- Phần đầu và phần thân được đặt trong cặp tag `<HTML></HTML>`

`<HTML>`

`<HEAD>`

**Nội dung thông tin của trang web**

`</HEAD>`

`<BODY>`

**Nội dung hiển thị trên trình duyệt**

`</BODY>`

`</HTML>`

### 2. Hiển thị trang web:

- Khởi động trình duyệt Internet Explorer
- Chọn menu file,open, dùng browse tìm tập tin html mới tạo
- Hoặc double click vào tên tập tin .htm

## I.5. CÁC TAG HTML CƠ BẢN :

### I.5.1. `<Title>` : Hiển thị nội dung tiêu đề của trang web trên thanh tiêu đề của trình duyệt.

- Cặp tag `<Title>` được đặt trong phần `<Head>` của trang HTML
- **Cú pháp:**

`<TITLE> Nội dung tiêu đề </TITLE>`

### I.5.2. `<Hn>`: Tạo header, gồm 6 cấp header, được đặt trong phần **BODY**

- **Cú pháp:**  
`<Hn ALIGN= "Direction"> Nội dung của Header </Hn>`

Trong đó:

- Direction: gồm các giá trị left, right, center, dùng để canh lề cho header, mặc định là canh trái

- Ví dụ:

```
<H1>Heading 1</H1>
<H2>Heading 2</H2>
<H3>Heading 3</H3>
<H4>Heading 4</H4>
<H5>Heading 5</H5>
<H6>Heading 6</H6>
```



### I.5.3. <P> :

- Dùng để ngắt đoạn và bắt đầu đoạn mới
- **Cú pháp:**

**<P ALIGN = "Direction"> Nội dung của đoạn </P>**

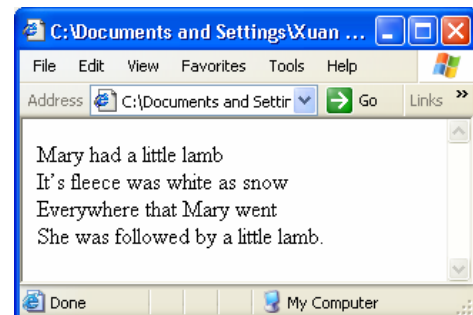
- Tag </P> không bắt buộc.
- Tag <P> kế tiếp sẽ tự động bắt đầu một đoạn mới.

### I.5.4. <BR>:

- Ngắt dòng tại vị trí của của tag.

Ví dụ:

```
<p>
Mary had a little lamb <br>
It's fleece was white as snow<br>
Everywhere that Mary went <br>
She was followed by a little lamb<br>
</p>
```



### I.5.5. <HR>:

- Dùng để kẻ đường ngang trang, không có tag đóng
- **Cú pháp:**

**<HR Align="direction" Width="Value" Size=value color=#rrggbb>**

Trong đó:

- ✧ Direction: gồm các giá trị left, right, center
- ✧ Width: độ dài đường kẻ, tính bằng Pixel hoặc %
- ✧ Size: độ dày của đường kẻ, tính bằng pixel
- ✧ Color: màu đường kẻ, có thể dùng tên màu hoặc dùng mã #rrggbb

Ví dụ:

```
<HTML>
<HEAD><TITLE>Welcome to HTML </TITLE></HEAD>
<BODY>
<H3> My first HTML document</H3>
<HR size = 5 align = center width = 50%>
<HR size = 15 align = left width = 80%>
<P> This is going to be real fun
</BODY>
</HTML>
```

**I.5.6. <FONT>:**

- Dùng định dạng font chữ
- Định dạng Font chữ cho cả tài liệu thì đặt tag <Font> trong phần <Body>
- Định dạng từng phần hoặc từng từ thì đặt tại vị trí muốn định dạng
- **Cú pháp:**

**<FONT Face="fontName1, fontName2, fontName3" size="value" Color="rrggbb">**

**Nội dung hiển thị**

**</FONT>**

Ví dụ:

```
<HTML>
  <HEAD>
    <TITLE>Welcome to HTML</TITLE>
  </HEAD>
  <BODY>
    <FONT SIZE = 3 COLOR = HOTPINK FACE = Arial>
    My first HTML document </FONT>
    <P> This is <FONT COLOR=BLUE SIZE = 6>going
    </FONT> to be real fun
  </BODY>
</HTML>
```

**I.5.7. <BODY > :**

- Chứa nội dung của trang web
- Cú pháp:
  - <BODY>**
  - Nội dung chính của trang web**
  - </BODY>**
- Các thuộc tính của <Body>
  - ✧ BgColor: thiết lập màu nền của trang
  - ✧ Text: thiết lập màu chữ
  - ✧ Link: màu của siêu liên kết
  - ✧ Vlink: màu của siêu liên kết đã xem qua
  - ✧ Background: dùng load một hình làm nền cho trang
  - ✧ LeftMargin: Canh lề trái
  - ✧ TopMargin: Canh lề trên của trang

Ví dụ:

```
<HTML>
  <HEAD><TITLE> Learning HTML</TITLE></HEAD>
  <BODY BGCOLOR="#0000FF" text="yellow">
    <FONT COLOR = LIMEGRREN>Welcome to HTML</FONT>
  </BODY>
</HTML>
```

- ✧ **Màu sắc:** Internet Explorer có thể xác lập 16 màu theo tên như sau:
  - Black, Silver, Gray, White, Maroon, Red, Purple, Fuchsia, Green, Lime, Olive, Yellow, Navy, Blue, Teal, Aqua.
  - Một số mã thập lục phân của màu :#RRGGBB

Mã thập lục phân	Màu
#FF0000	RED
#00FF00	GREEN
#0000FF	BLUE
#000000	BLACK
#FFFFFF	WHITE

**I.5.8. <IMG> :**

- Dùng để chèn một hình ảnh vào trang Web
- **Cú pháp:**

**<Img src="URL" alt="Text" width=value height=value border=value>**

- o Src: xác định đường dẫn tập tin cần load, sử dụng đường dẫn tương đối <Img src="../images/h1.gif"> .
- o Alt: chứa nội dung văn bản thay thế cho hình ảnh khi hình không load về được, nếu load về được thì sẽ xuất hiện nội dung trong textbox mỗi khi người dùng đưa chuột tới hình.
- o Width, Height: dùng để xác định chế độ phóng to thu nhỏ hình ảnh.
- o Align =" left/ right/top/bottom": so hàng giữa hình ảnh và text

**I.5.9. <BgSound> :**

- Dùng để chèn một âm thanh vào trang Web. Âm thanh này sẽ được phát mỗi khi người sử dụng mở trang Web.
- **Cú pháp:**

**<BgSound src="filenhac" Loop=value>**

- o Src chứa địa chỉ file nhạc, file này có phần mở rộng .mp3 , mdi, ...
- o Loop xác định chế độ lặp đi lặp lại của bài hát, nếu value< 0 thì lặp vô hạn, value=n thì lặp lại n lần rồi tự động tắt.

**I.5.10. <EMBED>:**

- Cho phép đưa âm thanh trực tiếp vào trang WEB.
- **Cú pháp:**

**<EMBED SRC="URL" >**

**Ví dụ:**

**<EMBED SRC="clouds.mid" WIDTH="145" HEIGHT="61">**

**I.5.11. <Marquee></Marquee> :**

- Dùng để điều khiển đối tượng chạy một cách tự động trên trang Web
- **Cú pháp:**

**<Marquee >Object</Marquee>**

- Các thuộc tính của Marquee :
  - o *Direction* =up/ down / left / right dùng để điều khiển hướng chạy.
  - o *Behavior*=alternate: đối tượng chạy từ lề này sang lề kia và ngược lại.

**Ví dụ:**

**<Marquee direction=up>Đối tượng chạy lên </Marquee>**

**I.5.12. <!-- Ghi chú -->: Nội dung trong cặp tag này không hiển thị trong trang**

Cú pháp: <! -- Nội dung lời chú thích -->

### I.5.13. <B>: định dạng chữ đậm

- Cú pháp

<B> Nội dung chữ đậm</B>

Ví dụ:

<P><B> This is good fun</B></P>

### I.5.14. Tag <I>: Định dạng chữ nghiêng

- Cú pháp:

<I> Nội dung chữ nghiêng</I>

### I.5.15. Tag <U>: Gạch chân văn bản

- Cú pháp:

<U> Nội dung chữ gạch chân</U>

Ví dụ:

Định dạng khối văn bản vừa đậm, nghiêng và gạch chân

<B><I><U> Trường ĐHCN TP HCM</U></I></B>

### I.5.16. Tag <BIG> và <SMALL>:

- Chính cỡ chữ to hoặc nhỏ hơn cỡ chữ xung quanh

- Cú pháp

<BIG>Nội dung chữ to </BIG>

<SMALL>Nội dung chữ nhỏ </SMALL>

### I.5.17. Tag <SUP> và <SUB> :

- Đưa chữ lên cao hoặc xuống thấp so với văn bản bình thường

- Cú pháp:

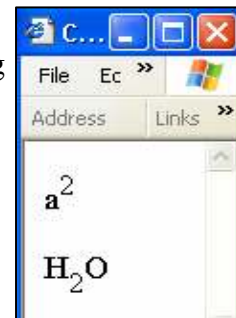
<SUP>Nội dung chữ đưa lên cao </SUP>

<SUB>Nội dung chữ đưa xuống thấp </SUB>

Ví dụ:

a<SUP>2</SUP>

H<SUB>2</SUB>O



### I.5.18. <STRIKE>:

- Gạch ngang văn bản

- Cú pháp:

<STRIKE>Nội dung văn bản bị gạch ngang </STRIKE>

### I.5.19. <CODE>...</CODE>:

- Dùng để nhập một dòng mã có định dạng ký tự riêng. Dòng mã này không được thực hiện mà được hiển thị dưới dạng văn bản bình thường

- Cú pháp:

<CODE>

Nội dung văn bản muốn định dạng

</CODE>



Ví dụ:

```
<CODE>
  If (x > 0) <br>
    x = x + 1<br>
  else <br>
    y = y + 1
</CODE>
```

#### I.5.20. <EM>: Văn bản được nhấn mạnh (giống tag <I>)

- **Cú pháp:**

```
<EM>Văn bản được nhấn mạnh</EM>
```

#### I.5.21. <STRONG>: Định dạng chữ đậm (giống <B>)

- **Cú pháp:**

```
<STRONG>Văn bản được nhấn mạnh</STRONG>
```

#### I.5.22. <BLOCKQUOTE>:

- Dùng phân cách một khối văn bản để nhấn mạnh, đoạn văn bản này được tách ra thành một paragraph riêng, thêm khoảng trắng trên và dưới đoạn đồng thời thụt vào so với lề trái (tương đương chức năng của phím tab)

- **Cú pháp:**

```
<BLOCKQUOTE>
  Nội dung khối văn bản nhấn mạnh
</BLOCKQUOTE>
```

Ví dụ:

```
<HTML>
  <HEAD><TITLE>Learning HTML</TITLE></HEAD>
  <BODY>
    <BLOCKQUOTE><FONT color = hotpink>
      Humpty Dumpty sat on a wall
      Humpty Dumpty had a great fall
      All the King's horses
      And all the King's men
      Could not put Humpty Dumty together again
    </FONT>
  </BLOCKQUOTE>
  </BODY>
</HTML>
```

#### I.5.23. <PRE>:

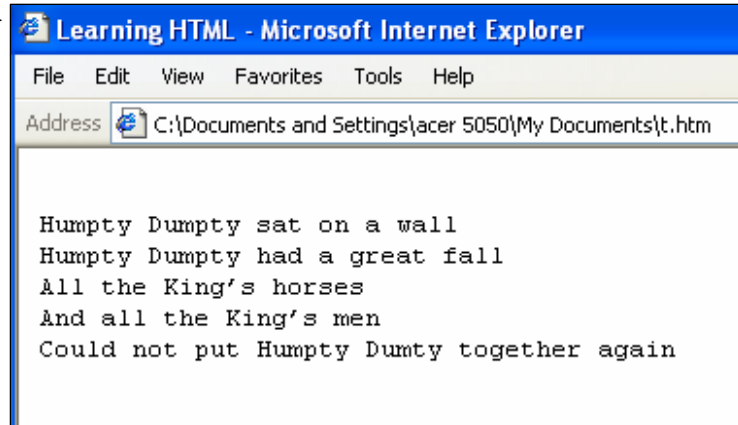
- Giữ nguyên các định dạng như: ngắt dòng, khoảng cách, thích hợp với việc tạo bảng

- **Cú pháp:**

```
<PRE>
  Nội dung văn bản cần định dạng trước với tất cả định dạng khoảng cách,
  xuống dòng và ngắt hàng
</PRE>
```

Ví dụ:

```
<HTML>
  <HEAD><TITLE>Learning HTML</TITLE></HEAD>
  <BODY>
    <PRE>
      Humpty Dumpty sat on a wall
      Humpty Dumpty had a great fall
      All the King's horses
      And all the King's men
      Could not put Humpty Dumty together again
    </PRE>
  </BODY>
</HTML>
```



#### 1.5.24. <DIV> <SPAN>:

- Chia văn bản thành các khối, có chung một định dạng
  - <DIV> chia văn bản thành một khối bắt đầu từ một dòng mới.
  - <SPAN> tách khối nhưng không bắt đầu từ một dòng mới
- **Cú pháp:**
  - <DIV>Nội dung của khối bắt đầu từ một dòng mới </DIV>
  - <SPAN>Nội dung của khối trong 1 dòng </SPAN>

Ví dụ:

```
<HTML>
  <HEAD><TITLE> Learning HTML</TITLE></HEAD>
  <BODY>
    <DIV>Division 1
      <P> The DIV element is used to group elements.
      <P>Typically, DIV is used for block level elements
    </DIV>
    <DIV align = right>
      <FONT size = 4 color = hotpink face = Arial>Division 2
      <P>This is a second division<Br>
      <H2>Are you having fun?</H2>
      </FONT>
    </DIV>
    <P> The second division is right aligned.
    <SPAN STYLE = "FONT-SIZE:25; Color:BLUE">Common formatting
    </SPAN> is applied to all the elements in the division
  </BODY>
</HTML>
```

**I.5.25. Các ký tự đặc biệt:**a. Lớn hơn (>): &gt;

Ví dụ:

```
<CODE>
    If A &gt; B
    Then <BR>
    A = A + 1
</CODE>
```

b. Nhỏ hơn (<): &lt;

Ví dụ:

```
<CODE>
    If A &lt; B
    Then <BR>
    A = A + 1
</CODE>
```

c. Cặp nháy””: &quot;

Ví dụ:

```
<BODY>
    &quot; To be or not to be? &quot; That is the question
</BODY>
```

d. Ký tự và &: &amp;

Ví dụ:

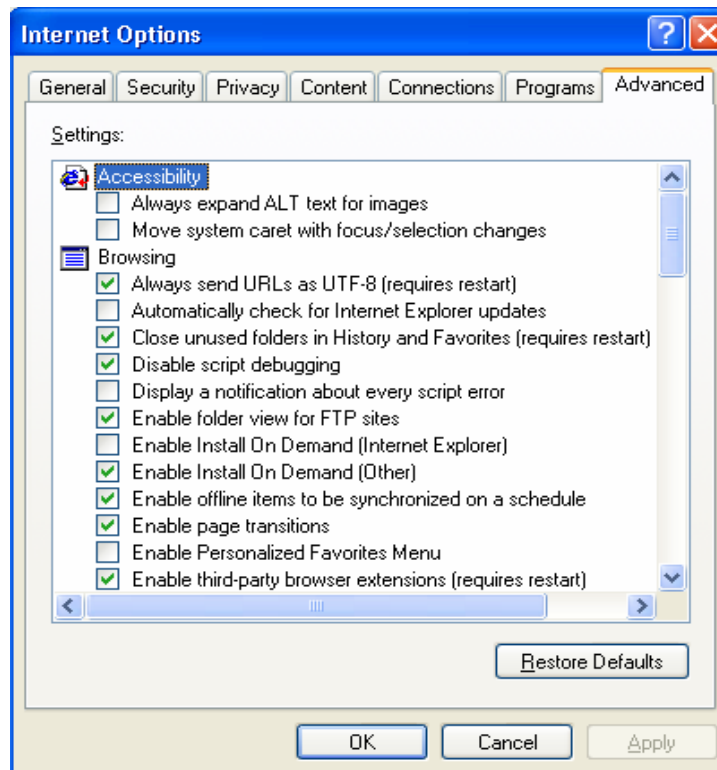
```
<P> William & Graham went to the fair
```

e. Ký tự khoảng trắng: &nbsp;**I.6. MỘT SỐ THAO TÁC TRONG CỬA SỔ TRÌNH DUYỆT**

- Cách load lại trang Web: Click biểu tượng Refresh (F5) trên thanh công cụ.
- Chỉnh sửa size chữ hiển thị trên trang: Chọn Menu View->Text size
- Chỉnh lại font chữ: Chọn Menu View->EnCoding
- Trong trường hợp trang Web không hiển thị được Font tiếng Việt:
- Chọn menu Tool chọn Internet Options->Chọn Tab Fonts→ chọn Font tiếng Việt
- Nếu chọn rồi mà không hiển thị được font tiếng Việt thì chọn Menu View→>EnCoding→chọn các font như User defined,Vietnamese....
- Các tùy chọn khác cho trang Web: Tools → Internet option: Không Load hình xuống, định dạng liên kết,...
- Chọn trang web mặc định khi mở trình duyệt



Hình 1



Hình 2

- *Tab General(Hình 1)*: thiết lập các tùy chọn cho trình duyệt
- *Click nút Use Current*: chọn trang hiện tại để load lên mỗi lần khởi động IE
- *Use Default*: địa chỉ trang Web mặc định mỗi khi mở trình duyệt  
Ví dụ mỗi khi mở IE thì tự động hiển thị trang Web Yahoo trong ô Address nhập: <http://yahoo.com> , nếu chọn Use Blank thì hiển thị trang trắng .
- *History*: lưu lại các trang web đã duyệt qua tại máy Client và thông tin đăng nhập của user hiện hành...
- Nếu không muốn lưu lại: Chọn Delete Cookies và Delete Files.
- Có thể thiết lập khoảng thời gian lưu trữ trang trong đối tượng History bằng cách thay đổi giá trị trong ô “Days to keep pages in history” .
- Nếu muốn xoá đối tượng này thì nhấn Clear history.
- *Tab Advance(Hình 2)*: có thể chọn các tùy chọn khác như:
  - Ngăn chặn không cho tải hình xuống trang web
  - Màu liên kết, cách thể hiện liên kết trên trang
  - Copy hình ảnh từ trang Web: Click phải vào hình ảnh cần sao chép rồi chọn Save picture as, hoặc Save background as,...
  - Load về trang Web bao gồm các hiệu ứng, script, hình ảnh... chứa trên trang: Chọn Menu File → Save As → Chọn vị trí lưu file → Save.
  - Hiệu chỉnh trang Web: View source->hiệu chỉnh→chọn File → Save để lưu lại→ F5 để cập nhật lại nội dung vừa hiệu chỉnh

## CHƯƠNG II: SIÊU LIÊN KẾT-HÌNH ẢNH

### II.1. GIỚI THIỆU SIÊU LIÊN KẾT

#### II.1.1. Siêu liên kết:

Khả năng chính của HTML là hỗ trợ các siêu liên kết. Một siêu liên kết cho phép người truy cập có thể đi từ trang web này đến trang web khác. Một liên kết gồm 3 phần:

- *Nguồn*: chứa nội dung hiển thị khi người dùng truy cập đến, có thể là một trang web khác, một đoạn film, một hình ảnh hoặc một hộp thoại để gửi mail...
- *Nhãn*: có thể là dòng văn bản hoặc hình ảnh để người dùng click vào khi muốn truy cập đến liên kết, nếu nhãn là văn bản thì thường được gạch dưới
- *Đích đến* (target): xác định vị trí để nguồn hiển thị.

#### II.1.2. Các loại liên kết

- *Internal Hyperlink*: (Liên kết trong) là các liên kết với các phần trong cùng một tài liệu hoặc liên kết các trang trong cùng một web site.
- *External Hyperlink* (Liên kết ngoài) là các liên kết với các trang trên web site khác.

### II.2. TẠO SIÊU LIÊN KẾT

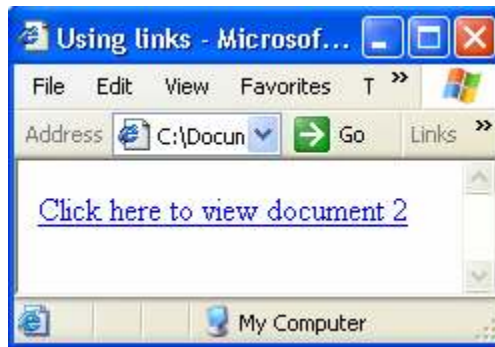
Cú pháp:

`<A HREF="URL"> Nhãn </A>`

- Dùng URL tương đối để liên kết đến các trang trong cùng một website

Ví dụ:

```
<HTML>
  <HEAD> <TITLE> Using links</TITLE></HEAD>
  <BODY>
    <A HREF = "Doc2.htm">Click here to view document 2</A>
  </BODY>
</HTML>
```



- Dùng URL tuyệt đối để liên kết đến các trang trong website khác

Ví dụ:

`<A href="http://www.google.com"> liên kết đến Google</A>`

#### II.2.1. Liên kết với các phần trong cùng một trang web

- Nếu nội dung của trang quá dài thì nên tạo các Bookmark để nhảy đến một phần cụ thể nào đó trên chính trang web hiện hành.
- Cách tạo liên kết đến các phần trong cùng trang: gồm 2 bước

✧ Tạo Bookmark:

**<A name="tên Bookmark"> Nhãn </A> Nội dung**

✧ Tạo liên kết đến Bookmark:

**<A Href ="#tên Bookmark">Nhãn của text liên kết</A>**

Ví dụ :

<HTML>

<HEAD><TITLE> Using htm links</TITLE> </HEAD>

<BODY>

<A HREF = "#Internet">Internet</A><BR>

<A HREF = "#HTML">Introduction to HTML</A><BR>

**<A name = "Internet">Internet</A>**

Internet là một mạng của các mạng. Nghĩa là, các mạng máy tính được liên kết với các mạng khác, nối các nước và ngày nay là toàn cầu. Giao thức truyền thông là TCP/IP cung cấp liên kết với tất cả các máy tính trên thế giới

**<A name = "HTML">Introduction to HTML</A><BR>**

Ngôn ngữ đánh dấu siêu văn bản là ngôn ngữ chuẩn mà web sử dụng để tạo và nhận ra tài liệu. Mặc dù không phải là một tập con của ngôn ngữ nâng cấp tiêu chuẩn tổng quát (SGML), ngôn ngữ đánh dấu siêu văn bản cũng có liên quan với SGML. SGML là một phương pháp trình bày các ngôn ngữ định dạng tài liệu. HTML là ngôn ngữ đánh dấu được sử dụng để tạo tài liệu HTML. Các hướng dẫn chỉ rõ một trang web nên được hiển thị như thế nào trong trình duyệt

</BODY>

</HTML>→ Kết quả trên trình duyệt



## II.2.2. Liên kết với một Bookmark ở một tài liệu khác

Cú pháp:

```
<A href="http://www.site.com/path/Page.htm#tên Bookmark">
```

Ví dụ:

- Trang main.htm

```
<HTML>
```

```
<HEAD><TITLE> Main document</TITLE></HEAD>
```

```
<BODY>
```

```
<A HREF = "C:\Doc1.htm#Internet">Internet</A><br>
```

```
<A HREF = "C:\Doc1.htm#HTML">Introduction to
```

```
HTML</A><br>
```

```
</BODY>
```

```
</HTML>
```

- Trang Doc1.htm

```
<HTML>
```

```
<HEAD><TITLE>Using Links</TITLE></HEAD>
```

```
<BODY>
```

```
<A name = "Internet">Internet</A><BR>
```

Internet là một mạng của các mạng. Nghĩa là, mạng máy tính được liên kết với các mạng khác, nối với các nước và ngày nay là toàn cầu. Giao thức truyền TCP/IP cung cấp liên kết với tất cả các máy tính trên thế giới.

```
<A name = "HTML">Introduction to HTML</A><BR>
```

Ngôn ngữ đánh dấu siêu văn bản là ngôn ngữ chuẩn mà web sử dụng để tạo và nhận ra tài liệu. Mặc dù không phải là một tập con của ngôn ngữ nâng cấp tiêu chuẩn tổng quát (SGML), ngôn ngữ đánh dấu siêu văn bản cũng có liên quan với SGML. SGML là một phương pháp trình bày các ngôn ngữ định dạng tài liệu. HTML là ngôn ngữ đánh dấu được sử dụng để tạo tài liệu HTML.

```
</BODY>
```

```
</HTML>
```

## II.2.3. Liên kết đến hộp thư e-mail

Cú pháp:

```
<A href="mailto:địa chỉ Email">Nhãn</A>
```

- Nếu siêu liên kết đặt ở cuối trang thì dùng tag <ADDRESS>

Cú pháp:

```
<Address><A href="mailto:địa chỉ Email">Nhãn</A></Address>
```

## II.3. HÌNH ẢNH TRÊN TRANG WEB:

### II.3.1. Các loại ảnh :

- a) **Ảnh .Gif** (Graphics Interchange Format): được sử dụng phổ biến nhất trong các tài liệu HTML, để chuyên tải, ngay cả các kết nối sử dụng MODEM tốc độ chậm, hỗ trợ 256 màu GIF. Các file GIF được định dạng không phụ thuộc phần nền
- b) **Ảnh JPEG** (Joint PhotoGraphic Expert Group) có phần mở rộng .JPG, là loại ảnh nén mất thông tin, nghĩa là ảnh sau khi bị nén không giống như ảnh gốc. Tuy nhiên, trong quá trình phát lại thì ảnh cũng tốt gần như ảnh gốc. JPEG hỗ trợ hơn 16 triệu màu và thường được sử dụng cho các ảnh có màu thực.



## c) Ảnh PNG (Portable Network Graphics) nén không mất dữ liệu

## II.3.2. Chèn hình ảnh

## Cú pháp:

**<IMG Src=URL Border=n Alt="Nội dung thay thế">**

URL: địa chỉ của tập tin hình ảnh, thường sử dụng địa chỉ tương đối, ví dụ:

`` không phụ thuộc vị trí của tập tin ảnh trên đĩa

n: độ dày của đường viền, tính bằng pixel

Alt: Nội dung thay thế sẽ hiển thị khi hình không load được, hoặc khi đưa chuột ngang qua hình

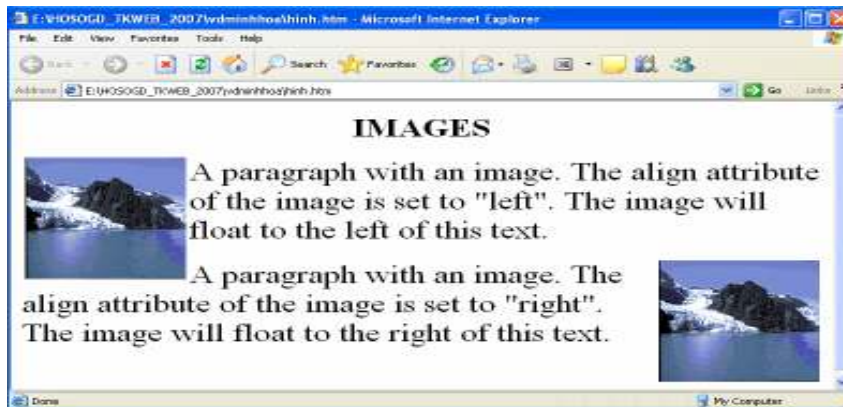
## II.3.3. Các thuộc tính của ảnh:

## a) Dàn văn bản quanh hình ảnh:

**<IMG SRC=URL Align= left> Nội dung văn bản quanh hình ảnh**

**<IMG SRC=URL Align= Right> Nội dung văn bản quanh hình ảnh**

## Ví dụ:



## b) Kích thước ảnh:

**<IMG width=Value Height=Value>**

## Ví dụ:

```
<html>
  <head><title>Image</title></head>
  <body>
    
  </body>
</html>
```

## c) Chặn văn bản bao quanh hình:

Canh lề khi dàn văn bản xung quanh một ảnh sẽ tác động đến tất cả các văn bản sau đó nếu không chèn vào một dòng kẻ đặc biệt. Thuộc tính CLEAR trong tag BR làm cho văn bản không bắt đầu nếu lề cụ thể không bị xóa đi (nghĩa là tại cạnh dưới của ảnh)

## Cú pháp:

**<BR CLEAR=Right>** : Ngăn chặn văn bản dàn bên lề phải của ảnh

**<BR CLEAR=Left>** : Ngăn chặn văn bản dàn bên lề trái của ảnh

**<BR CLEAR=All>** : Ngăn chặn văn bản dàn hai bên lề của ảnh

**d) Thêm khoảng trống xung quanh ảnh**

Nếu không muốn văn bản dàn xung quanh lề trái của ảnh thì ta có thể thêm khoảng trống xung quanh ảnh

**Cú pháp:**

```
<IMG SRC=URL HSPACE=n VSPACE=m>
```

**HSPACE=n:** Khoảng trống được tính bằng pixel sẽ thêm vào cả bên phải và bên trái của ảnh

**VSPACE=m:** Khoảng trống được tính bằng pixel sẽ thêm vào cả bên trên và bên dưới của ảnh

**e) Canh lề cho ảnh:** Có thể canh lề cho ảnh so với một dòng văn bản trong một đoạn

**Cú pháp:**

```
<IMG SRC=URL ALIGN= "Direction">
```

Văn bản muốn canh lề so với ảnh

Direction: gồm các giá trị: top, bottom, middle, texttop

Ví dụ:

**II.3.4. Dùng ảnh làm liên kết:**

Có thể dùng hình ảnh để tạo một liên kết đến một trang khác, hoặc nếu có một ảnh lớn, bạn có thể tạo ảnh nhỏ hơn hoặc một biểu tượng cho nó để nó có thể hiển thị nhanh chóng trên trang web, sau đó tạo liên kết để đưa người truy cập đến ảnh có kích thước thật

**Cú pháp:**

```
<A HREF="Địa chỉ trang liên kết">
```

```
<IMG SRC=URL Alt="nội dung thay thế">Nhấn
```

```
</A>
```

**II.3.5. Bản đồ ảnh:**

Bản đồ ảnh là một ảnh trong trang web được chia ra làm nhiều vùng, mỗi vùng khi click vào sẽ liên kết đến một địa chỉ URL

**Cách tạo:** Trước hết phải chèn vào trang một ảnh và đặt nhãn cho ảnh

```
<IMG UseMap="Label">
```

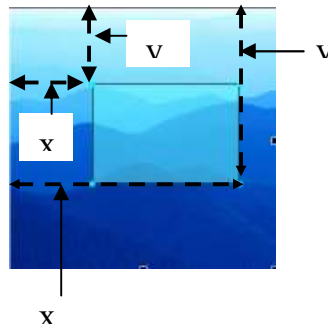
```
<Map Name="Label">
```

```
<Area Shape= "type" coords="x1,y1,x2,y2, ..." href="URL">
```

```
</Map>
```

Trong đó:

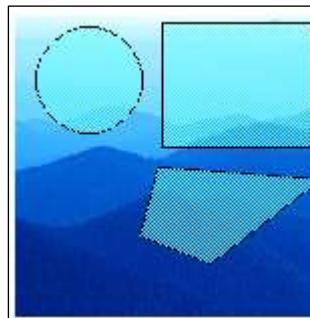
- Label: tên của bản đồ ảnh
- Type: hình dạng của các vùng trên ảnh, gồm các loại:
  - Rect: Vùng hình chữ nhật
  - Circle: Vùng hình tròn
  - Poly: Vùng hình đa giác
- Coords: toạ độ các đỉnh của hình
  - Rect: (x1, y1, x2, y2) là toạ độ 2 đỉnh chéo của vùng hình CN
  - Circle: (x, y, r) lần lượt là toạ độ tâm và bán kính của vùng hình tròn
  - Poly: (x1, y1, x2, y2, x3, y3, ...) là các đỉnh của vùng hình đa giác



**Ví dụ:**

```
<html>
<head>
<title>Image</title>
</head>
<body>

<map name="Map1">
<area shape="rect" coords="73,3,149,66" href="B1.htm">
<area shape="poly" coords="152,81,71,75,62,109,97,123" href="B3.htm">
<area shape="circle" coords="37,32,27" href="b2.htm">
</map>
</body>
</html>
```



### II.3.6. Hình nền :

Trong hầu hết các trang web thường sử dụng nền màu, với mục đích là làm nổi bật nội dung trang đó. Tuy nhiên cũng có thể sử dụng hình ảnh để làm nền bằng thuộc tính BACKGROUND của thẻ BODY.

**<BODY BACKGROUND= "bgimage.gif">**

## CHƯƠNG III: DANH SÁCH

### III.1. DANH SÁCH KHÔNG CÓ THỨ TỰ (Unorder List -UL)

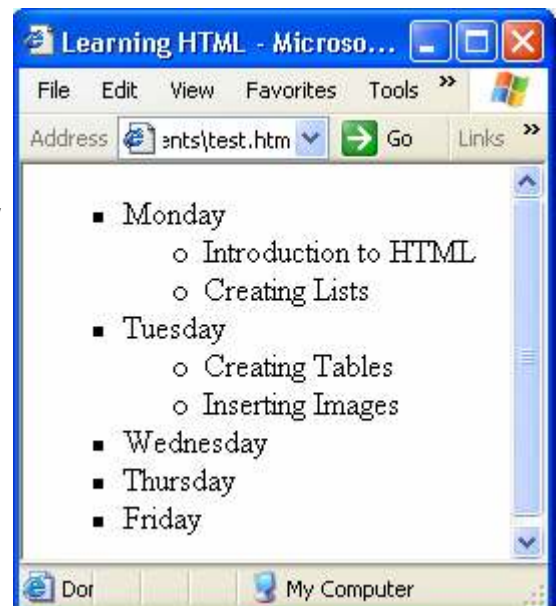
**Cú pháp:**

```
<UL Type= Shape1>
  <LI Type= Shape 2> Nội dung 1
  <LI Type= Shape 2> Nội dung 2
  ...
</UL>
```

- Shape 1, Shape 2 là loại bullet tự động đặt ở đầu dòng trong danh sách
- Shape 1: ảnh hưởng đến toàn danh sách
- Shape 2: ảnh hưởng đến một mục trong danh sách
- Các loại shape:
  - Circle: Bullet tròn, rỗng
  - Square: Bullet vuông
  - Disc: Bullet tròn không rỗng

**Ví dụ:**

```
<HTML>
  <HEAD><TITLE>Learning HTML</TITLE>
  <BODY>
    <UL type="Square">
      <LI>Monday
        <UL>
          <LI>Introduction to HTML
          <LI>Creating Lists
        </UL>
      <LI>Tuesday
        <UL>
          <LI>Creating Tables
          <LI>Inserting Images
        </UL>
      <LI>Wednesday
      <LI>Thursday
      <LI>Friday
    </UL>
  </BODY>
</HTML>
```



### III.2. DANH SÁCH CÓ THỨ TỰ (OrderList – OL)

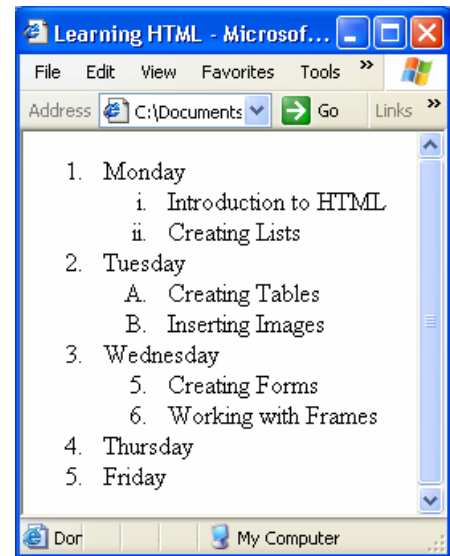
**Cú pháp:**

```
<OL Type=x Start =n >
  <LI Type =x1 Value=m> Nội dung 1
  <LI Type =x1 Value=m> Nội dung 2
  ...
</OL>
```

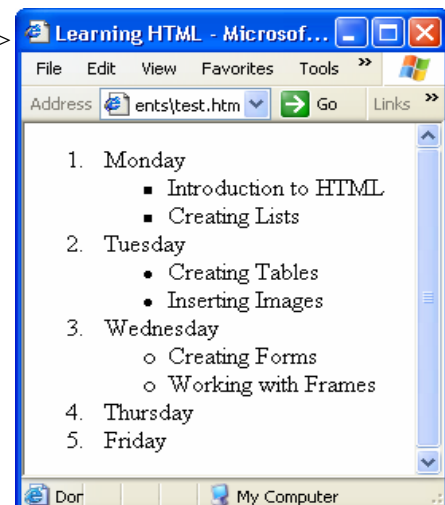
- **x**: loại ký tự muốn sử dụng trong danh sách gồm :
  - ✧ A: Chữ hoa
  - ✧ a: Chữ thường
  - ✧ I: Số la mã hoa
  - ✧ i: Số la mã thường
  - ✧ 1: Cho số mặc định
- **n**: giá trị đầu tiên của danh sách
- **x1**: là loại ký tự sử dụng cho dòng này và dòng tiếp theo, làm mất ảnh hưởng của x
- **m**: giá trị đầu tiên của dòng này, làm thay đổi giá trị của n

**Ví dụ 1:**

```
<HTML>
<HEAD><TITLE>Learning HTML</TITLE>
<BODY>
  <OL>
    <LI>Monday
      <OL>
        <LI TYPE = i>Introduction to HTML
        <LI TYPE = i>Creating Lists
      </OL>
    <LI>Tuesday
      <OL>
        <LI TYPE = A>Creating Tables
        <LI TYPE = A>Inserting Images
      </OL>
    <LI>Wednesday
      <OL START = 5>
        <LI >Creating Forms
        <LI >Working with Frames
      </OL>
    <LI>Thursday
    <LI>Friday
  </OL>
</BODY>
</HTML>
```

**Ví dụ 2:** Có thể lồng 2 loại danh sách có thứ tự và không có thứ tự vào nhau

```
<HTML>
<HEAD><TITLE>Learning HTML</TITLE></HEAD>
<BODY>
  <OL>
    <LI>Monday
      <UL>
        <LI>Introduction to HTML
        <LI>Creating Lists
      </UL>
    <LI>Tuesday
      <UL type='Disc'>
        <LI>Creating Tables
        <LI>Inserting Images
      </UL>
  </OL>
```



```

<LI>Wednesday
  <UL type='cycle'>
    <LI>Creating Forms
    <LI>Working with Frames
  </UL>
<LI>Thursday
<LI>Friday
</OL>
</BODY>
</HTML>

```

### III.3. DANH SÁCH ĐỊNH NGHĨA:

Trong HTML có một tag đặc biệt dùng để tạo danh sách định nghĩa dành riêng cho việc tra cứu, nhưng cũng thích hợp cho danh sách nào để nối một từ với một diễn giải dài.

**Cú pháp:**

```

<DL>
  <DT>Nhập từ muốn định nghĩa
  <DD>Nhập nội dung định nghĩa
  ...
</DL>

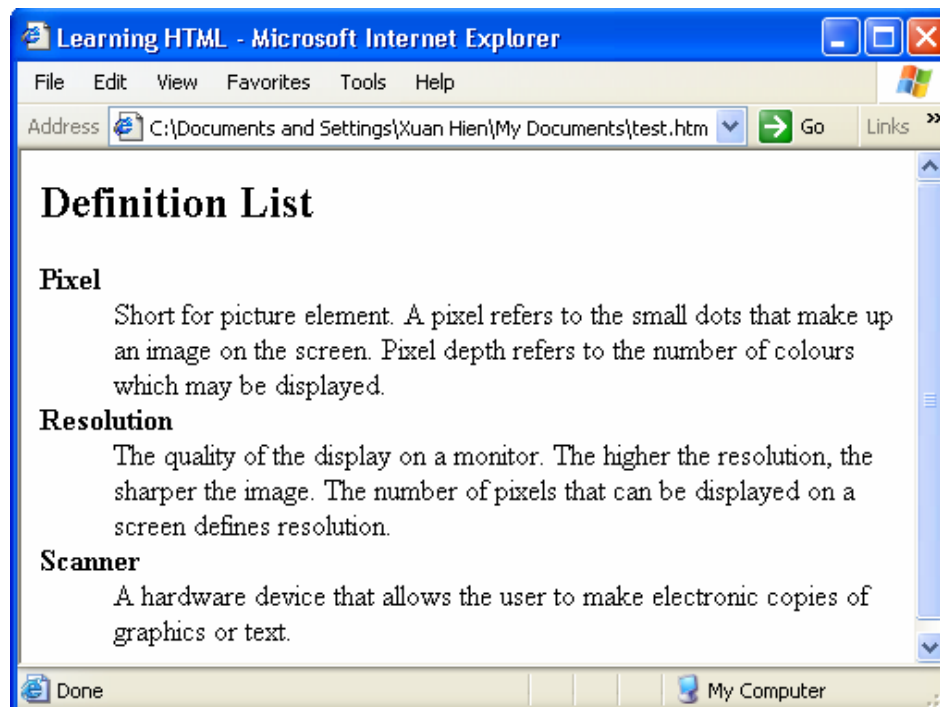
```

**Ví dụ:**

```

<HTML>
  <HEAD>
    <TITLE>Learning HTML</TITLE>
  </HEAD>
  <BODY>
    <h2> Definition List</h2>
    <DL>
      <DT>Pixel
      <DD> Short for picture element. A pixel refers to the small dots that
        make up an image on the screen. Pixel depth refers to the number of
        colours which may be displayed.
      <DT>Resolution
      <DD>The quality of the display on a monitor. The higher the
        resolution, the sharper the image. The number of pixels that can be
        displayed on a screen defines resolution.
      <DT>Scanner
      <DD> A hardware device that allows the user to make electronic
        copies of graphics or text.
    </DL>
  </BODY>
</HTML>

```



## CHƯƠNG IV: BẢNG VÀ TRÌNH BÀY TRANG

### IV.1. BẢNG

Bảng thường được sử dụng để tạo các văn bản nhiều cột hoặc phân chia trang thành nhiều vùng khác nhau rất tiện lợi trong thiết kế và trình bày trang web

#### Cú pháp:

```

<TABLE >
  {
    <TR>
      <TD>Nội dung trong ô 1</TD> ← Cột 1
      <TD>Nội dung trong ô 2</TD> ← Cột 2
      ...
      <TD>Nội dung trong ô n</TD>
    </TR>
    {
      <TR>
        <TD>Nội dung trong ô 1</TD>
        <TD>Nội dung trong ô 2</TD>
        ...
        <TD>Nội dung trong ô n</TD>
      </TR>
      ...
    }
  }
</TABLE>

```

- Tag <table> </table> : chỉ thị một bảng
- Tag <tr>.....</tr> : xác định một dòng của bảng
- Tag <td>.....</td>: xác định một ô chứa dữ liệu của bảng. Dữ liệu trong ô có thể là văn bản hoặc hình ảnh...

#### Ví dụ 1:

```

<HTML>
  <HEAD>
    <TITLE>TABLE</TITLE>
  </HEAD>
  <BODY >
    <table border="1">
      <TR>
        <TD>Cell 1</TD>
        <TD>Cell 2</TD>
        <TD>Cell 3</TD>
        <TD>Cell 4</TD>
      </TR>
    </table>
  </BODY>
</HTML>

```

→ 

Cell 1	Cell 2	Cell 3	Cell 4
--------	--------	--------	--------



**Ví dụ 2:**

```

<HTML>
  <HEAD>
    <TITLE>TABLE</TITLE>
  </HEAD>
  <BODY >
    <table border="1">
      <TR>
        <TD>Cell 1</TD>
      </TR>
      <TR>
        <TD>Cell 2</TD>
      </TR>
      <TR>
        <TD>Cell 3</TD>
      </TR>
      <TR>
        <TD>Cell 4</TD>
      </TR>
    </table>
  </BODY>
</HTML>

```

Cell 1
Cell 2
Cell 3
Cell 4

**Ví dụ 3:**

```

<HTML>
  <HEAD>
    <TITLE>TABLE</TITLE>
  </HEAD>
  <BODY >
    <table border="1">
      <TR>
        <TD>Cell 1</TD>
        <TD>Cell 2</TD>
      </TR>
      <TR>
        <TD>Cell 3</TD>
        <TD>Cell 4</TD>
      </TR>
    </table>
  </BODY>
</HTML>

```

Cell 1	Cell 2
Cell 3	Cell 4

**IV.2. CÁC THUỘC TÍNH:****IV.2.1. Thuộc tính của bảng****f) Thêm khung viền:**

**<Table Border =n>...<Table>**

n: độ dày của khung viền tính bằng Pixel

**g) Định màu của khung viền và màu nền:**

**<Table BorderColor= “Color” BgColor=”Color”>...</Table>**

**h) Tạo bóng :**

**<Table BorderColorDark= “Color”>** : Bóng đổ ở cạnh dưới và phải của bảng

**<Table BorderColorLight= “Color”>** : Bóng đổ cạnh trên trái của bảng

**i) Định chiều rộng và chiều cao của bảng:**

**<Table Width =n height=m>**, n là chiều rộng tính bằng Pixel

**j) Canh lề bảng:**

**<Table Align= left/ right/ center>...</table>**

**k) Thuộc tính Cellpadding và CellSpacing:**

**<Table CellSpacing =”value”>**: Khoảng cách giữa đường viền của các ô

**<Table CellPadding=”Value”>**: Khoảng cách giữa đường viền của ô với văn bản

**l) Tag tiêu đề của Table:**

**<Caption> tiêu đề </Caption>**

- Tag <Caption> nằm trong cặp Tag <Table>...</Table>

#### IV.2.2. Thuộc tính của cột

**a) Canh lề theo chiều ngang:**

**<Td Align=left/ right/center>...</Td>**

**b) Canh lề theo chiều đứng:**

**<Td Valign= Top/ Bottom/ Middle>...</Td>**

**c) Trộn ô:**

**<Td Colspan=n>**: trộn n cột

**<Td RowSpan=n>**: trộn n dòng

**d) Tag <th>:**

Có tác dụng như <td> nhưng làm cho dữ liệu trong ô được in đậm và canh giữa

```
<tr>
  <th> Nội dung </th>
</tr>
```

Ví dụ:

```
<TABLE border=2>
  <tr>
    <th> Cell 1 </th>
  </tr>
  <tr>
    <th> Cell 2 </th>
  </tr>
</TABLE>
```

Cell 1
Cell 2

**Ví dụ:**

```

<HTML>
  <HEAD>
    <TITLE>TABLE</TITLE>
  </HEAD>
  <BODY >
    <table border="5" CellSpacing =10 BorderColorDark=red width=50%>
      <TR>
        <TD>Cell 1</TD>
        <TD>Cell 2</TD>
      </TR>
      <TR>
        <TD>Cell 3</TD>
        <TD>Cell 4</TD>
      </TR>
    </table>
  </BODY>
</HTML>

```

Cell 1	Cell 2
Cell 3	Cell 4

**Ví dụ:**

```

<Table border="1" bgcolor= "fuschia" bordercolor="red" align="center" Width=50%
Height=30%>

```

```

  <caption> Properties of Table</caption>
  <tr>
    <th colspan="3"> Colspan</th>
  </tr>
  <tr>
    <th Rowspan="2"> Rowspan</th>
    <td align=center>Cell 1</td>
    <td align=center>Cell 2</td>
  </tr>
  <tr>
    <td align=center> Cell 3</td>
    <td align=center> Cell 4</td>
  </tr>
</table>

```

Properties of Table

Colspan		
Rowspan	Cell 1	Cell 2
	Cell 3	Cell 4

**Ví dụ:** Thiết kế một trang web như mẫu

Computer Model	
Tin tức	
Giai tri	
Quang cao	
The thao	

```

<html>
<head>
  <title> Trình bày trang</title>
</head>
<body>
<Table width="68%" height="135" border="1" cellspacing="0"
bordercolor="#990033">
  <tr>
    <th colspan="2" bgcolor="#FFCCFF">
      <div align="center">Computer Model </div>
    </th>
  </tr>
  <tr>
    <td width="24%" height="98" valign="top">
      <table width="100%" border="1" cellspacing="0">
        <tr>
          <td>Tin tức</td>
        </tr>
        <tr>
          <td>Giải trí</td>
        </tr>
        <tr>
          <td>Quảng cáo</td>
        </tr>
        <tr>
          <td height="23">Thể thao</td>
        </tr>
      </table>
    </td>
    <td width="76%" align="center">
      
    </td>
  </tr>
</table>
</body>
</html>

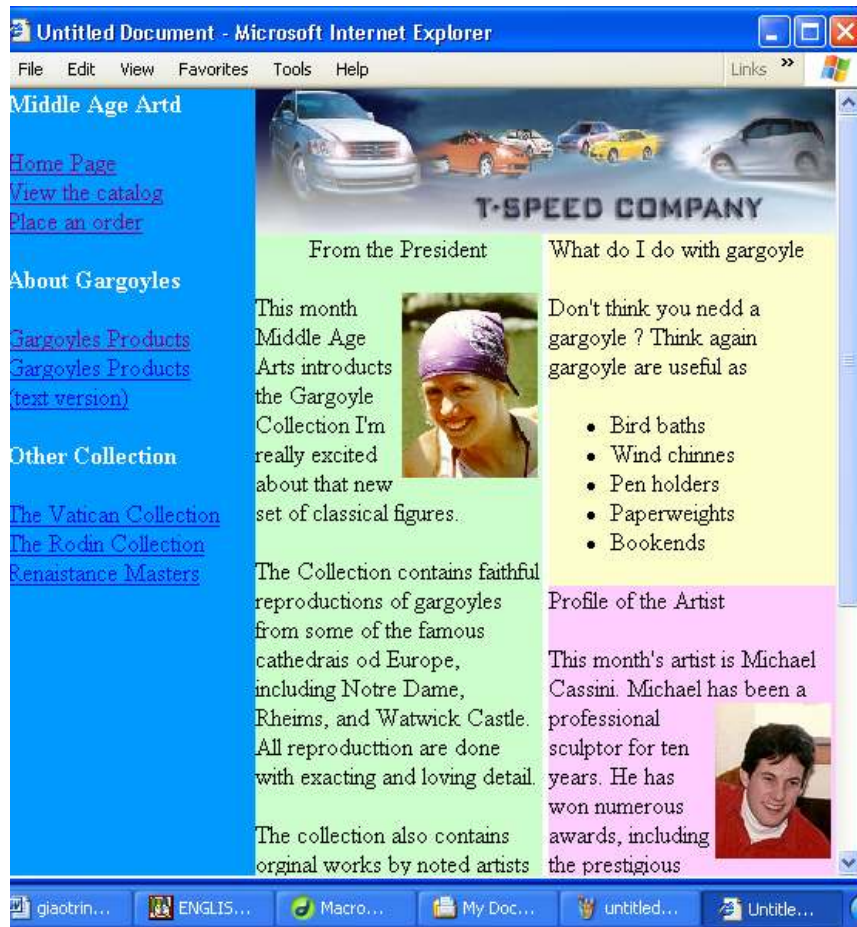
```

### IV.3. TRÌNH BÀY TRANG

Trong thực tế, bảng thường được sử dụng để trình bày bố cục cho toàn bộ trang web. Nếu muốn thiết kế một trang thể hiện văn bản trong cột dạng báo chí hoặc phân trang thành những vùng có chủ đề khác nhau, thì bảng là một công cụ cần thiết. Một trong những nét đặc trưng hữu dụng của bảng đó là trong mỗi table cell bạn có thể sử dụng bất kỳ tag HTML nào, ví dụ bạn có thể chèn một tag <H1> trong một cell hoặc một danh sách có thứ tự các mục hoặc có thể chèn một bảng con trong một bảng khác...

#### Ví dụ :

Cần thiết kế trang web gồm nhiều vùng với những chủ đề khác nhau như hình dưới đây, thì bảng là công cụ hữu hiệu



Bước 1: Tạo một table thứ nhất gồm 1 dòng và 2 cột

```

<table>
  <tr>
    <td>
      <!--Danh sách các mục liên kết-->
    </td>
    <td>
      <!--Table 2 -->
    </td>
  </tr>
</table>
    
```

Danh sách các mục liên kết	table 2
----------------------------	---------

Bước 2: tạo table thứ 2 gồm 3 dòng và 2 cột

```

<table>
  <tr>
    <td colspan =2>
      <!--Chèn hình logo-->
    </td>
  </tr>
  <tr>
    <td rowspan =2>
      <!--Nội dung 1 -->
    </td>
    <td>
      <!--Nội dung 2 -->
    </td>
  </tr>
  <tr>
    <td>
      <!--Nội dung 3 -->
    </td>
  </tr>
</table>

```

danh sách các mục liên kết  ~~~~~ ~~~~~ ~~~~~ ~~~~~	Logo	
	Nội dung 1 ~~~~~ ~~~~~ ~~~~~ ~~~~~	Nội dung 2 ○ ~~~~~ ○ ~~~~~ ○ ~~~~~ ○ ~~~~~
	Nội dung 3 ~~~~~ ~~~~~ ~~~~~	

## CHƯƠNG V: FRAME

### V.1. KHÁI QUÁT VỀ FRAME

- Có thể phân chia một trang thành các khung, cho phép người truy cập cùng một lúc có thể xem nhiều trang mà không cần cuộn màn hình, mỗi khung chứa một trang web riêng
- Nếu tài trang sử dụng Frame thì không sử tag Body

### V.2. CÁCH TẠO MỘT FRAME LAYOUT

```
<HTML>
  <HEAD>
    <TITLE>Page Title</TITLE>
  </HEAD>
  <FRAMESET>
    Frame Definitions
  </FRAMESET>
</HTML>
```

#### V.2.1. Các dạng frame : Tag <FRAMESET> có 2 thuộc tính ROWS và COLS

##### a) Tạo frame theo dòng

Cú pháp:

```
<HTML>
  <HEAD><TITLE>Nội dung tiêu đề</TITLE></HEAD>
  <Frameset Rows="a, b..." >
    <Frame name="Name1" Src="Content1.htm">
    <Frame name="Name2" Src="Content2.htm">
    ....
    <Frame name="Name_n" Src="Content_n.htm">
  </Frameset>
</HTML>
```

Trong đó:

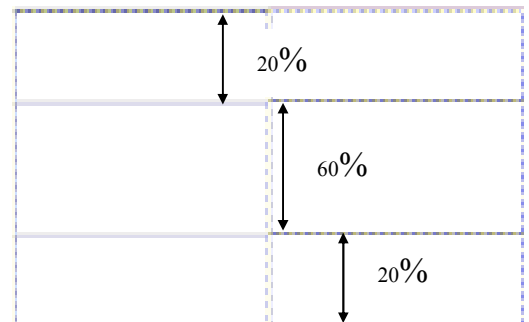
*a, b*: là độ cao của các dòng thứ 1, thứ 2 ..., có thể tính bằng pixel hoặc bằng %

*Name*: tên khung, (xác định chức năng của khung)

*Content.htm*: địa chỉ trang web xuất hiện đầu tiên trong khung

**Ví dụ:**

```
<HTML>
  <HEAD><TITLE>Frame</TITLE></HEAD>
  <Frameset Rows=20%, 60%, 20% >
    <Frame name="Head" Src="head.htm">
    <Frame name="Content1" Src="Content1.htm">
    <Frame name="Content2" Src="Content2.htm">
  </Frameset>
</HTML>
```



<Frameset Rows=20%, 60%, 20%

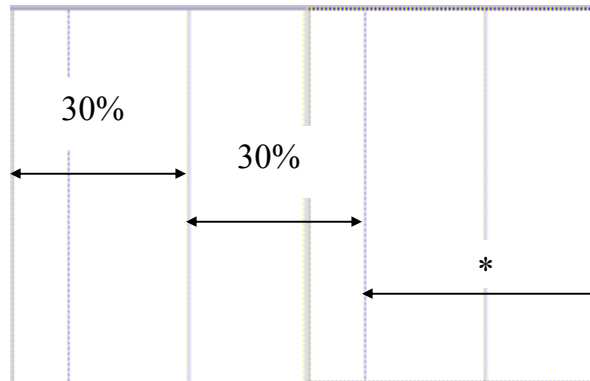
b) Tạo frame theo cột

Cú pháp:

```
<HTML>
  <HEAD><TITLE>Nội dung tiêu đề</TITLE></HEAD>
  <Frameset Cols="a, b..." >
    <Frame name="Name1" Src="Content1.htm">
    <Frame name="Name2" Src="Content2.htm">
    ....
    <Frame name="Name_n" Src="Content_n.htm">
  </Frameset>
</HTML>
```

Ví dụ:

```
<HTML>
  <HEAD><TITLE>Frame</TITLE></HEAD>
  <Frameset Cols=30%, 30%, * >
  <Frame name="Baner" Src="head.htm">
  <Frame name="Content1" Src="Content1.htm">
  <Frame name="Content2" Src="Content2.htm">
  </Frameset>
</HTML>
```



```
<Frameset Cols=30%, 30%, * >
```

**V.2.2. Các thuộc tính của Frame:**

- a) Noresize: Không đổi kích thước
- b) Scrolling: có/không có thanh cuộn  
*Auto*: Xuất hiện thanh cuộn khi nội dung dài  
*Yes*: luôn xuất hiện thanh cuộn  
*No*: không xuất hiện thanh cuộn

Ví dụ:

```
<frameset rows="80,*" cols="*" frameborder="NO" border="0" >
  <frame noresize src="topFrame" scrolling="NO" >
  <frame noresize src="leftFrame" scrolling="NO" >
</frameset>
```

- c) Frameborder: đường viền của khung mặc định là 1, muốn giữa các khung không còn đường viền thì trong tag Frameset nhập thêm Border=0,
- d) Marginwidth: hiệu chỉnh khoảng cách từ nội dung đến lề trái và phải của khung (tính bằng pixel)
- e) Marginheight: hiệu chỉnh khoảng cách từ nội dung đến lề trên và dưới của khung (tính bằng pixel)



### V.2.3. Các frame lồng nhau:

```

<Frameset >
  <Frame name="name" src="Page.htm">
  <Frameset>
    <Frame name="name" src="Page.htm">
    ...
  </Frameset>
  ...
</Frameset>

```

### V.2.4. Liên kết frame:

Trang đầu tiên của khung được chỉ ra trong thuộc tính SRC, ta có thể chỉnh các trang khác cùng xuất hiện trong khung đó bằng cách chỉ ra vị trí trang đích  
 Tại trang muốn tạo liên kết với khung, ta nhập cú pháp:

```
<a Href="Page.htm Target="name">
```

Nhãn mục liên kết

```
</a>
```

Trong đó :

*Target=Name* : tên của khung mà trang muốn liên kết đến trong tag <Frame>

*Page.htm*: trang hiển thị trong khung liên kết

#### ❖ Tag <Base>:

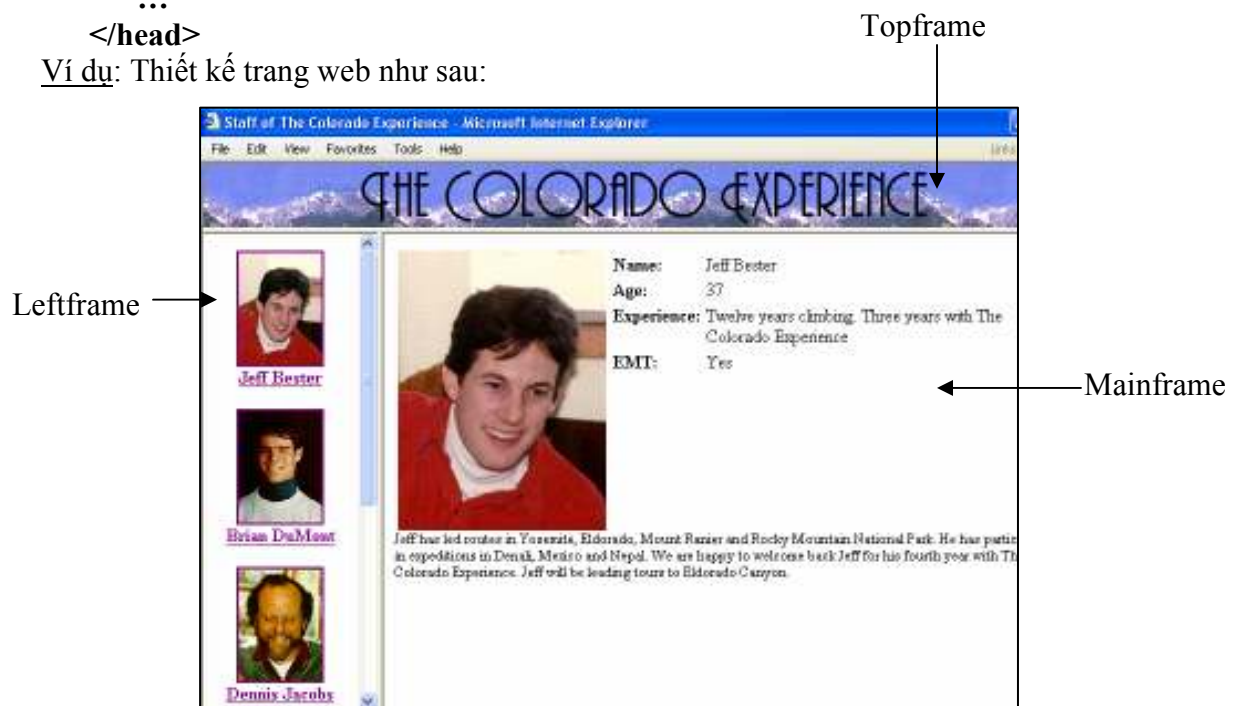
Nếu có nhiều liên kết đến các trang xuất hiện trong cùng một khung thì thuộc tính target mặc định đặt trong tag <Base>

```

<Head>
  <Base target="name">
  <a href="URL">Nhãn liên kết</a>
  ...
</head>

```

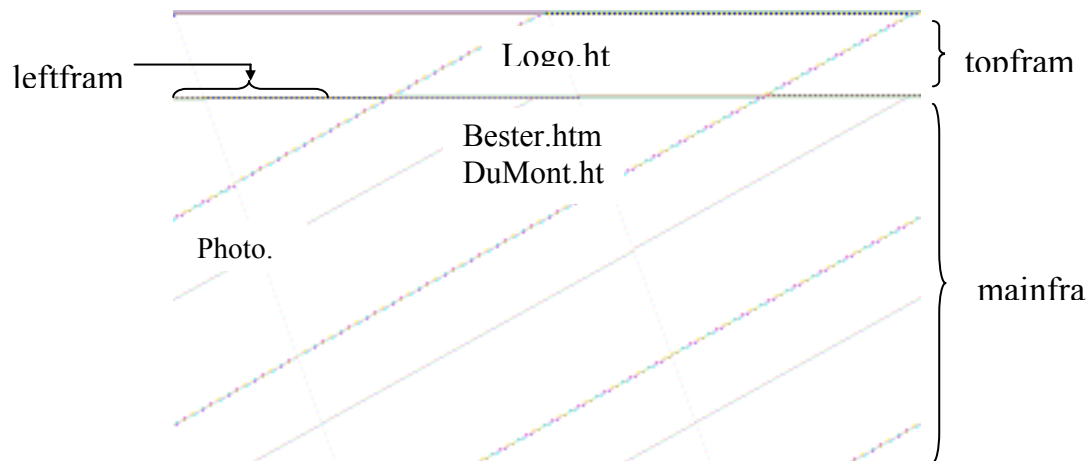
Ví dụ: Thiết kế trang web như sau:



Cách thực hiện:

**Trang chính:** Chia trang thành 3 khung: topframe, leftframe và mainframe. Trang Photo.htm đặt trong left frame, logo.htm đặt trong Topframe, Bester.htm, DuMont.htm, Jacobs.htm đặt trong mainframe

```
<HTML>
<HEAD><TITLE>Staff of The Colorado Experience</TITLE>
<FRAMESET ROWS="60,*">
  <!-- Company logo --->
  <FRAME SRC="Logo.htm" marginheight=1 noresize scrolling=NO
  name=Topframe>
  <!-- Nested frames --->
  <FRAMESET COLS="140,*">
    <!-- A list of staff photos --->
    <FRAME SRC="Photos.htm" NAME=Leftframe>
    <!-- Individual staff biographies --->
    <FRAME SRC="Bester.htm" NAME=Mainframe>
  </FRAMESET>
</FRAMESET>
</HEAD>
</HTML>
```



Trong đó các trang Logo.htm, Photos.htm, Bester.htm phải được tạo trước

### Photos.htm

```
<HTML>
<HEAD><TITLE>Staff hypertext links</TITLE>
  <BASE TARGET=Mainframe>
</HEAD>
<BODY>
  <CENTER>
  <A HREF="Bester.htm"><IMG SRC="Bester.jpg" width=75 height=101> <BR>
  Jeff Bester</A><BR><BR>
  <A HREF="DuMont.htm"><IMG SRC="DuMont.jpg" width=75 height=101>
  <BR> Brian DuMont</A><BR><BR>
  <A HREF="Jacobs.htm"><IMG SRC="Jacobs.jpg" width=75 height=101><BR>
  Dennis Jacobs</A><BR><BR>
  </CENTER>
</BODY>
</HTML>
```

**Bester.htm**

&lt;HTML&gt;

&lt;HEAD&gt;&lt;TITLE&gt;Jeff Bester&lt;/TITLE&gt;&lt;/HEAD&gt;

&lt;BODY&gt;

&lt;IMG SRC="Bester2.jpg" WIDTH=90 HEIGHT=125 ALIGN=LEFT&gt;

&lt;FONT SIZE=2&gt;

&lt;TABLE&gt;

&lt;TR&gt;

&lt;TD VALIGN=TOP&gt;&lt;B&gt;Name:&lt;/B&gt;&lt;/TD&gt;

&lt;TD VALIGN=TOP&gt;Jeff Bester&lt;/TD&gt;

&lt;/TR&gt;

&lt;TR&gt;

&lt;TD VALIGN=TOP&gt;&lt;B&gt;Age:&lt;/B&gt;&lt;/TD&gt;

&lt;TD VALIGN=TOP&gt;37&lt;/TD&gt;

&lt;/TR&gt;

&lt;TR&gt;

&lt;TD VALIGN=TOP&gt;&lt;B&gt;Experience:&lt;/B&gt;&lt;/TD&gt;

&lt;TD VALIGN=TOP&gt;Twelve years climbing. Three years with The Colorado Experience&lt;/TD&gt;

&lt;/TR&gt;

&lt;TR&gt;

&lt;TD VALIGN=TOP&gt;&lt;B&gt;EMT:&lt;/B&gt;&lt;/TD&gt;

&lt;TD VALIGN=TOP&gt;Yes&lt;/TD&gt;

&lt;/TR&gt;

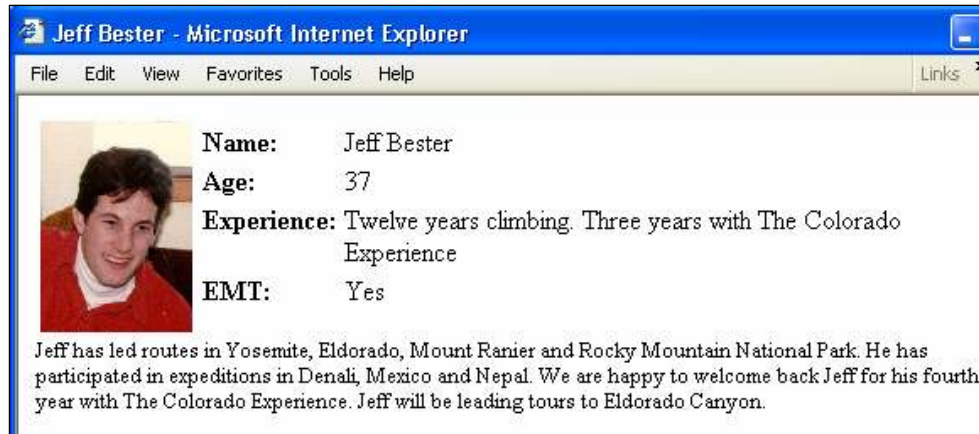
&lt;/TABLE&gt;&lt;BR CLEAR=LEFT&gt;

Jeff has led routes in Yosemite, Eldorado, Mount Ranier and Rocky Mountain National Park. He has participated in expeditions in Denali, Mexico and Nepal. We are happy to welcome back Jeff for his fourth year with The Colorado Experience. Jeff will be leading tours to Eldorado Canyon.

&lt;/FONT&gt;

&lt;BODY&gt;

&lt;/HTML&gt;



Các trang DuMont.htm, Jacobs.htm cũng thiết kế tương tự.

### Logo.htm

```
<HTML>
  <HEAD><TITLE>HEAD</TITLE></HEAD>
  <BODY BACKGROUND="Mountain.jpg">
    <IMG SRC="TCELogo.gif" WIDTH=550 HEIGHT=60>
  </BODY>
</HTML>
```



### V.2.5. Phần tử NOFRAMES

Phần tử NOFRAMES được sử dụng để chỉ nội dung thay thế cho frame khi trình duyệt không hỗ trợ frame.

#### Cú pháp:

```
<HTML>
  <HEAD><TITLE>Page title</TITLE></HEAD>
  <FRAMESET>
    Frame Definitions
  </TRAMESET>
  <NOFRAME>
    <BODY>
      Page Layout
    </BODY>
  </NOFRAME>
</HTML>
```

### V.2.6. Phần tử IFRAME

Nếu muốn trộn văn bản và khung trong cùng một trang thì phải tạo một khung bên trong trang bằng tag **<iframe>**, khi trình duyệt không hỗ trợ thì nội dung trong IFRAME sẽ bị trả lại

#### Cú pháp:

– Tại vị trí muốn chèn frame, nhập cú pháp:

**<Iframe Src="Page.htm" Name="name" Width= x Height=y Align=left/ right>**

Nội dung thay thế khi trình duyệt không chấp nhận khung

**</Iframe>**

Trong đó:

- *Page.htm*: là trang đầu tiên xuất hiện trong khung
- *Name*: tên của khung
- *x, y*: kích thước của khung
- *Align*: canh lề

Ví dụ: thiết kế trang web có dạng sau:



Phần tử Iframe

```
<html>
<head>
    <title>Untitled Document</title>
</head>
<body>
    <center></center>
    <iframe width="350" height="150" align="right" src="Bester.htm">
</iframe>
    <h1> <font face="Arial, Helvetica, sans-serif">Staff </font></h1>
    The staff at the Colorado Experience is here to helpwith all of your
    climbing needs. All of our instructors are fully qualified, with yearsof
    climbing and teaching experience. Scroll through the biographies at the
    right for more information
</body>
</html>
```

## CHƯƠNG VI: FORM

### VI.1. GIỚI THIỆU FORM

#### VI.1.1. Sử dụng Form: Form được sử dụng khi cần:

- Thu thập thông tin tên, địa chỉ, số điện thoại, email, ... để đăng ký cho người dùng vào một dịch vụ hoặc một sự kiện
- Tập hợp thông tin để mua hàng
- Thu thập thông tin phản hồi về một Website
- Cung cấp công cụ tìm kiếm trên website

#### VI.1.2. Cách tạo:

Cú pháp:

```
<Form Method=(Post Get) Action=script.url>
    Nội dung của Form
</Form>
```

Trong đó:

-*Method*: xác định phương thức đưa dữ liệu lên máy chủ, có 2 giá trị :Post và Get

- Nếu giá trị là GET thì trình duyệt sẽ tạo một câu hỏi chứa trang URL, một dấu hỏi và các giá trị do biểu mẫu tạo ra. Trình duyệt sẽ đổi script của câu hỏi thành kiểu được xác định trong URL để xử lý.
- Nếu giá trị là POST thì dữ liệu trên biểu mẫu sẽ được gửi đến script như một khối dữ liệu

-*Action*: là địa chỉ của script sẽ thực hiện khi form được submit

### VI.2. CÁC PHẦN TỬ CỦA FORM:

Các phần tử của form thường sử dụng trên trang web gồm

- **Input boxes**: nhập dữ liệu dạng text và number
- **Radio buttons**: dùng để chọn một tùy chọn trong danh sách
- **Selection lists**: dùng cho một danh sách dài các lựa chọn, thường là trong Drop-down list box
- **Check boxes**: chỉ định một item được chọn hay không
- **Text area**: một text box có thể chứa nhiều dòng
- **Submit và Reset button**: để gửi form đến CGI script vừa để reset form về trạng thái ban đầu

#### VI.2.1. Input boxes

Là một hộp dòng đơn dùng để nhập văn bản hoặc số. Để tạo các input boxes, sử dụng tag <INPUT>, tag <INPUT> còn được sử dụng cho nhiều loại field khác trên form.

Cú pháp:

```
<FORM>
    <INPUT TYPE=Object NAME=Text>
</FORM>
```

Các giá trị của thuộc tính TYPE: Mặc định giá trị của TYPE là text, nếu trong tag <INPUT> không nhập thuộc tính TYPE thì loại input boxes là text

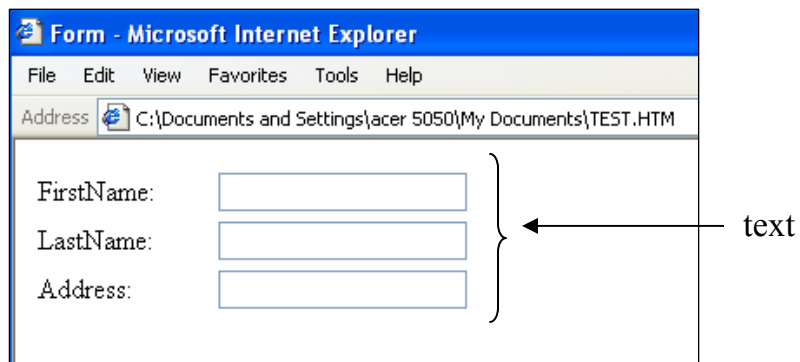
- TEXT
- PASSWORD
- CHECKBOX
- RADIO
- HIDDEN
- RESET
- SUBMIT
- TEXTAREA
- BUTTON
- IMAGE

**Ví dụ:**

```

<html>
<head><title>Form</title></head>
<body>
  <form>
    <table>
      <tr>
        <td width=100>FirstName: </td>
        <td><input name =Firstname></td>
      </tr>
      <tr>
        <td>LastName: </td>
        <td><input name =Lastname></td>
      </tr>
      <tr>
        <td>Address: </td>
        <td><input name =Address></td>
      </tr>
    </table>
  </form>
</body>
</html>

```



### 1. **Text box:** Hộp văn bản, do người sử dụng nhập vào

**Cú pháp:**

**<Input Type="Text" Value="Value" Name="name" Size=n Maxlength=m>**

- Name : tên dữ liệu đầu vào server
- Value: Dữ liệu ban đầu có sẵn trong text box
- Size: chiều rộng của text box tính bằng số ký tự (mặc định là 20)
- Maxlength: số ký tự tối đa có thể nhập vào text box

2. **Tạo hộp Password:** Những ký tự nhập vào hiển thị dưới dạng dấu chấm, thông tin sẽ không bị mã hoá khi gửi lên server

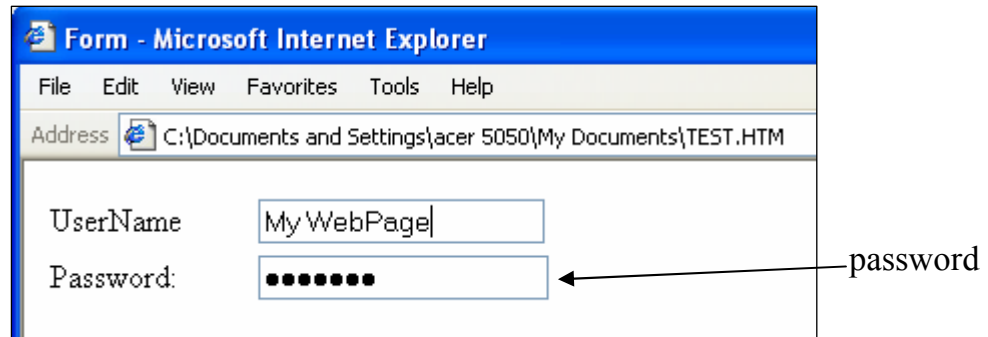
**Cú pháp:**

**<Input Type="password" Name="name" size=n maxlength=n>**

- Size: chiều rộng của hộp Password, tính bằng ký tự
- Maxlength: Số ký tự tối đa có thể nhập vào hộp Password

**Ví dụ:**

```
<html>
  <head><title>Form</title></head>
  <body>
    <form><table>
      <tr>
        <td width=100>UserName </td>
        <td><input name =UserName></td>
      </tr>
      <tr>
        <td>Password: </td>
        <td><b><input Type='password' name =Password></b></td>
      </tr>
    </table></form>
  </body>
</html>
```



3. **Checkbox:** Hộp chọn, người xem có thể đánh dấu nhiều checkbox trong cùng 1 bộ

**Cú pháp:**

**<Input Type="Checkbox" Name="Name" Value="Value" Checked> Nhãn**

- Name: tên của checkbox
- Value: xác định mỗi giá trị cho mỗi hộp checkbox được gửi cho server khi người xem đánh dấu vào checkbox
- Checked: thuộc tính để hộp check box được chọn mặc định

**Ví dụ:**

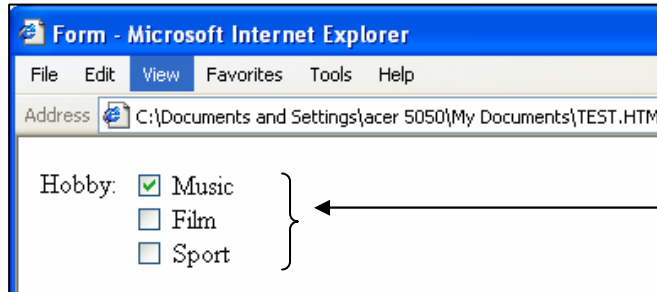
```
<html>
  <head><title>Form</title></head>
  <body>
    <form><table>
      <tr>
        <td width=50 valign=top>Hobby: </td>
        <td>
          <input Type='Checkbox' Name='st' Value='nhac' Checked> Music<br>
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```



```

<Input Type='Checkbox' Name='st' Value='film'> Film<br>
<Input Type='Checkbox' Name='st' Value='thethao' > Sport
</td>
</tr>
</table></form>
</body>
</html>

```



4. **Radio button:** Cho phép người xem chỉ chọn một tùy chọn tại mỗi thời điểm

**Cú pháp:**

```
<input type="radio" name="name" value="Value" checked>Nhãn
```

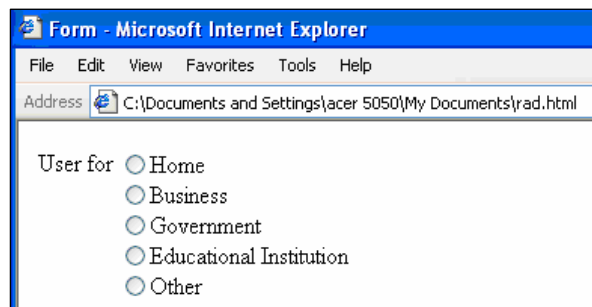
- Name: tên của radio, kết nối các radio button với nhau
- Value: Những dữ liệu sẽ gửi đến server khi radio button được chọn
- Checked: thuộc tính để radio button được chọn mặc định

**Ví dụ:**

```

<html>
<head><title>Form</title></head>
<body>
  <form><table>
    <tr>
      <td valign=top>User for</td>
      <td>
        <input type='radio' name=use value=home>Home<br>
        <input type='radio' name=use value=bus>Business<br>
        <input type='radio' name=use value=gov>Government<br>
        <input type='radio' name=use value=ed>Educational Institution<br>
        <input type='radio' name=use value=other>Other<br>
      </td>
    </tr>
  </table></form>
</body>
</html>

```



5. **Submit Button:** Tất cả thông tin của người xem nhập vào sẽ được gửi đến server khi người xem click nút Submit

**Cú pháp:**

```
<Input Type="Submit" Value="Submit Message" Name="Name">
```

- Submit Message: Là chữ xuất hiện trên Button
- Name: tên của button

6. **Reset Button:** Thiết lập giá trị ban đầu của tất cả các điều khiển trên form

**Cú pháp:**

`<Input Type="reset" Value="Reset Message" Name="Name">`

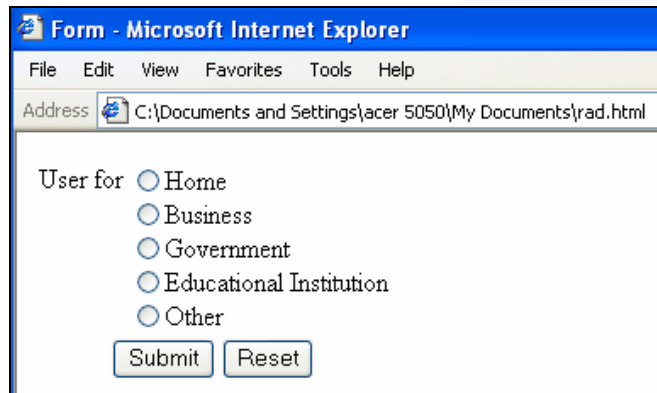
Có thể tạo nút Reset và Submit bằng hình ảnh với cú pháp:

`<Button Type="reset" Name="reset" Value="reset">`

**Nhãn chữ lề trái**

`<Image src="Image.gif">Nhãn chữ lề phải</Button>`

**Ví dụ:**



7. **Button:** Nút dùng để thực hiện các lệnh do người sử dụng đưa ra

**Cú pháp:**

`<input type="button" name="Button" value="Button">`

8. **Hidden:** là các field mà người xem không nhìn thấy trên trình duyệt, nhưng vẫn là một phần tử trên form. Hidden field dùng để lưu trữ thông tin trong các form trước, các thông tin này cần đi kèm với các dữ liệu trong form hiện hành mà không muốn người xem nhập lại

**Cú pháp:**

`<Input Type='hidden' Name='Name' Value='Value'>`

*Name:* tên mô tả ngắn gọn thông tin cần lưu trữ

*Value:* Thông tin cần lưu trữ

### VI.2.2. Selection List:

#### 1. Drop down menu:

**Cú pháp:**

`<Select Name="Name" Size=n Multiple>`

`<Option Value="Value" selected> Option 1`

`<Option Value="Value" > Option 2`

...

`</Select>`

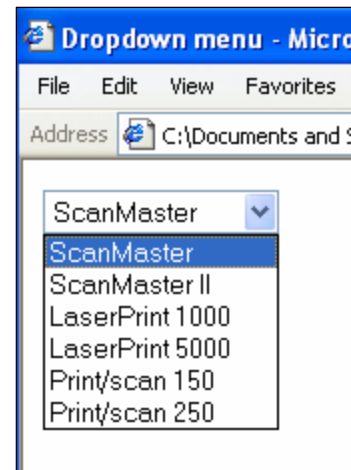
- Nhãn: Giới thiệu Menu
- Name: tên dữ liệu đầu vào server
- Size: là chiều cao của menu tính bằng hàng chữ
- Multiple: là thuộc tính cho phép chọn nhiều đề mục (listbox)
- Selected: đề mục được chọn mặc định
- Value: xác định dữ liệu gửi cho server nếu đề mục được chọn

**Ví dụ:**

```

<html>
  <head><title>Dropdown menu</title></head>
  <body>
    <form>
      <select Name=Product>
        <option value=1>ScanMaster
        <option value=3>ScanMaster II
        <option value=4>LaserPrint 1000
        <option value=5> LaserPrint 5000
        <option value=6>Print/scan 150
        <option value=2> Print/scan 250
      </Select>
    </form>
  </body>
</html>

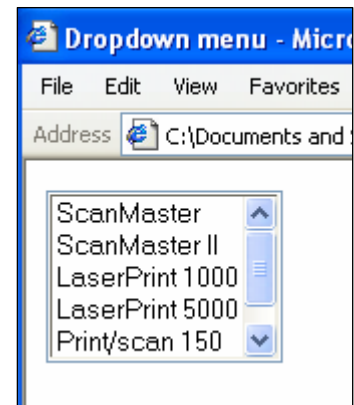
```

**2. Nếu thêm thuộc tính Multiple thì ta được dạng listbox**

```

<html>
  <head><title>Dropdown menu</title></head>
  <body>
    <form>
      <select Name=Product size=5 Multiple>
        <option value=1>ScanMaster
        <option value=3>ScanMaster II
        <option value=4>LaserPrint 1000
        <option value=5> LaserPrint 5000
        <option value=6>Print/scan 150
        <option value=2> Print/scan 250
      </Select>
    </form>
  </body></html>

```

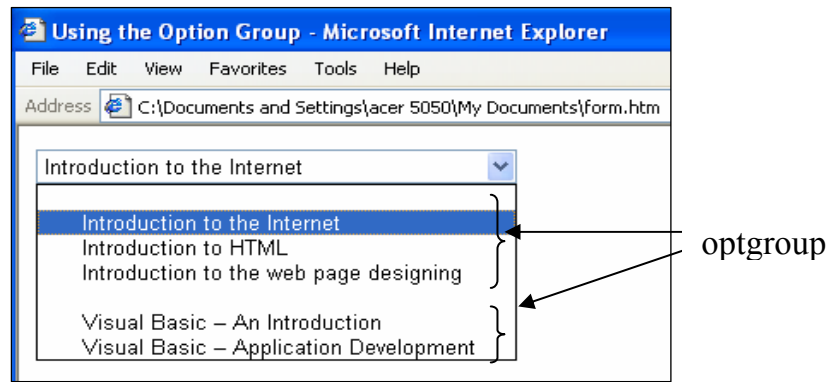
**3. Phần tử OPTGROUP:** được sử dụng để nhóm các chọn lựa thành các nhóm riêng.

Ví dụ:

```

<HTML>
<HEAD><Title>Using the Option Group</Title></head>
<BODY>
  <FORM action= 'http: // somesite.com / processform' method= 'post'>
  <SELECT name= 'course'>
    <OPTGROUP>
      <OPTION value= "Internetintro">Introduction to the Internet
      <OPTION value= "Introhtml">Introduction to HTML
      <OPTION value= "Introweb">Introduction to the web page designing
    </OPTGROUP>
    <OPTGROUP>
      <OPTION value= "vbinintro">Visual Basic – An Introduction
      <OPTION value= "vbdev">Visual Basic – Application Development
    </OPTGROUP>
  </SELECT>
</FORM></BODY></HTML>

```



### VI.2.3. TextArea: Hộp văn bản cho phép nhập nhiều dòng

#### Cú pháp:

`<TextArea Name="name" Rows=n Cols=m Wrap>`

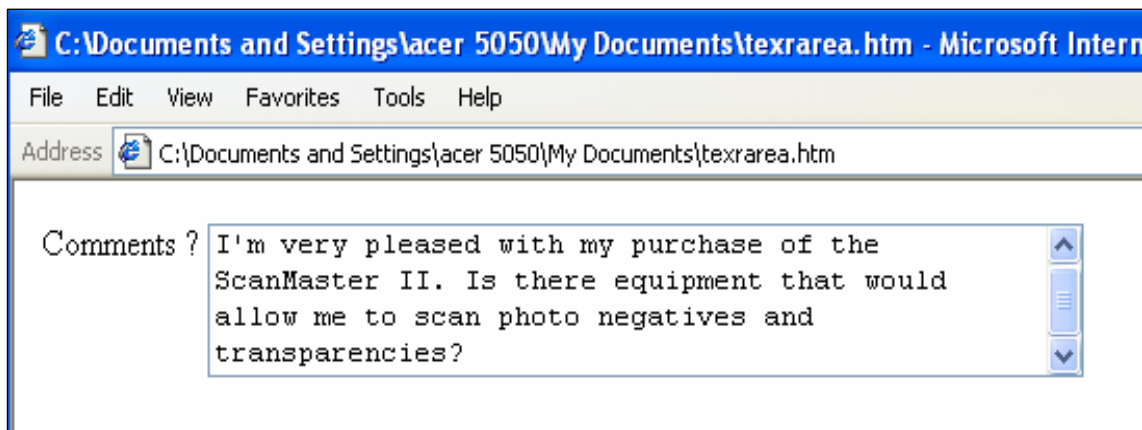
Default text

`</textarea>`

- *Rows*: số dòng có thể nhập vào TextArea (mặc định là 4)
- *Cols*: độ rộng của textarea (tính bằng số ký tự, mặc định là 40)
- *Wrap*: các dòng chữ tự động dàn ra trong lề của vùng text area, Value: virtual,physical

#### Ví dụ:

```
<html>
  <head><title>Textarea</title></head>
  <body>
    <table>
      <tr>
        <td valign=top> Comments ?</td>
        <td><textarea rows=4 cols=50 name=comments wrap=virtual></textarea>
        </td>
      </tr>
    </table>
  </body>
</html>
```



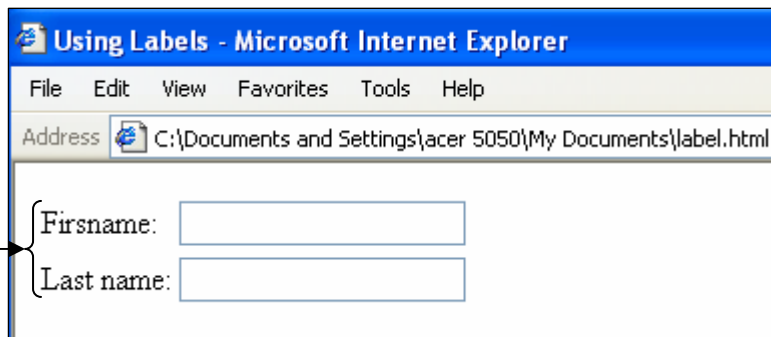
**VI.2.4. Nhãn: Dùng để tạo nhãn liên kết với thành phần đi kèm****Cú pháp:****<Label For="idname"> Nội dung label</label>***Idname*: là giá trị của thuộc tính ID trong thành phần Form tương ứng**Ví dụ:**

```

<HTML>
  <HEAD><TITLE>Using Labels</TITLE></HEAD>
  <BODY>
    <FORM action= 'http: // somesite.com' method = 'post'>
      <table>
        <tr>
          <td><LABEL for= 'firstname'>Firsname: </LABEL></td>
          <td><INPUT type='text' id='firstname'></td>
        </tr>
        <tr>
          <td><LABEL for= 'lastname'> Last name: </LABEL></td>
          <td><INPUT type= 'text' id= 'lastname'><BR></td>
        </tr>
      </table>
    </FORM>
  </BODY>
</HTML>

```

Label

**VI.2.5. Fieldset: Nhóm các đối tượng giống nhau vào một phần logic****Cú pháp:****<Fieldset>****<Legend Align="left, right">Chú thích****</Legend>****Các thành phần trong nhóm****</Fieldset>**

-Tag&lt;legend&gt;: tạo chú thích cho nhóm

-Align=left, right: chỉ vị trí của chú thích

**Ví dụ:**

```

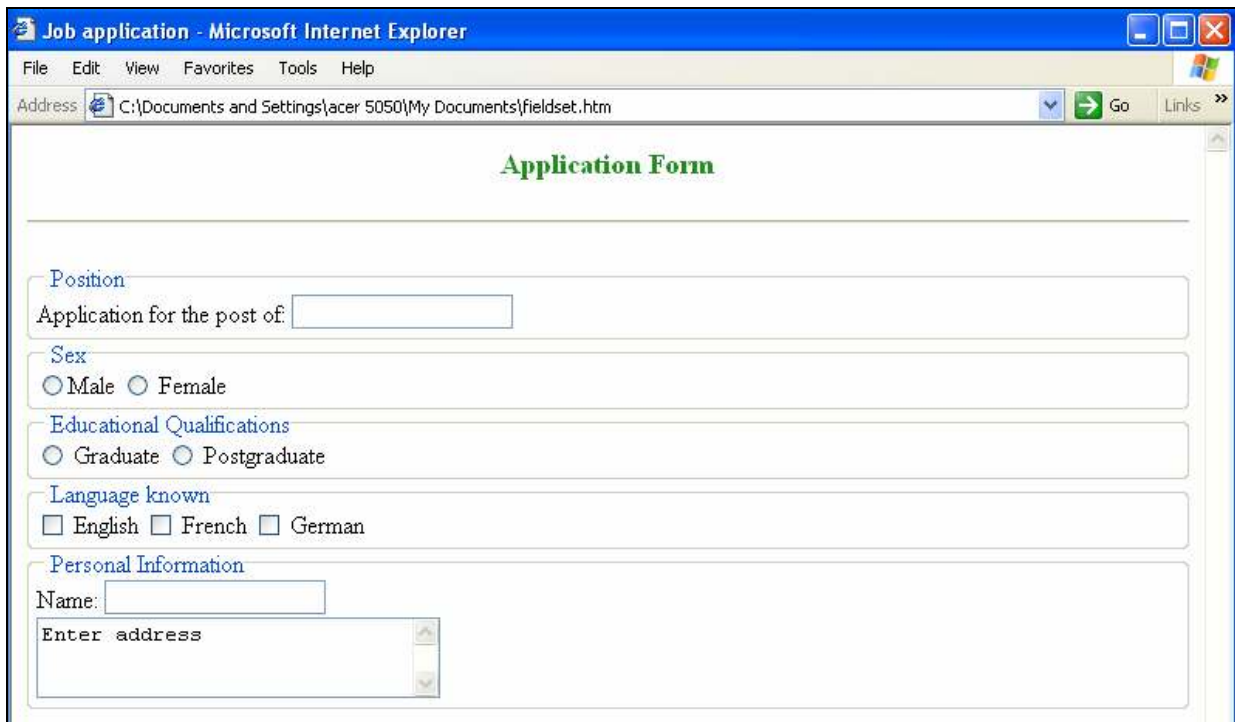
<HTML>
  <HEAD><TITLE>Job application</TITLE></HEAD>
  <BODY >
    <H1><CENTER><FONT SIZE = 4 COLOR= Forestgreen>Application Form
    </CENTER></FONT></H1>
    <HR><BR><FORM action= "http: // somesite.com / processform" method = "post"><P>
    <FIELDSET>
      <LEGEND>Position</LEGEND>
      Application for the post of: <INPUT name= 'name' type= 'text' tabindex= '1'>

```

```

</FIELDSET>
<FIELDSET>
  <LEGEND>Sex</LEGEND>
  <INPUT name= 'sex' type= 'radio' value= 'Male' tabindex= '4' >Male
  <INPUT name= 'sex' type= 'radio' value= 'Female' tabindex= '4'> Female
</FIELDSET>
<FIELDSET>
  <LEGEND>Educational Qualifications</LEGEND>
  <INPUT name= 'qualif' type='radio' value= 'grad' tabindex= '5'> Graduate
  <INPUT name= 'qualif' type='radio' value='postgrad' tabindex='5'> Postgraduate
</FIELDSET>
<FIELDSET>
  <LEGEND>Language known</LEGEND>
  <INPUT name= 'lang' type='checkbox' value= 'english' tabindex= '6'> English
  <INPUT name= 'lang' type='checkbox' value= 'french' tabindex= '7'> French
  <INPUT name= 'lang' type='checkbox' value= 'german' tabindex= '8'> German
</FIELDSET>
<FIELDSET>
  <LEGEND> Personal Information</LEGEND>
  Name: <INPUT name = 'name' type= 'text' tabindex= '2'><BR>
  <TEXTAREA name = 'address' rows= '3' cols = '30' tabindex = "3">
    Enter address</TEXTAREA>
</FIELDSET>
</FORM>
</BODY></HTML>

```



## VI.2.6. Điều khiển các phần tử trên form:

### 1. Định thứ tự Tab:

Dùng phím tab để di chuyển giữa các đối tượng trong form, mặc định theo thứ tự của mã HTML, muốn định lại thứ tự ta dùng thuộc tính **TabIndex=n** trong tag tạo các thành phần của form, trong đó n là thứ tự của tab, có giá trị từ 0 đến 32767

Trong một form ta thường định thứ tự tab cho các thành phần : textbox, password, checkbox, radio button, textarea, menu và button

### 2. Tạo phím tắt:

– Cách tạo:

Trong tag tạo các phần tử của form ta dùng thuộc tính **Accesskey="Phím tắt"**

– Sử dụng phím tắt: Nhấn tổ hợp phím **Alt+Phím tắt**

## CHƯƠNG VII: CASCADING STYLE SHEET-CSS

### VII.1. GIỚI THIỆU

- Bảng kiểu (style sheet) nhằm thỏa mãn nhu cầu thẩm mỹ, tiện dụng nhưng giữ tính thống nhất cho trang HTML. CSS cho phép định dạng một số tính chất thông thường cùng một lúc cho tất cả các đối tượng trên trang được đánh dấu bằng tag đặc biệt
- Tiện ích của CSS là :
  - Tiết kiệm thời gian
  - Khi thay đổi định dạng chỉ cần thay đổi CSS, các trang khác sẽ tự động cập nhật sự thay đổi đó
  - Có thể dùng các CSS cùng với JavaScript để tạo các hiệu ứng đặc biệt
- Bất lợi của CSS:
  - Không một trình duyệt nào chấp nhận nó hoàn toàn
  - Phải mất thời gian để học cách sử dụng

### VII.2. CÁCH TẠO:

Một bảng mẫu được tạo bằng một tên tag và một hay nhiều các định nghĩa để xác định cách thức hiển thị của các đối tượng được đánh dấu bằng tag đó

#### VII.2.1. Phân loại và cách tạo:

Có 3 loại :

- Inline style
- Internal style
- External style

#### a. Inline style:

Là kiểu được gán cho một dòng hoặc một đoạn văn bản, bằng cách sử dụng thuộc tính style bên trong tag muốn định dạng

**Cú pháp:**

```
<TagName Style="property1:value1;property2: value2;...">
```

**Nội dung văn bản muốn định dạng**

```
</TagName>
```

Ví dụ : <HTML>

```
<HEAD>
```

```
<TITLE>Setting Properties</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<P style = "color:aqua ; font- Style:italic, text- Align:center">
```

```
    This paragraph has an inline style applied to it
```

```
<P> This paragraph is displayed in the default style.
```

```
<P> Can you see the <SPAN style = color:red>difference
```

```
</SPAN> inthis line
```

```
</BODY>
```

```
</HTML>
```

#### b. Internal style :

Là bảng mẫu thích hợp cho trang riêng lẻ với nhiều văn bản, bằng cách tạo bảng mẫu chung trên đầu trang và dùng cho cả trang HTML



**Cú pháp:****<Head>****<Style>****TagName{property1: value 1; property2: value 2...}***(lặp lại cho mỗi tag có thuộc tính cần định dạng)***</Style>****</Head>****Ví dụ :****<HTML>****<HEAD>****<STYLE TYPE="text/css">****H1,H2 { color: limegreen; font-family: Arial }****</STYLE>****</HEAD>****<BODY>****<H1>This is the H1 element</H1>****<H2>This is the H2 element</H2>****<H3>This is the H3 element with its default style as displayed in the browser</H3>****</BODY>****</HTML>****c. External style :**

Là một bảng kiểu được lưu trữ thành một file bên ngoài và được liên kết với trang HTML. Bảng kiểu này sẽ được áp dụng và ảnh hưởng cho tất cả các trang của một website.

**- Cách tạo:**

- Tạo một tập tin văn bản mới
- Nhập tên các tag muốn định dạng thuộc tính theo mẫu:

**TagName{property1: value1; property2:value2;...}**

- Lưu tập tin với định dạng Text Only và có phần mở rộng .css

**- Cách dùng External style:****Cú pháp:****<Head>****<Link Rel=StyleSheet Type="text/css" Href="tên tập tin.css">****</Head>****Ví dụ:**Tạo tập tin **Sheet1.css**

H2 {color:blue; font-style:italic}

P {text-align:justify; text-indent:8pt; font:10pt/15pt "Myriad Roman","Verdana"}

**Trang1.htm**

```

<HTML>
  <HEAD><TITLE> Changing the rules</TITLE>
    <LINK REL=stylesheet HREF="sheet1.css" TYPE="text/css">
  </HEAD>
  <BODY>
    <H2> Changing the rules is fun</H2>
    <P> Changing the rules may not be such fun
    <H2>The H2 element again</H2>
  </BODY>
</HTML>

```

**Trang2.htm**

```

<HTML>
  <HEAD><TITLE> Changing the rules</TITLE>
    <LINK REL=stylesheet HREF="sheet1.css" TYPE="text/css">
  </HEAD>
  <BODY>
    <H2> This document ues the sheet1 style sheet</H2>
    <P>Selecting this option could delete all your files
    <H2>The H2 element again</H2>
  </BODY>
</HTML>

```

**VII.3. ĐỊNH BẢNG MẪU CHO LỚP (CLASS):**

Có thể chia các yếu tố trong HTML thành các lớp để áp dụng kiểu mẫu hiệu quả hơn

**Cú pháp:**

- Trong phần <Style > nhập cú pháp:

```

<STYLE>
  .ClassName{thuộc tính1:giá trị1;thuộc tính2:giá trị2;...}
</STYLE>

```

- Trong phần <Body>, đánh dấu phần nằm trong lớp bằng cú pháp:

```

<Body>
  <TagName Class="ClassName">Nội dung </TagName>
</Body>

```

**Ví dụ:** <HTML>

```

  <HEAD>
    <STYLE>
      .water{color:blue}
      .danger{color:red}
    </STYLE>
  </HEAD>
  <BODY>
    <p class=water>test water
    <P class=danger>test danger
  </BODY></HTML>

```

### VII.3.1. Định các tag riêng biệt:

Dùng áp dụng cho một phần tử riêng biệt trên trang Web

#### Cú pháp:

- Trong Tag Style nhập :

**TagName#IDName{th/tính1: giá trị1; thuộc tính2: giá trị 2;...}**

- Trong tag Body nhập :

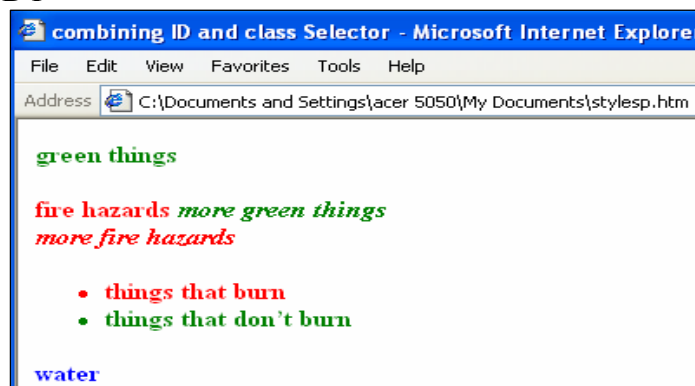
**<TagName ID=IDName> Nội dung</TagName>**

#### Ví dụ 1:

```
<HTML>
  <HEAD><TITLE> ID Selectors</TITLE>
    <STYLE>
      #control { color: red ;FONT-WEIGHT:BOLD}
    </STYLE>
  </HEAD>
  <BODY>
    <SPAN id='control'>Fire is this color</SPAN>This paragraph has
    no style applied
  </BODY></HTML>
```

#### Ví dụ 2:

```
<HTML>
  <HEAD><TITLE> combining ID and class Selector</TITLE>
  <STYLE>
    .forest { color: green;font-weight:bold }
    .danger { color: red;font-weight:bold }
    #control { color: blue;font-weight:bold }
  </STYLE>
  </HEAD>
  <BODY>
    <P class='forest'>green things
    <P class='danger'>fire hazards
    <EM class='forest'> more green things</EM><BR>
    <EM class='danger'>more fire hazards</EM>
    <UL>
      <LI class='danger'>things that burn
      <LI class='forest'>things that don't burn
    </UL>
    <P id='control'> water </P>
  </BODY>
</HTML>
```



### VII.3.2. Tạo các tag tùy ý:

Có 2 loại tag chung có thể kết nối Class hay các ID để tạo các tag tùy ý. cần phân biệt đối tượng cấp khối và cấp hàng:

- Đối tượng cấp khối như một đoạn văn, thường bắt đầu một dòng mới và có thể chứa các đối tượng cấp khối khác gồm các tag: P, H1, Body, table
- Đối tượng cấp hàng thường không tạo dòng mới, thường chứa văn bản và các yếu tố trong hàng khác gồm các tag: B, Font
- Các tag DIV và SPAN: có thể kết nối với các phần tử cấp khối và ID để tạo ra các tag tùy ý. Trong đó DIV phù hợp với các đối tượng cấp khối, SPAN phù hợp với các đối tượng cấp hàng

#### 1. Tạo tag cấp khối tùy ý:

Cú pháp: Bằng cách thêm một lớp hoặc ID vào tag DIV và định mẫu cần có

- Trong phần Style hoặc một bảng mẫu bên ngoài ta nhập:  
**DIV.ClassName{th/tính1: giá trị 1; thuộc tính 2: giá trị 2...}**  
 với ClassName là tên lớp sẽ sử dụng. hoặc:  
**DIV#IDname{thuộc tính1: giá trị 1; thuộc tính 2: giá trị 2...}**  
 với IDname là tên cá biệt của tag DIV
- Áp dụng tag cấp khối tùy ý vào trang HTML: Tại đầu phần văn bản muốn định dạng, nhập cú pháp  
**<DIV Class="ClassName" IDname="IDname">Nội dung </DIV>**  
 (bên trong có thể chứa các tag <P> hoặc <H1>)

#### 2. Tạo các tag trong hàng tùy ý:

Kết nối nhiều kiểu định dạng văn bản trong một tag

Cú pháp:

- Trong phần Style, nhập cú pháp:  
**SPAN.Classname {th/tính1: giá trị 1; th/tính2: giá trị 2...}**  
 Hoặc:  
**SPAN#IDname {th/tính1: giá trị 1; th/tính 2: giá trị 2...}**
- Áp dụng tag trong hàng tùy ý vào trang HTML: Tại đầu đoạn văn bản muốn định dạng, nhập cú pháp:  
**<SPAN Class="classname"> nội dung văn bản</SPAN>**  
 Hoặc:  
**<SPAN ID="IDName"> Nội dung văn bản</SPAN>**

#### 3. Các thuộc tính định dạng văn bản:

- a) Chọn bộ font:  
**font-family: familyname1, familyname2...**
- b) Tạo chữ nghiêng:  
**Font-style: italic**
- c) Tạo chữ đậm:  
**Font-weight: bold**
- d) Định cỡ chữ:

**Font-size: xx-small hoặc x-small, small, medium, large, x-large, xx-large hoặc Font-size: 12pt (giá trị cụ thể)**

Có thể định dạng các thuộc tính chữ nghiêng, đậm và cỡ chữ cùng một lúc:

**Font: italic bold size**

e) Màu của chữ:

**Color: colorName/#rrggbb**

f) Màu nền của chữ:

**Background: colorName/#rrggbb**

g) Định khoảng các giữa các từ, các ký tự:

**Word-spacing: n** (n: khoảng cách giữa các từ, tính bằng pixel)

**Letter-spacing: n** (n: khoảng cách giữa các từ, tính bằng pixel)

h) Canh lề cho văn bản:

**Text-Align: left, right, center, justify**

i) Thay đổi dạng chữ:

**Text-transform: capitalize, uppercase, lowercase**

#### 4. Định dạng danh sách:

List-style: circle chấm tròn rỗng

List-style: disc chấm tròn đen

List-style: square chấm đen vuông

List-style: decimal đánh số Ả Rập

List-style: lower-alpha thứ tự alpha

List-style: upper-alpha thứ tự alpha chữ in hoa

List-style: upper-roman số la mã hoa

List-style: lower-roman số la mã thường

#### 5. Định dạng màu nền:

Body {color: #rrggbb}

blockquote {background-color: #rrggbb}

background: background-color

background: background-image

background: background-position

background: background-repeat

background: background-attachment

#### 6. Định dạng Hypertext link

A {Text-Decoration: none}: không gạch dưới

A:visited {color: #rrggbb}

A:link {styles cho vị trí chưa được xem}

A:active {style cho những link đang click}

A:hover {style khi trỏ lướt qua link}

## CHƯƠNG VIII: GIỚI THIỆU DREAMWEAVER

### VIII.1. GIỚI THIỆU

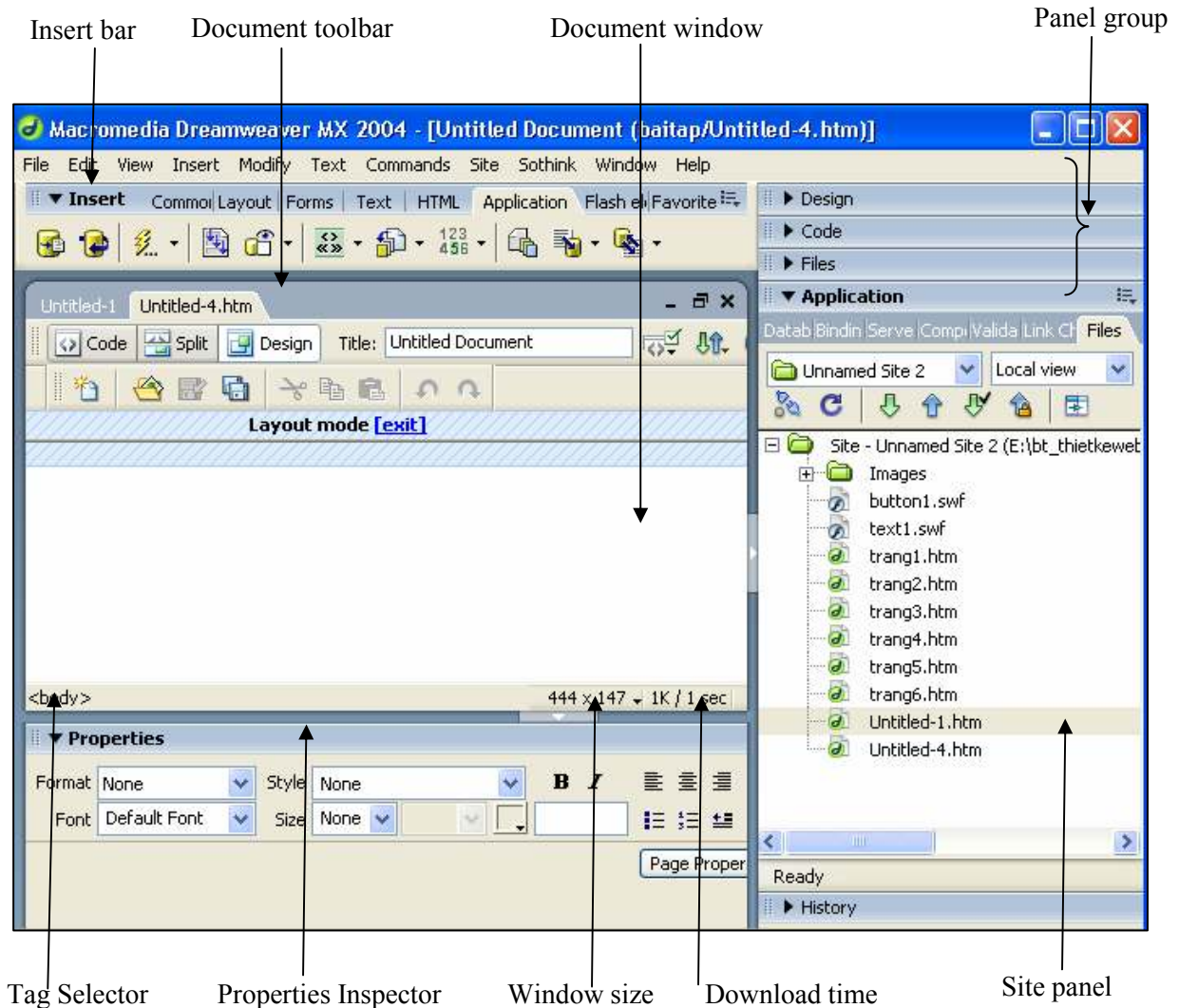
- Macromedia Dreamweaver MX 2004 là một công cụ thiết kế web chuyên nghiệp, phần cốt lõi của nó là HTML (HyperText Markup Language), là một công cụ mạnh, dễ dùng, bạn có thể dễ dàng thiết kế và phát triển một trang web hoặc một website lớn
- Dreamweaver MX 2004 là một công cụ trực quan, trong đó có thể bổ sung Javascript, biểu mẫu, bảng biểu và nhiều loại đối tượng khác mà không cần viết một đoạn mã nào.
- Dreamweaver MX 2004 sử dụng các công nghệ web, chuẩn HTML và cung cấp khả năng tương thích với các trình duyệt web cũ, với Dreamweaver MX 2004 bạn có thể thiết kế bằng chế độ Design view hoặc Code view hoặc Code and Design

### VIII.2. CÀI ĐẶT

- Macromedia Dreamweaver MX 2004 là một chương trình trong bộ Macromedia MX, bạn nên cài đặt trên máy trọn bộ Macromedia MX để có đầy đủ các chương trình hỗ trợ cho Dreamweaver thiết kế trang web đẹp và sinh động hơn
- Sau khi cài đặt, bạn khởi động Macromedia Dreamweaver MX 2004 theo đường dẫn Start → Programs → Macromedia → Macromedia Dreamweaver MX 2004

### VIII.3. MÀN HÌNH DREAMWEAVER:

- 1. Insert Bar:** Gồm các chức năng tiện ích dùng để chèn các đối tượng vào trang web, và định các thuộc tính cho đối tượng
  - Common: Chèn các đối tượng: Image, Flash, Date, Template, ...
  - Layout: Chứa các công cụ trình bày trang, gồm 3 chế độ: Standard, Expanded, Layout
  - Forms: Chứa các công cụ tạo Form
  - Text: Dùng định dạng văn bản
  - HTML: chứa các công cụ tạo trang web bằng code view
- 2. Document Toolbar:** Chứa các nút cho phép xem trang web ở dạng Design hay dạng Code
  - Show code view: Xem dạng trang HTML
  - Show Design view: Xem trang dạng thiết kế, sử dụng các công cụ của Dreamweaver
  - Show code and design view: Chia cửa sổ làm 2 phần: phần trên dạng code view, phần dưới dạng Design view
  - Title: tiêu đề của trang Web
  - Preview/Debug in Browser: Xem kết quả trang web thông qua trình duyệt web
  - Document Window: Cửa sổ dùng để tạo và hiệu chỉnh trang Web
- 3. Properties Inspector:** Hiện thị các thuộc tính của các đối tượng đang được chọn, đồng thời cho phép chỉnh sửa các thuộc tính đó
- 4. Panel groups:** Là nhóm các Panel cho phép quản lý các đối tượng trong trang Web
  - Bật / tắt các thanh Panel: Chọn menu Window → Chọn thanh Panel tương ứng
  - Mở rộng các thanh Panel: Click vào mũi tên ở góc trái của mỗi Panel



## 5. Site Panel: Cho phép quản lý các tập tin, thư mục trong bộ Web (giống Windows Explorer).

Thanh công cụ của Site Panel



**Connect/Disconnect:** Chức năng kết nối/ ngắt kết nối với Remote site, chỉ có tác dụng với Remote site sử dụng phương thức truyền FTP, WebDAV hoặc Sourcesafe, mặc định Dreamweaver MX 2004 sẽ ngắt kết nối khỏi remote site nếu nó ở trạng thái chờ 30 phút. Có thể thay đổi thời gian này bằng cách chọn: Edit/ Preferences/ Site



**Refresh:** Chức năng cập nhật tập tin, thư mục cho Remote site giống với Local Site của chính nó. Thường không sử dụng chức năng này vì khi tạo Site mới Dreamweaver MX 2004 luôn đánh dấu check vào mục Refresh Remote File list Automatically



**Get File:** chức năng chép File từ remote Site vào Local Site. Tùy thuộc vào Enable File Check in và Check out mà các tập tin chép vào có thuộc tính được phép ghi hay chỉ đọc



**Put File:** Chép tập tin từ Local Site lên Remote Site



**Check out files:** Kiểm tra tập tin ở Remote Site chép vào hay chép chồng lên tập tin ở Local Site



Check in files: Kiểm tra tập tin ở Local Site chép vào hay chép chông lên tập tin ở Remote Site



Expand/Collapse: hiển thị 2 cửa sổ Local Site và Remote Site

- 6. Status bar:** Thanh trạng thái, nằm dưới đáy của Document Window, hiển thị Tag Selector, Window size, Document size và Download time.
- Tag Selector: Hiển thị các tag HTML tại vị trí hiện hành của con trỏ, khi tạo trang Web mới thì phần tử duy nhất hiển thị trong Tag Selector là BODY.
  - Document size and Download time: Kích cỡ ước chừng của tài liệu và thời gian tải tài liệu xuống, có thể điều chỉnh tốc độ download bằng cách:
    - Chọn Edit → Preferences → Chọn mục Status bar
    - Tại mục Connection speed → Chọn tốc độ tương ứng
  - Window size: Hiển thị kích thước hiện tại của tài liệu, được tính bằng Pixel. Khi định kích thước của trang web phải tính đến việc sao cho an toàn đối với mọi độ phân giải. Cách thay đổi kích thước của tài liệu theo một trong các kích thước định sẵn hoặc theo một giá trị khác: Click chuột vào mũi tên bên cạnh Window size
    - Chọn một kích thước có sẵn, hoặc
    - Chọn Edit size để định một kích thước khác
    - Trong hộp thoại Preferences
    - Width: định chiều rộng
    - Height: định chiều cao

## VIII.4. KẾ HOẠCH THIẾT KẾ MỘT WEBSITE

### VIII.4.1. Các yêu cầu cơ bản khi thiết kế website:

- Xác định yêu cầu và mục đích của Website
- Chuẩn bị nội dung cho các trang
- Phác thảo khuôn mẫu (Template) cho trang, thường các trang có cùng chủ đề thì sử dụng chung một template
- Xác định cấu trúc của Website, có 3 kiểu cấu trúc:
  - Tuyến tính
  - Phân cấp
  - Hình chóp
- Tùy theo mục đích của Website mà chọn kiểu phù hợp

### VIII.4.2. Thao tác tạo bộ Web cơ bản:

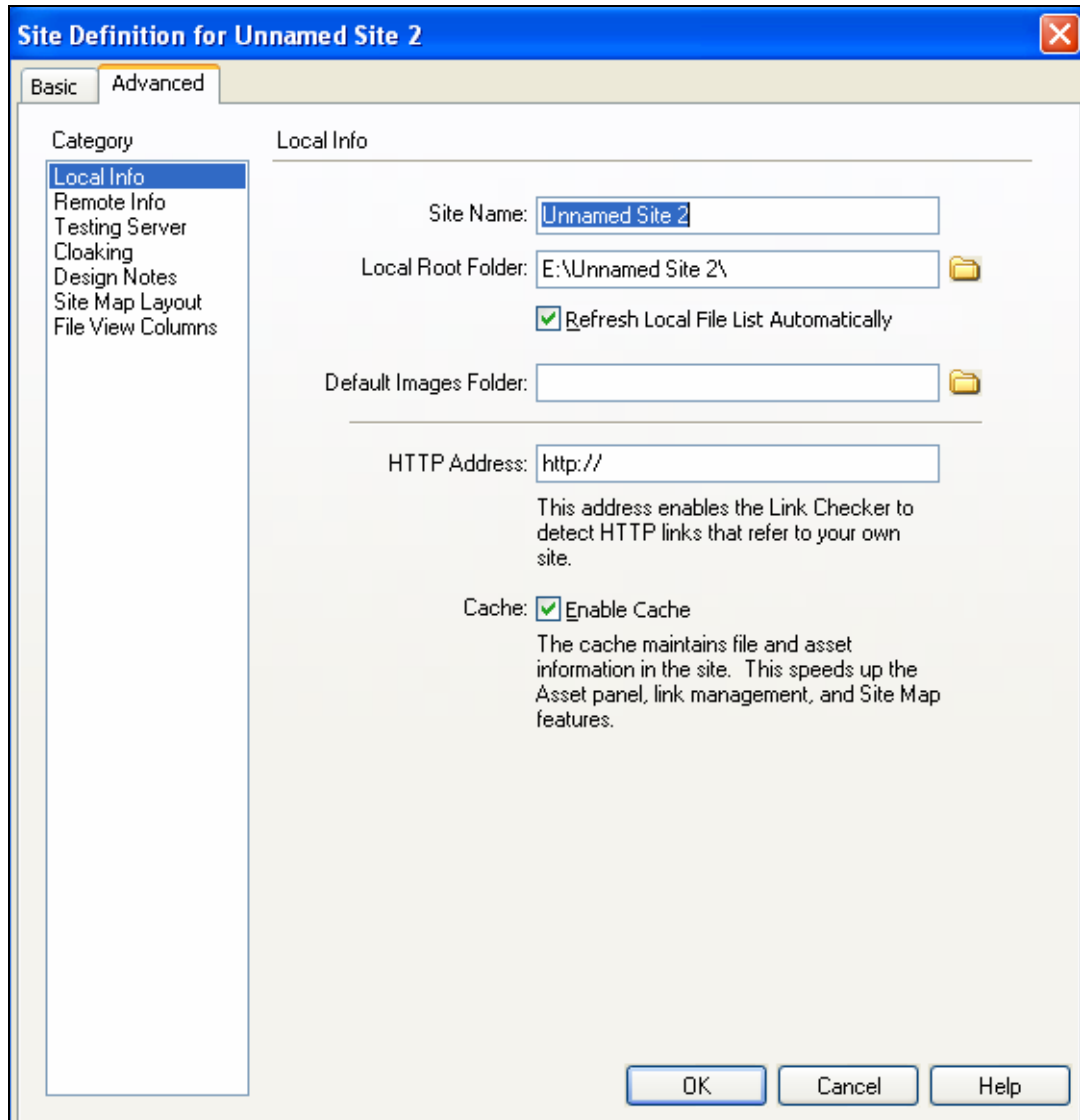
#### 1. Khi thiết kế một Website cần quan tâm đến 2 vấn đề:

- Nội dung chủ đề chính, từ đó chọn bố cục, hệ màu cho tương ứng, (ví dụ: Website thương mại phải sáng sủa, rõ ràng về bố cục, ...) sau đó thu thập đầy đủ tài liệu, phân nhóm theo nội dung, từ đó quyết định cần bao nhiêu trang, nội dung của từng trang
- Chọn hình ảnh, logo, Banner, hệ thống nút liên kết, ảnh minh họa, ảnh bố cục, ảnh trang trí...
- Phác họa sơ đồ liên kết trên giấy để thấy rõ mối liên kết giữa các trang đơn trong một website

#### 2. Cách tạo một Website mới:

- Trong Document Window, chọn Site → Manage sites → New → Site → xuất hiện hộp thoại Site Definition → Chọn Tab Advance, trong mục Local info:

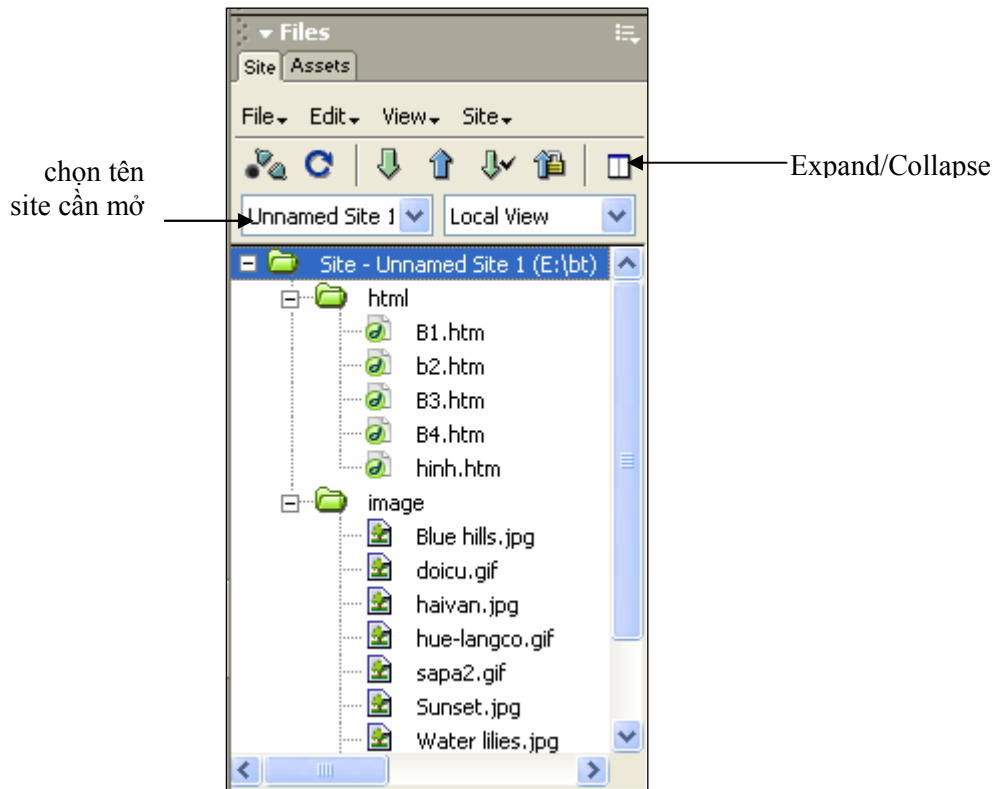




- *Site name*: đặt tên Site, tên này xuất hiện trong hộp thoại Edit Site
- *Local Root Folder*: Khai báo đường dẫn của folder lưu trữ Website trên ổ đĩa cứng bằng cách nhập đường dẫn đến thư mục hoặc Click vào biểu tượng Folder và chỉ đường dẫn đến folder
- *Default Images folder*: khai báo đường dẫn đến thư mục chứa các hình ảnh của trang Web, thư mục này phải nằm trong Local root Folder đã khai báo ở trên, tất cả các hình ảnh trong trang web mặc định được lưu trong thư mục này
- *Refresh Local file list Automatically*: khi được chọn, Dreamweaver tự động cập nhật cấu trúc file trong bảng Local Folder của Site Panel, việc cập nhật này sẽ sử dụng một ít tài nguyên của hệ thống, ta có thể cập nhật khi cần bằng cách chọn View → Refresh Local trong Site Window
- *HTTP Address*: Nhập địa chỉ của site, Dreamweaver sẽ sử dụng địa chỉ này để quản lý site và liên kết các file trong site
- *Enable Cache*: khi được chọn, Dreamweaver tạo một file lưu trữ các thông tin về link giữa các file trong site. Sau khi chọn xong Click OK → Click Done để hoàn tất công việc tạo site mới

### 3. Kiểm tra Website đã tạo:

- Mở site Panel :bằng cách chọn Window → File ta thấy được thư mục chứa Site, nếu chưa tạo Folder chứa hình ảnh thì tại đây bạn Click phải chuột vào tên Site → Chọn New Folder→ Nhập tên thư mục là Images. Trong website phải chứa 2 folder : folder HTML gồm các file .htm và folder image chứa hình ảnh của website
- Để mở rộng cửa sổ Site Panel → Click chọn nút Expand/Collapse



#### 4. Mở một site có sẵn:

- Cách 1: Click vào menu hiển thị tên Site trên Toolbar của Site Panel, chọn tên Site muốn mở trong danh sách xổ xuống
- Cách 2: Chọn menu Site→ Manage Sites → Chọn tên Site muốn mở → Done

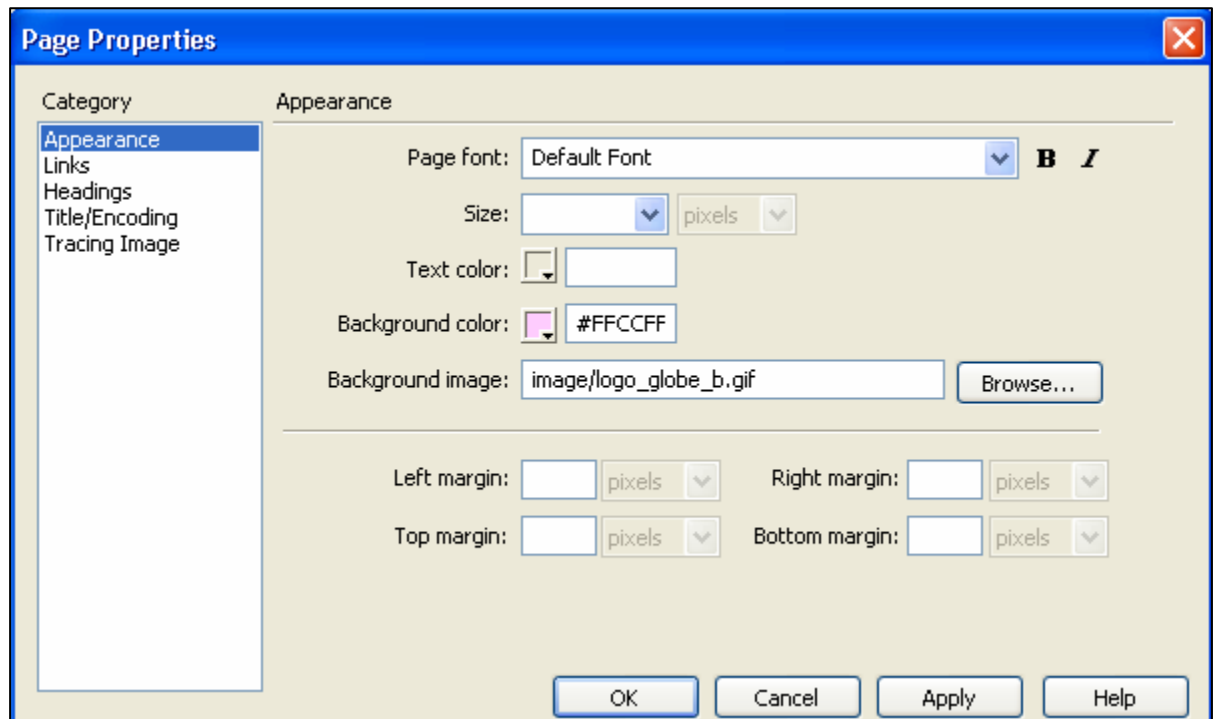
#### 5. Hiệu chỉnh Site:

- Chọn menu Site→ Manage Sites
- Chọn tên Site cần hiệu chỉnh → Click nút Edit
- Xuất hiện hộp thoại Site Definition→ thực hiện hiệu chỉnh → OK → Done



## 6. Dùng hình Bitmap làm ảnh nền cho trang Web:

- Trong cửa sổ Document, chọn Modify → Page Properties, xuất hiện hộp thoại Page Properties, chọn mục Appearance
- Text Color: chọn màu cho text
- Background Color: Chọn màu nền cho trang
- Background Image: Chọn tập tin ảnh làm nền bằng cách click nút Browse...



## VIII.4.3. Thiết kế các trang Web đơn



- Tại màn hình khởi động chọn mục Create new → Chọn HTML

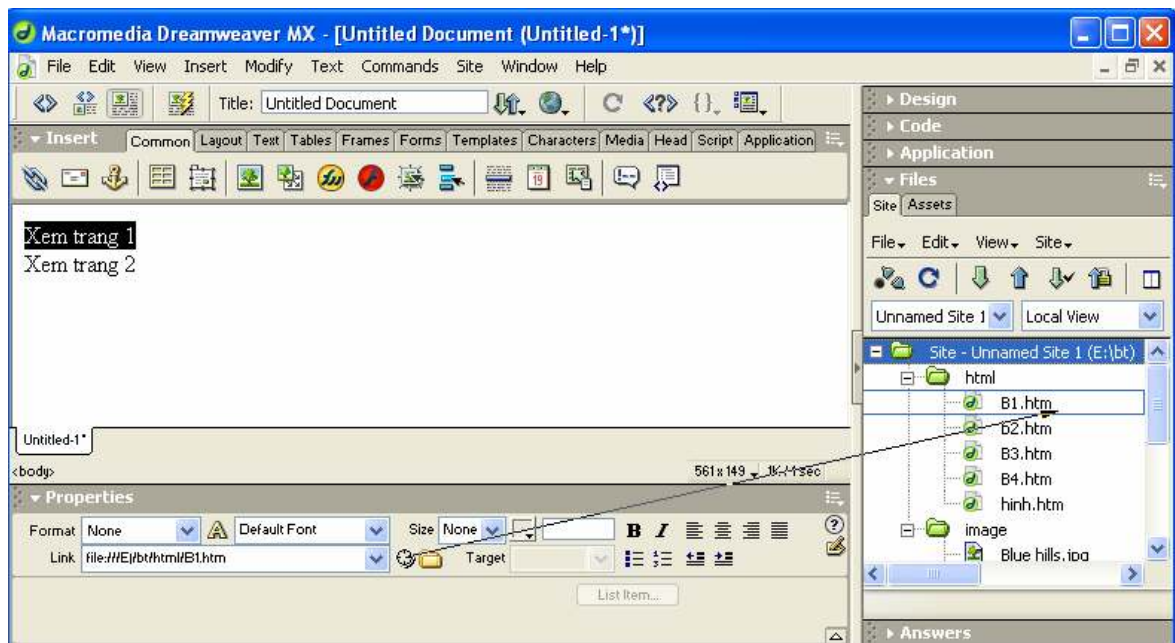


- Xuất hiện Document Window, đây là nơi thiết kế trình bày nội dung của từng trang web đơn, sau khi thiết kế xong mỗi trang web được lưu dưới dạng một tập tin có phần mở rộng .HTM (hoặc .HTML) trong thư mục đã được khai báo trong mục Local Root Folder

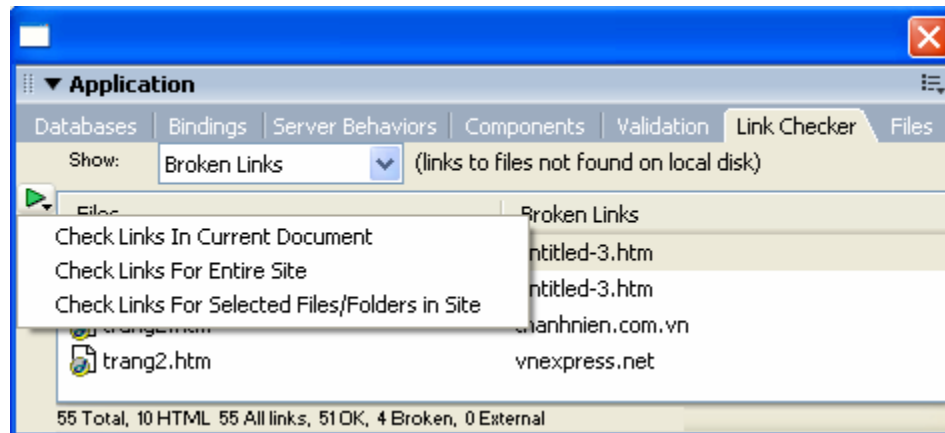
#### VIII.4.4. Tạo liên kết các trang Web đơn thành một Website

##### 1. Cách tạo:

- Để tạo liên kết, cần phân biệt trang nguồn và trang đích.
  - Trang nguồn là trang chứa các nút liên kết (có thể là đối tượng ảnh hoặc chữ)
  - Trang đích là trang cần liên kết đến
- Mở trang nguồn
- Chọn nút liên kết
- Trong Properties Inspector, tại mục link, thực hiện một trong hai cách sau:
  - *Cách 1:* Click nút  kéo mũi tên chỉ đến tên tập tin cần liên kết trong Site Panel
  - *Cách 2:* Click nút  mở hộp thoại Select File
    - Look in: Chọn tên Site
    - File name: Chọn tên trang Web cần liên kết đến



- 
- Check links for entire Site: kiểm tra liên kết cho tất cả các trang trong site
- Check links for Selected files /folders in Site: kiểm tra nhóm tập tin/ thư mục được chọn trong Site



### 3. Xem kết quả bằng trình duyệt và hiệu chỉnh

- Chọn File / Preview in Browser / iexplore
- Hoặc Click nút Preview /Debug in Browser

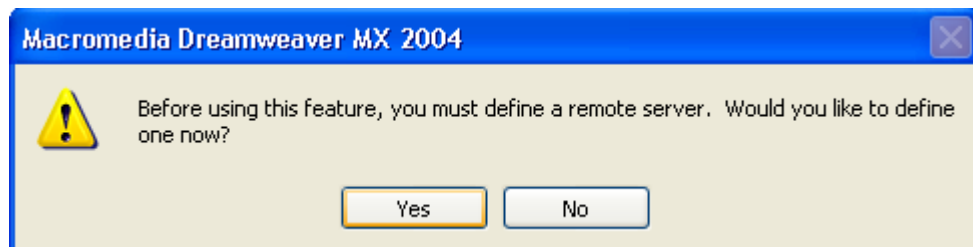
### 4. Kết nối và đưa Website lên Web Server

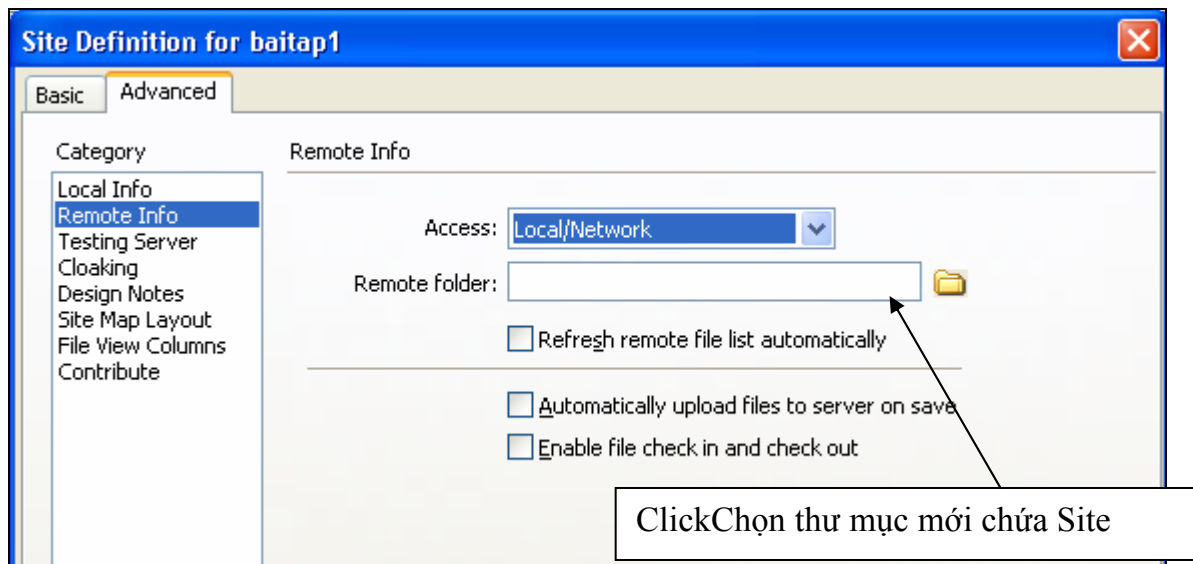
Cần phải lưu lại tất cả các tập tin trước khi xuất bản Website. Xuất bản Website là chép thư mục gốc (root) của Site lên Server của các nhà cung cấp dịch vụ Internet (ISP).

Trong Macromedia Dreamweaver MX 2004, xuất bản Website cần có bước kết nối với Server trước rồi mới Put File lên sau

❖ Cách thực hiện:

- Trong Site Panel, chọn Site cần xuất bản
- Click nút Put Files, để đưa Site lên Server
- Kết nối với Remote Site: Nếu khi tạo Site mới ta chưa xác định Remote Site (Thư mục chứa Site trên Server), nên sau khi click Put File sẽ xuất hiện thông báo yêu cầu kết nối với Remote Site





- Chọn Yes, Xuất hiện hộp thoại Site Definition
- Chọn mục Remote Info, trong khung Access, chọn Local/ Network (giả lập một thư mục trên mạng cục bộ, hoặc trên một thư mục khác của ổ đĩa cứng)
- Tại mục Remote Folder, Click biểu tượng Folder, để tìm thư mục mới chứa Site

#### 5. Xuất bản Site lên Remote Site:

- Trong Site Panel, chọn lại tên Site cần xuất bản
- Click nút Put File
- Xuất hiện hộp thoại: Are you sure you wish to put the entire site?
- Click OK
- Xuất hiện hộp thoại kết nối, các tập tin và thư mục của Site lần lượt được chép từ site lên Remote Site

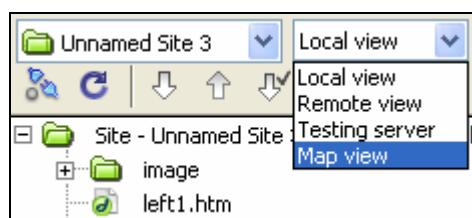
#### 6. Kiểm tra lại trên Remote Site

- Click nút Expand trong Site Panel: Màn hình chia làm 2 phần: Bên trái là Remote Site, Bên phải là Local Site

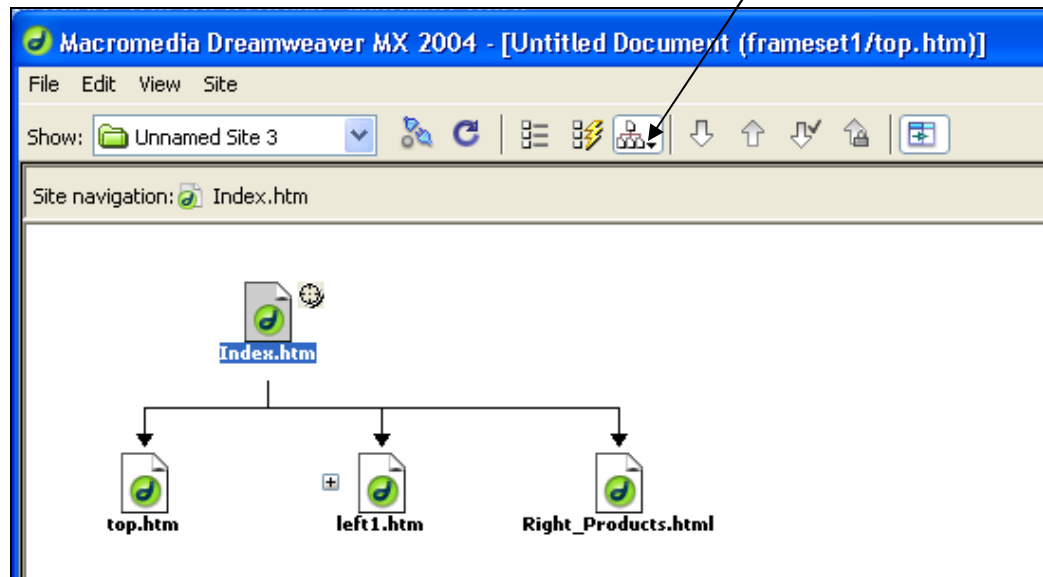
### VIII.4.5. Site Map:

#### 1. Xem một Site Map:

- Site map là một sơ đồ cấu trúc Site, nó hiển thị vị trí và sự phân cấp của các tập tin trong Site. Một Site khi được tạo đầy đủ liên kết, có thể xem dưới dạng Site map
- Cần phải định nghĩa trang HomePage trước hoặc trong Site phải có trang Index.htm
- Trong Site Panel, chọn Map view trong khung Site view

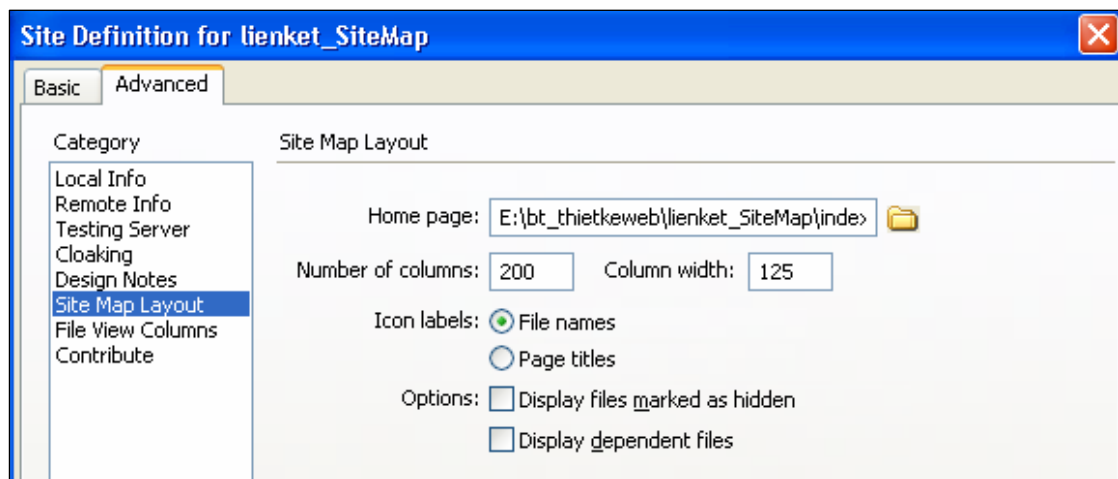


- Hoặc click nút Expand/Collapse → Click chọn Site Map



## 2. Tạo liên kết trong Site Map: Có thể tạo liên kết trang một cách trực quan và đơn giản bằng cách sử dụng Site Map. Cách thực hiện như sau:

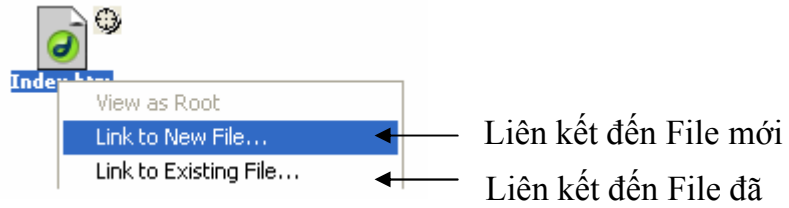
- Tạo Site mới trong đó phải có trang Index.htm hoặc Home Page
  - Chọn Site → Manage Sites → Click nút Edit
  - Xuất hiện cửa sổ Definition → Chọn Site Map Layout
  - Home Page: đường dẫn đến tập tin Index → OK → Done



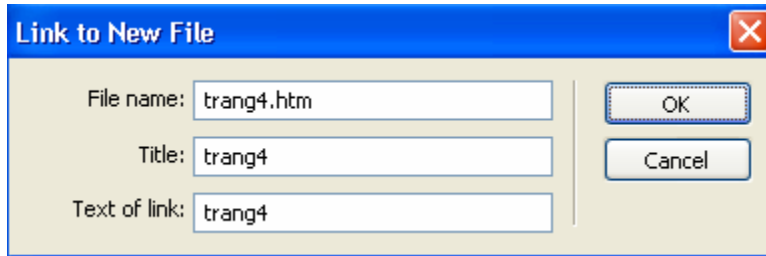
- Tạo liên kết bằng SiteMap
  - Click nút Expand/Collapse để mở rộng Site Panel
  - Click chọn nút SiteMap
  - Màn hình xuất hiện trang Index.htm trong site



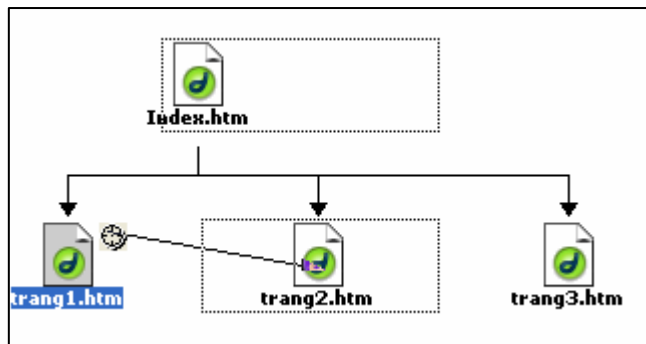
- Tạo liên kết phân cấp:
  - Click phải trên file Index → chọn Link to new File



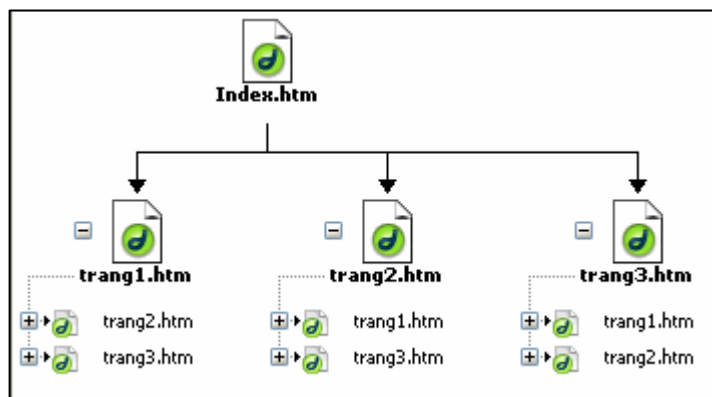
- Xuất hiện hộp thoại Link to New File:
- File Name: Nhập tên file
- Title: tiêu đề của trang
- Text of Link: dòng text để liên kết



- Tạo liên kết nhanh:
  - Chọn tập tin cần tạo liên kết
  - Click biểu tượng liên kết bên cạnh tập tin được chọn
  - Kéo mũi tên liên kết đến tập tin liên kết đến



- Sau khi liên kết, được SiteMap



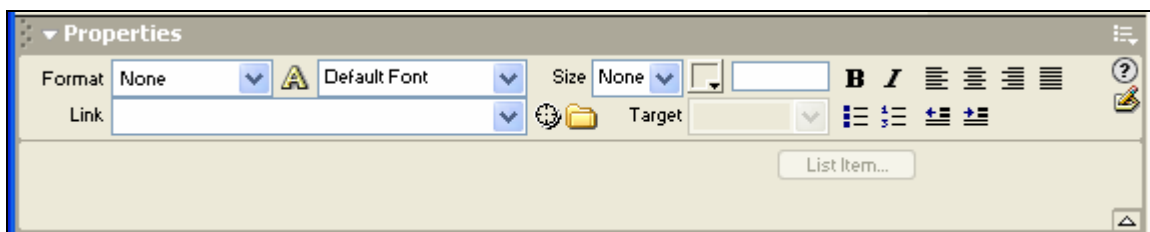


## CHƯƠNG IX: ĐỊNH DẠNG VĂN BẢN- SỬ DỤNG CSS

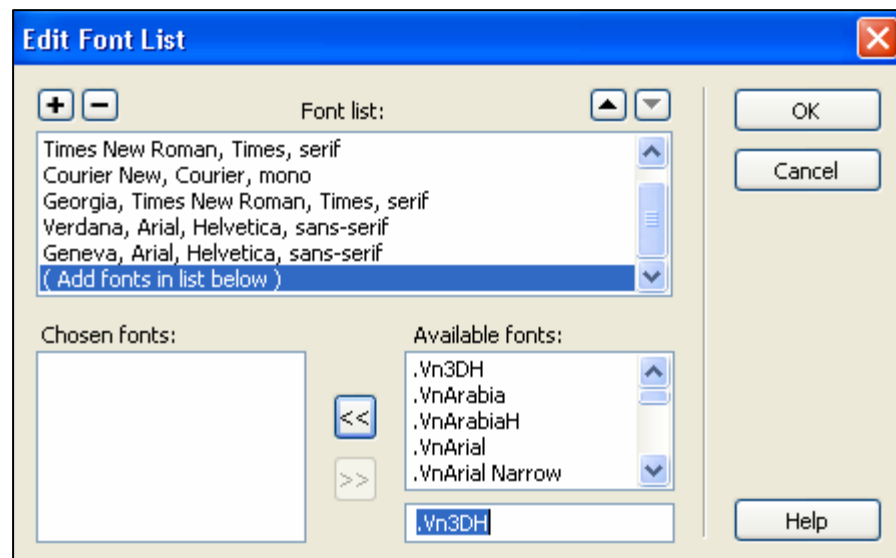
### IX.1. ĐỊNH DẠNG VĂN BẢN

- Cách nhập giống như các trình soạn thảo văn bản khác, mỗi phần nội dung được phân cách bằng cách xuống dòng, nếu xuống dòng trong cùng một đoạn (Paragraph) thì nhấn Shift + Enter, nếu ngắt đoạn thì nhấn Enter.
- Để hiệu chỉnh văn bản thường sử dụng thanh công cụ Properties Inspector. Cách tổng quát là đánh dấu khối văn bản → chọn kiểu định dạng
- Sử dụng thanh công cụ Properties Inspector:

#### IX.1.1. Font:



- Cách 1: Tại mục format Chọn các heading, đây là các định dạng mẫu, bao gồm Font chữ, kiểu chữ, size, ... thường dùng làm tiêu đề
- Cách chọn nhóm Font chữ: Chọn văn bản, rồi chọn nhóm Font trên Font menu của Properties Inspector hoặc chọn menu Text → Font. Trong Dreamweaver, kiểu Font chữ được định thành từng nhóm, mỗi nhóm gồm nhiều font, một Font chính và các Font dự phòng. Có thể tạo ra các nhóm Font tùy ý bằng cách tại mục Font → Chọn Edit Font List



- Chọn Font trong khung Available Fonts, Click nút >> đưa các font được chọn qua khung Chosen fonts

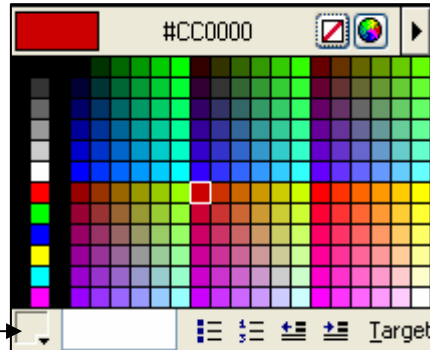
- + Tạo thêm nhóm Font mới
- Xoá nhóm ra khỏi Font List
- ▲ ▼ Sắp xếp các nhóm Font theo thứ tự

**1. Font size:**

- Chọn khối văn bản, Chọn cỡ chữ trong mục Size của Properties Inspector, hoặc chọn Text → Size. Size chữ trong Dreamweaver gồm 17 Font Size, trong đó có 8 mức thể hiện bằng số, từ 9 đến 36 và 9 mức thể hiện bằng chữ

**2. Font Color:**

- Chọn khối văn bản, Click nút Text Color, chọn màu hoặc chọn Text → Color



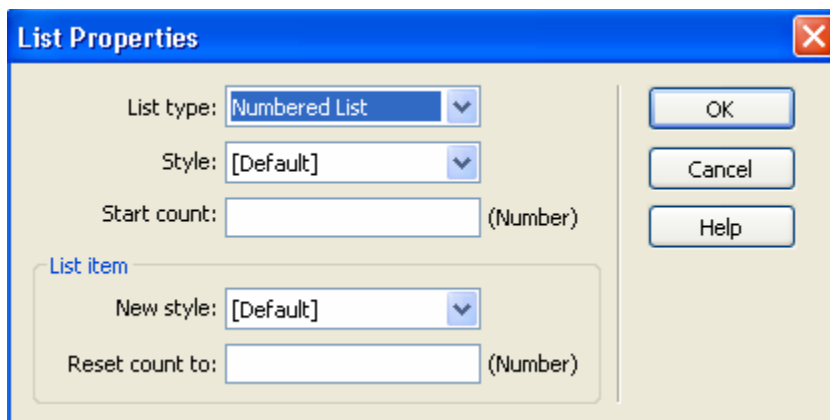
Click nút Text color chọn

**3. Canh lề đoạn văn bản**

- Chọn Text → Align hoặc Click công cụ

**IX.1.2. Danh sách dạng liệt kê:**

- Chọn Text → List
- Unordered List: Chèn Bulleted đầu dòng
- Ordered List: Đánh số thứ tự đầu dòng
  - ❖ Thay đổi thuộc tính liệt kê:
- Đặt dấu nhảy trong danh sách liệt kê
- Chọn Text → List → Properties hoặc click nút List Item. Xuất hiện hộp thoại List Properties



- List Type: Chọn kiểu danh sách (Bullets hoặc Numbered)
- Style: Loại Chấm tròn hoặc vuông
- Start count: Số bắt đầu cho danh sách liệt kê
  - ❖ List item:
- New Style: liệt kê nhiều cấp
- Reset count to: số bắt đầu cho danh sách con

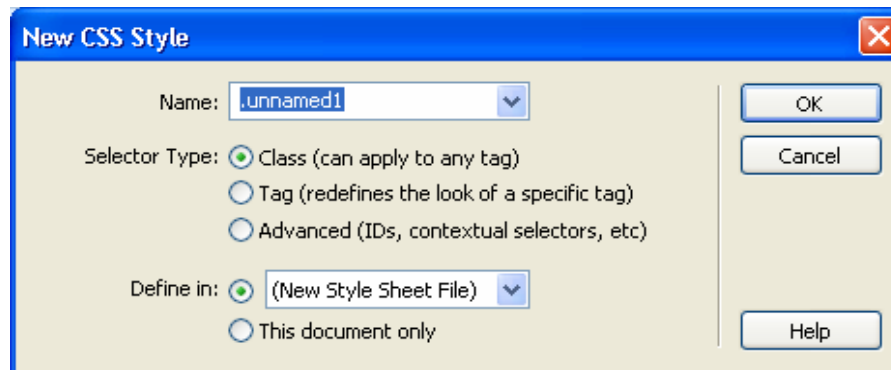
## IX.2. SỬ DỤNG CSS (CASCADING STYLE SHEETS)

- CSS (Cascading Style Sheets) cũng là một dạng HTML Style. Nhưng phong phú hơn về thuộc tính và ứng dụng. Một CSS không những tập hợp các định dạng, mà còn có thể giúp định vị, viền khung, đặt màu nền...
- CSS có thể đính kèm trong trang hoặc lưu riêng thành một tập tin kiểu CSS phục vụ cùng lúc cho nhiều trang

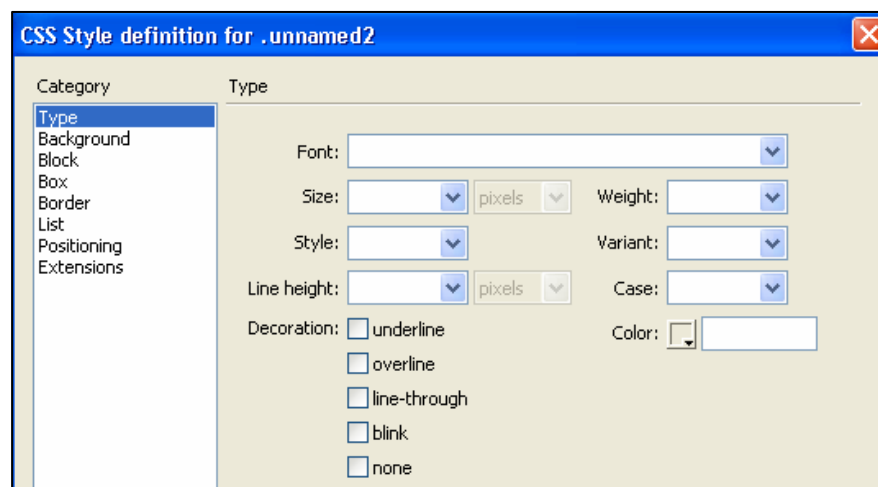
### IX.2.1. Tạo CSS cục bộ:

#### 1. Cách tạo:

- Chọn Text → CSS Styles → New... Xuất hiện hộp thoại New CSS Style:



- Hoặc Window → CSS Style, mở CSS Panel, Click nút New CSS Style
- Name: Tên của CSS mới, phải bắt đầu bằng dấu chấm(.)
- Selector Type: Chọn loại CSS
- Define in: New Style Sheet File: tạo một tập tin CSS
- This document only: Chỉ sử dụng CSS cho trang cục bộ
- Chọn OK, xuất hiện hộp thoại CSS Style definition



- Trong mục category, chọn Type, sau đó chọn các định dạng cho CSS
- Chọn xong, Click Apply → OK
- Trong CSS Panel xuất hiện Style vừa tạo

#### 2. Áp dụng CSS cục bộ:

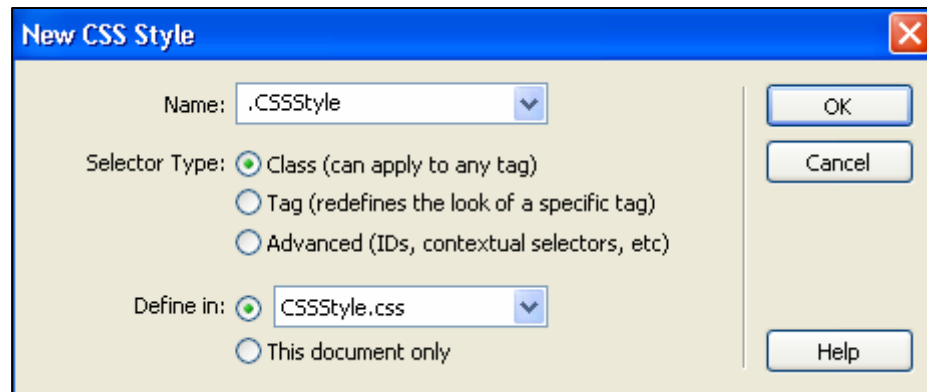
- Chọn nội dung văn bản cần định dạng
- Trong CSS Style Panel, chọn tên CSS

## IX.2.2. Tạo một tập tin CSS:

Tập tin kiểu CSS là một tập tin phụ trợ cho Site, nằm trong thư mục root của Site

### 1. Cách tạo:

- Text → CSS Styles → New
- Trong hộp thoại New CSS Style
- Selector Type: Chọn Advanced
- Define in: New Style Sheet file, OK
- Hộp thoại yêu cầu lưu tập tin CSS, có phần mở rộng .CSS



### 2. Áp dụng CSS từ tập tin CSS:

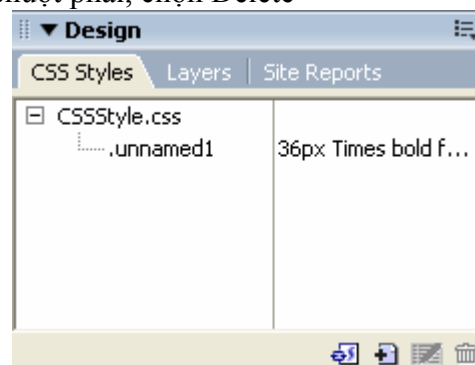
- Mở trang HTML cần sử dụng tập tin CSS
- Chọn Text → CSS Style → Attach Style Sheet.
- Hoặc Click nút Attach Style Sheet trong Style Panel.
- Chọn tập tin CSS cần kết nối, Click nút Browse...
- Add as:
  - o Link: Chỉ liên kết với tập tin CSS để sử dụng
  - o Import : Chép tập tin CSS vào trang

### 3. Hiệu chỉnh một CSS:

- Click phải trên tên CSS trong CSS Style Panel
- Chọn Edit, thực hiện hiệu chỉnh

### 4. Xoá một CSS Styles:

- Khi xoá một CSS Style thì những nội dung áp dụng CSS Style bị xoá sẽ trở về trạng thái ban đầu
- Chọn CSS Style cần xoá
- Click nút Delete CSS Style trong CSS Style Panel
- Hoặc Click chuột phải, chọn Delete



Attach Style Sheet      New Style Sheet      Delete Style

## CHƯƠNG X: HÌNH ẢNH VÀ LIÊN KẾT TRANG TRONG DREAMWEAVER

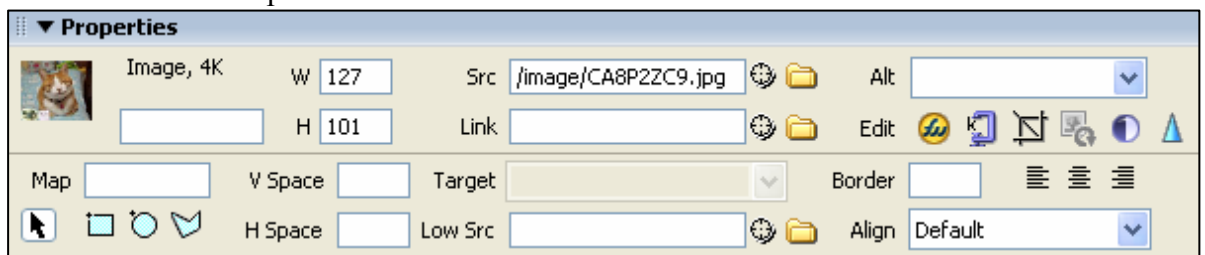
### X.1. CHÈN HÌNH ẢNH VÀO TRANG WEB:


#### X.1.1. Chèn ảnh vào trang:

- Ảnh trong thư mục Images của Site:
  - Đặt dấu nháy tại vị trí cần chèn ảnh
  - Drag chuột kéo tập tin ảnh trong Site Panel thả vào trang
- Ảnh ngoài Site:
  - Chọn Insert → Image
  - Xuất hiện hộp thoại Select Image Source
  - Chọn tập tin ảnh cần chèn → OK

#### X.1.2. Hiệu chỉnh thuộc tính của ảnh:

- Chọn ảnh đã chèn
- Window → Properties



- Image: Đặt tên cho ảnh
- W (Width), H (Height): Độ rộng và chiều cao của ảnh, tính bằng Pixel
- Src: đường dẫn tương đối đến tập tin ảnh
- Alt: câu thông báo xuất hiện trên trình duyệt khi rê chuột vào ảnh
- Link: Địa chỉ URL nơi cần liên kết đến
- Edit: Chuyển qua Macromedia Fire Works hiệu chỉnh ảnh 



Crop: Cắt xén ảnh



Brightness/Contrast: Chỉnh độ sáng tối của ảnh



Sharpen: Chỉnh độ sắc nét cho ảnh



Resample: Lưu lại kích thước đã điều chỉnh

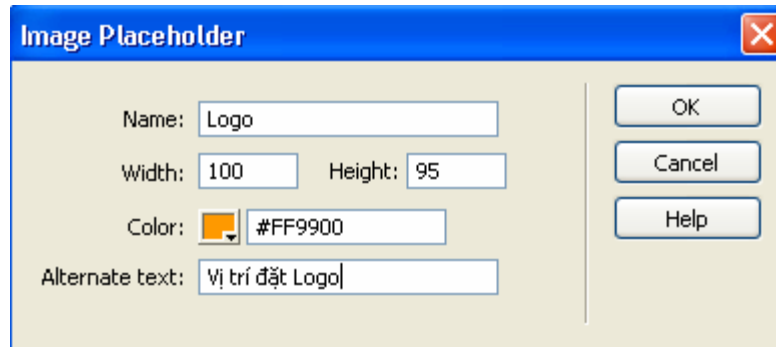


Optimize in Fireworks: chuyển qua Macromedia FireWoks để hiệu chỉnh

- Map : bảng đồ liên kết ảnh
- VSpace, Hspace: Khoảng cách trên, dưới, trái, phải giữa phần nội dung văn bản đến ảnh
- Target: Khung chứa trang liên kết đến
- Low Src: tên tập tin ảnh phụ có độ phân giải thấp, làm ảnh thay thế khi chờ hiển thị ảnh chính trên trình duyệt
- Border: đường viền ảnh
- Align: canh lề trái, phải, giữa...

### X.1.3. Chèn khung ảnh:

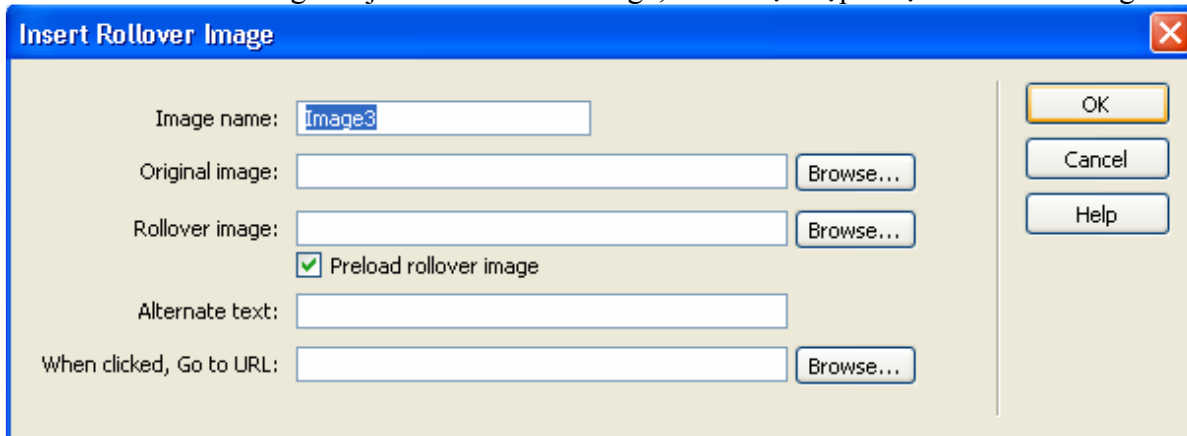
- ❖ Trong thiết kế, nhiều lúc cần dự phòng trước cho ảnh trang trí, nhưng chưa có ảnh thích hợp, ta có thể chèn trước một khung ảnh với kích thước xác định để giữ chỗ
  - Chọn Insert → Image Objects → Image Placeholder
  - Xuất hiện hộp thoại Image Placeholder
  - Nhập tên, kích thước, màu cho khung ảnh



- ❖ Chèn ảnh vào khung ảnh:
  - Double click vào khung cần chèn ảnh
  - Xuất hiện hộp thoại Select Image Source, chọn tập tin ảnh cần chèn vào khung

### X.1.4. Insert Rollover Image:

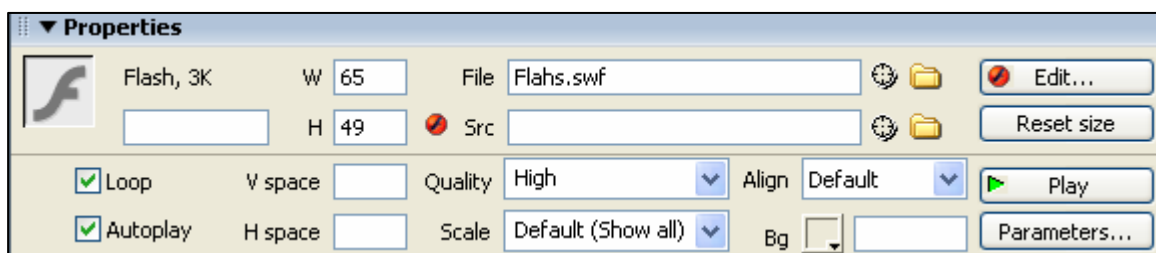
- Insert → Image Objects → Rollover Image, Xuất hiện hộp thoại Rollover Image



- Original Image: ảnh gốc
- Rollover Image: ảnh khi rê chuột vào

### X.1.5. Chèn Flash:

- Insert → Media → Flash
- Chọn tập tin kiểu .swf
- Tại vị trí chèn xuất hiện biểu tượng Flahs
- ❖ Hiệu chỉnh thuộc tính của Flash



- Flash: Tên đối tượng Flash
- W (Width), H (Height): Chiều rộng và chiều cao của ảnh Flash
- File: tên tập tin Shockware của Flash
- Src: tên tập tin gốc của Flash
- Edit: Hiệu chỉnh trong Macromedia Flash
- Reset size: trả về kích thước ban đầu
- Loop: ảnh Flash lập vô tận
- AutoPlay: tự động diễn hoạt khi mở trang
- Vspace, Hspace: khoảng cách trên, dưới, trái, phải của đối tượng Flash đến văn bản
- Quality: chất lượng khi hiển thị
- Show All: hiển thị tất cả đối tượng trong khung
- No Border: không giới hạn trong khung viền
- Exact fit: vừa khít trong khung viền
- Align: canh lề
- Bg: màu nền dưới ảnh Flash

### X.1.6. Ảnh nền trang

- Ảnh nền là ảnh tự động lặp đầy trang Web. Khi thiết kế, bạn nên chọn những mẫu nền thật nhạt, chữ sậm hoặc nền thật sậm, chữ màu sáng để người xem dễ đọc
- Tùy thuộc vào từng loại nền mà xác định kích thước ảnh nền cho phù hợp, tạo ảnh nền với số Kb càng nhỏ thì trang hiển thị càng nhanh

#### 1. Cách tạo ảnh nền:

- Đặt trỏ trong trang
- chọn Modify → Page Properties
- Background Images : nhập đường dẫn đến tập tin ảnh làm nền

#### 2. Ảnh nền trang cố định không lặp

Đây là dạng nền khó thực hiện, kích thước ảnh thường rất lớn, nên phải nén để giảm số Kb tối đa,

#### ❖ Tạo ảnh nền cố định, không lặp bằng CSS:

- Thiết kế ảnh cần làm nền
- Chọn Text → CSS Styles → New CSS Style, Trong hộp thoại New CSS Style:
- Selector: Nhập tên Style
- Selector Style: Chọn mục Advanced
- Define in: This document only → OK, Xuất hiện hộp thoại CSS Style definition:
- Trong mục Category, chọn Background
- Background Image: tên tập tin ảnh làm nền
- Repeat: No Repeat (không lặp)
- Attachment: fixed (nền cố định không bị cuộn)
- Horizontal, vertical Position: center → OK

#### ❖ Sử dụng CSS ảnh nền

- Khi tạo CSS Style, trong CSS Style Panel xuất hiện tên Style vừa tạo
- Để đưa CSS Style ảnh nền vừa tạo làm nền cho trang, Click phải trên Tag <Body> của trang, chọn Set Class → Chọn tên Style vừa tạo

### X.1.7. Tạo Web Photo album:

1. **Chức năng:** Tạo bộ sưu tập các hình ảnh hay cuốn Album giúp người sử dụng quản lý và chọn xem từng hình một cách nhanh nhất
2. **Cách tạo:** Để thực hiện chức năng này cần phải cài đặt Macromedia Fireworks và một thư mục chứa các hình photo
  - Chọn Commands → Create Web Photo album
  - Xuất hiện hộp thoại Create Web Photo Album
  - Nhập các thông số:



- Photo Album Title: Nhập tiêu đề của Album
- Subheading Info : Nhập tiêu đề phụ ( Xuất hiện dưới tiêu đề chính)
- Other Info: Những thông tin bổ xung
- Source Images Folder: Địa chỉ của folder chứa hình
- Destination folder: Địa chỉ của Photo Album sau khi tạo
- Thumbnail size: Kích thước của hình trong trang chính
- Show files name : Ghi/ Không ghi tên file dưới mỗi hình
- Column: số cột hình có trong trang chính
- Thumbnail format: Chọn kiểu định dạng hình trong trang Index
- Chọn kiểu định dạng hình trong trang con
- Scale : tỉ lệ hình so với trang Index
- Creat navigation page for each photo: Có / không tạo cho mỗi hình một trang riêng để bổ xung thêm trong tin

Nhập xong click OK, chờ kết quả, xuất hiện thông báo Album đã được tạo, Vào cửa sổ Site Panel xuất hiện thêm các folder:

- Folder Thumbnail: chứa các file JPG
  - Folder Page chứa các file .HTM cho mỗi Image tương ứng (trang con)
  - Tập tin Index.htm trong Folder chứa Website, đây là tập tin Album chính
- Mở tập tin Index.htm và di chuyển giữa các trang bằng các Hyperlink Next Previous, Home

### 3. Bổ sung thông tin cho mỗi trang con:

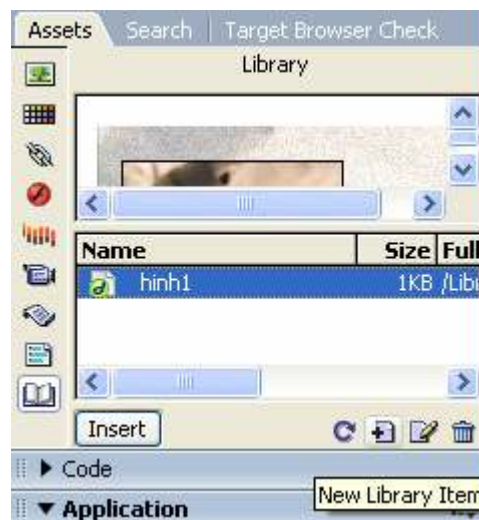
- Trong Site Panel, mở tập tin muốn bổ sung thông tin trong Folder Page
- Nhập thông tin, trình bày theo ý muốn, lưu lại
- Đổi tên tập tin này thành Gallery.htm



- Gán định dạng Template vào cho File Gallery.htm, mở tập tin Index, xem kết quả

### X.1.8. Tạo Library Item:

- 1. Khái niệm:** Library chứa các thành phần của trang như hình ảnh, văn bản và các đối tượng khác mà ta có nhu cầu sử dụng lại hoặc cần cập nhật thường xuyên. Các thành phần này gọi là Library Items. Cách tạo Library Item:
- 2. Cách tạo:** Mở AssetsPanel (Window → Assets), trong Assets Panel, Click nút Library, Chọn thành phần muốn tạo Library Item và thực hiện một trong các bước sau
  - Drag chuột kéo thành phần được chọn thả vào khung Library → Đặt tên
  - Click nút New Library Item và đặt tên
  - Chọn Modify → Library → Add Object to Library và đặt tên
- 3. Nhập Library Item vào trang mới:**
  - Đặt dấu nháy vào nơi muốn nhập Library Item
  - Kéo Library Item từ Assets Panel thả vào document Window, hoặc click nút Insert trong Assets Panel
- 4. Hiệu chỉnh Library Item:**
  - Chọn Library Item trong khung Library → Click nút Edit
  - Xuất hiện hộp thoại cho phép hiệu chỉnh library Item
  - Click Save → xuất hiện hộp thoại bạn có muốn cập nhật tất cả trang Web có sử dụng Library trong site không, Yes : để cập nhật, No : không.
  - Sau này có thể cập nhật bằng cách chọn Modify → Library → Update pages: cập nhật tất cả những trang có sử dụng Library Item
- 5. Tách Library Item trong Document khỏi Library:**
  - Chọn Library Item trong trang Web hiện hành
  - Click nút Detach from Original trên Properties Inspector, khi đó library bị tách ra thành từng đối tượng và không còn liên quan đến Library, ta có thể hiệu chỉnh từng đối tượng
  - Tạo lại Library Item bằng một library đang sử dụng trong trang
  - Có thể dùng một Library trong trang để tạo lại một Library Item, nếu Library Item bị mất
  - Chọn Library trên trang Web hiện hành
  - Click nút Recreate trên Properties Inspector, Library Item sẽ được tạo lại

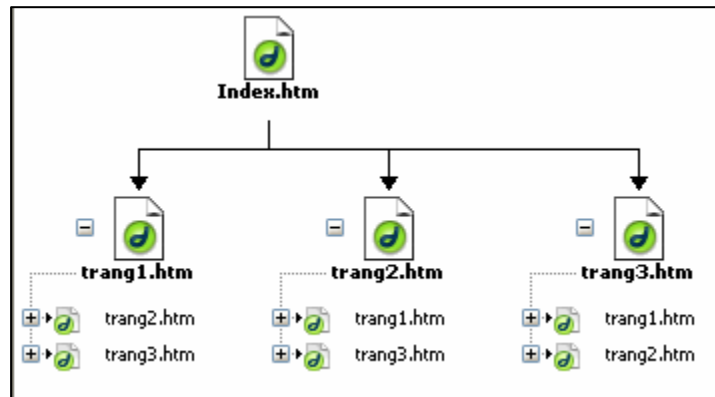


## X.2. LIÊN KẾT TRANG TRONG DREAMWEAVER:

### X.2.1. Liên kết :

Một liên kết nối từ trang nguồn đến trang đích gồm 2 thành phần

- Đối tượng được chọn làm nút liên kết có thể là văn bản (Text), hình ảnh (Image) hoặc nút (Button). Trong Macromedia Dreamweaver MX 2004 còn cung cấp thêm một số đối tượng đặc biệt làm nút liên kết như Flash Text, Flash Button, Navigation bar, Rollover Images...



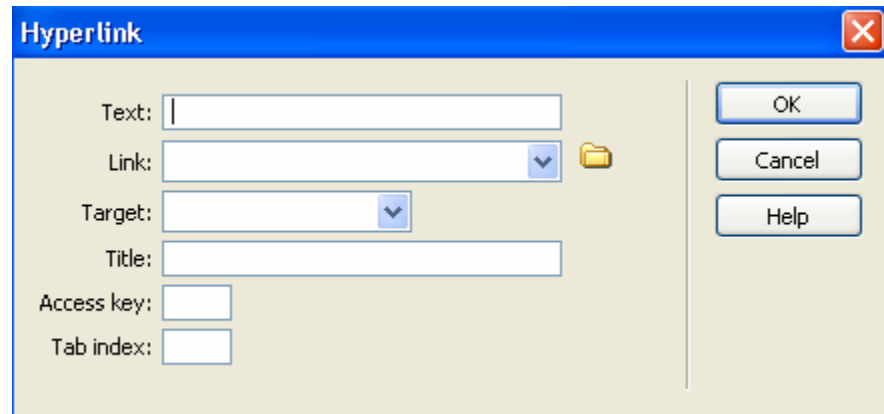
- Địa chỉ URL của trang cần liên kết đến: Tùy thuộc vào địa chỉ URL mà có thể chia làm 3 loại liên kết
  - Liên kết nội: Là liên kết nội bộ các trang trong cùng một Website.
  - Liên kết ngoại: Là liên kết đến trang thuộc một Website khác.
  - Liên kết Email: Là liên kết đến chương trình gửi thư tín điện tử.

### X.2.2. Các dạng liên kết:

- Dạng liên kết vòng: Là dạng liên kết nối đuôi nhau, trang 1 liên kết đến trang 2, trang 2 liên kết đến trang 3, ... trang n liên kết đến trang 1, dạng liên kết này đảm bảo người xem có thể xem tất cả các trang, nhưng bất lợi là phải duyệt hết một vòng.
- Dạng liên kết đầy đủ: Tại mỗi trang đều tạo liên kết đầy đủ đến tất cả các trang còn lại, đây là dạng liên kết trong site có độ phân cấp thấp. Nếu Site có nhiều phân cấp thì nên chọn dạng liên kết cây phân cấp.
- Dạng liên kết cây phân cấp: Trong các Site lớn, mức độ quan trọng của từng trang được phân cấp theo từng mức, với trang gốc là trang chủ, mức 1 là nhóm chủ đề chính, mức 2 là nhóm chủ đề con, mức 3 là trang chứa các thông tin chi tiết, ... Ở dạng này tồn tại liên kết giữa các trang cùng mức (Same Level), liên kết về mức trên (Parent Level), liên kết về mức dưới (Child level).
- Dạng liên kết tiện nghi: Ngoài các dạng liên kết trên, để thuận tiện cho người xem khi lật trang, có thể tạo thêm một số liên kết phụ, như liên kết đến điểm dừng (Bookmark) trong trang có nội dung dài, nhiều phân đoạn, hoặc tạo thêm một hệ thống liên kết text ở cuối mỗi trang, khi người xem đến cuối trang có thể nhanh chóng lật sang trang khác mà không cần trở về
- Liên kết trang chủ: Trang chủ thường chứa các nút liên kết đến các trang con, do đó cần phải tạo liên kết từ trang con trở về trang chủ

**X.2.3. Cách tạo:**

- Mở trang nguồn
- Chọn Insert → Hyperlink



- Text: nội dung văn bản làm nút liên kết
- Link: Địa chỉ URL của trang cần liên kết đến
- Target: Khung chứa trang đích
- Title: Câu ghi chú khi chuột chạm vào nút
- Access Key: Khi xem trang, nhấn tổ hợp phím Alt + Ký tự nhập để chọn nút và nhấn Enter để liên kết.
- Tab Index: trình tự chọn nút khi nhấn phím Tab

**1. Tạo liên kết nội:**

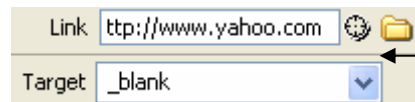
Liên kết giữa các trang trong nội bộ Site, đây là liên kết đơn giản, chỉ cần chọn tên tập tin trang đích, sử dụng địa chỉ tương đối, hoặc trong mục Link nhập tên tập tin đích

**2. Tạo liên kết ngoại**

Tạo liên kết đến các tập tin ngoài site, cần phải nhập địa chỉ http://của trang đích thuộc Site khác

Khi tạo liên kết ngoại, người xem có thể qua site khác có thể không tiếp tục xem site của bạn nữa, để tránh trường hợp này, có thể thực hiện một trong các cách sau:

- Liên hệ với người chủ của Site ngoài, hợp đồng tạo quan hệ qua lại giữa 2 Website
- Đưa Site ngoài vào trong khung (Frame ) của Site mình
- Mở thêm một cửa sổ mới, sau đó đưa WebSite này vào. Cách mở cửa sổ mới cho liên kết với WebSite ngoài
- Trong Properties Inspector, Tại mục Target nhập tên của khung chứa trang đích. Chọn \_blank: mở một cửa sổ khung trống để chứa trang đích



Liên kết với trang Yahoo trong cửa sổ khác

**3. Tạo liên kết địa chỉ Email:**

Đây là liên kết đến chương trình gửi nhận thư điện tử, giúp người xem có thể liên hệ, trao đổi với người chủ của Website

Cách 1:

- Chọn đối tượng làm nút liên kết Email
- Trong ô Link nhập trực tiếp dòng mailto: theo sau là địa chỉ Email của người chủ Website


Cách 2:

- Không cần tạo đối tượng làm nút liên kết
- Chọn Insert → Email Link
- Trong hộp thoại Email link:
  - Text*: nhập nội dung văn bản làm nút
  - E-Mail* : nhập đúng địa chỉ E-mail

#### 4. Tạo liên kết đến tập tin để người xem tải xuống máy cục bộ:

Có những tập tin bạn muốn cung cấp cho người xem trang nhưng chúng không phải là các trang HTML, ví dụ một số dạng font chữ mới, tập tin nén, hoặc các chương trình nhỏ .EXE...

Cách tạo:

- Mở trang cần tạo liên kết
- Chọn đối tượng làm nút liên kết
- Trong Properties Inspector, trong mục Link, nhập vào tên tập tin cần liên kết (hoặc Click nút  để mở hộp thoại Select File tìm tập tin cần cung cấp cho người xem. Nếu tập tin này không ở cùng Site thì xuất hiện hộp thoại yêu cầu chép tập tin này vào site

#### 5. Liên kết điểm dừng (Named Anchor)

a) Tạo liên kết điểm dừng cùng trang:

Đối với các trang Web dài, hoặc trang có nhiều mục, thì việc dùng thanh cuộn không được tiện, do đó có thể giúp người xem đến từng mục một cách nhanh chóng, chính xác nhờ vào việc đặt sẵn các điểm dừng. Cách tạo gồm 2 bước:

- Đặt tên cho điểm dừng:
  - Trong Document window, đặt dấu nháy tại vị trí sẽ làm điểm dừng
  - Chọn Insert → Named Anchor (Ctrl+Alt+A) hoặc Click nút Insert Named Anchor trong bảng Common của thanh Insert
  - Trong hộp thoại Insert Anchor: Nhập tên cho Anchor (không thừa khoảng trắng, không có ký tự lạ)
- Tạo liên kết điểm dừng:
  - Trong Document Windows, chọn một đoạn văn bản hoặc một hình để tạo link đến điểm dừng
  - Nếu điểm dừng nằm cùng trang thì tại ô link nhập #tên Anchor

b) Tạo liên kết điểm dừng của trang khác:

Thao tác tạo liên kết đến điểm dừng ở trang khác giống như liên kết đến trang khác, nhưng tại mục link phải chỉ ra điểm dừng nào. Theo cấu trúc:<Tên tập tin>#<Tên điểm dừng>

#### X.2.4. Kiểm tra liên kết:

##### 1. Kiểm tra liên kết bằng Check Link:

Ta có thể sử dụng Check Links để có một bảng thống kê tất cả các liên kết trong Website.

- Mở trang cần kiểm tra liên kết
- Chọn File → Check Page → Check links
- Trong Results inspector → Chọn mục Link Checker
- Trong mục Show: Chọn Broken Link → hiển thị danh sách liên kết gãy
- Chọn External links: hiển thị danh sách liên kết ngoài

##### 2. Kiểm tra liên kết bằng trình duyệt

- Chọn File → Preview in Browser

- Có thể thay đổi trình duyệt bằng cách chọn File → Preview in Browse → Edit Browser List
- Chọn tên trình duyệt trong phần Browser, chọn iexplorer. Nếu không tìm thấy thì click nút (+) để thêm trình duyệt mới vào danh sách Browser, click dấu (-) để bỏ bớt trình duyệt trong danh sách
  - o Default: Mặc định độ ưu tiên trình duyệt khi xem trang
  - o Option: Preview Using Temporary file: chọn chức năng này để tạo tập tin tạm tương ứng với trang cần xem. Trình duyệt sẽ hiển thị trang tạm, nếu không chọn, bạn sẽ xem trực tiếp trang trên trình duyệt.

### X.2.5. Hiệu chỉnh liên kết:

- Chọn nút liên kết cần thay đổi
- Chọn menu Modify → Change Link...
- Hộp thoại Select file cho phép chọn tập tin trang cần liên kết đến
- Nếu biết rõ tập tin liên kết mới thì có thể nhập trực tiếp trong ô link của Properties Inspector

#### 1. Xoá liên kết

- Chọn nút muốn loại bỏ liên kết
- Chọn Modify → Remove Link. Hoặc xoá tên trang liên kết trong ô link của Properties Inspector

#### 2. Chọn khung cho trang liên kết

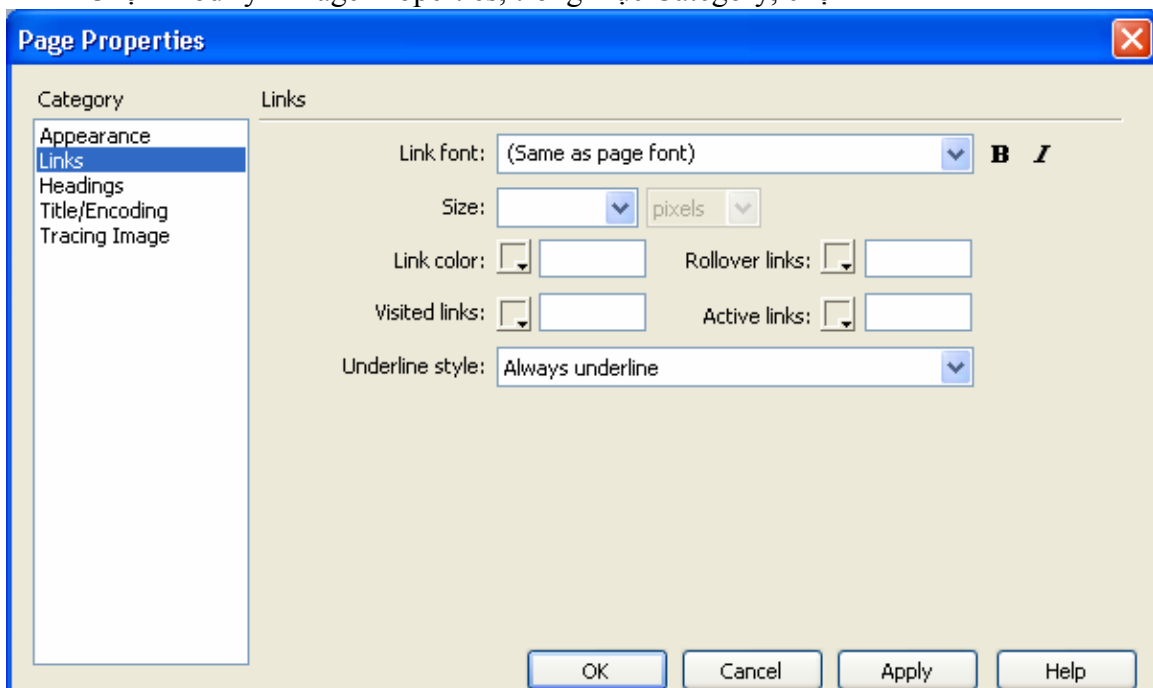
Khi liên kết thường sử dụng trang khung (Frameset)

- Chọn nút muốn thay đổi khung liên kết
- Chọn menu Modify → Link Target
- Chọn tên khung phù hợp

#### 3. Chọn màu cho liên kết

Thường một liên kết dạng văn bản sẽ được gạch dưới và có màu xanh, đôi khi màu này không phù hợp với màu sách trong trang thiết kế, có thể chọn lại màu liên kết bằng cách:

- Chọn Modify → Page Properties, trong mục Category, chọn Link



- Links color: chọn màu mặc định cho nút liên kết
- Visited Links: màu liên kết đến các trang đã được xem
- Active Links: màu nút liên kết khi đang được chọn

### X.2.6. Bản đồ ảnh liên kết :

Khi chọn ảnh làm liên kết, thì có một số ảnh kích thước lớn, thường chia nhỏ ảnh đó ra thành nhiều vùng mỗi vùng liên kết đến một trang Web khác, dạng này gọi là bản đồ ảnh liên kết.

#### 1. Những thuận tiện khi sử dụng bản đồ ảnh liên kết:


- Giúp tạo nhanh các liên kết
- Hình ảnh trực quan, dễ liên tưởng đến trang tương ứng.
- Giúp bố cục các liên kết nhanh, không chiếm nhiều khu vực nút trên trang Web

#### 2. Cách tạo:

- Chèn ảnh vào trang, click chọn ảnh
- Trong Properties inspector, hiển thị công cụ Map
- Chọn công cụ muốn chia vùng
- Drag chuột quanh phần trên hình mà ta muốn chia vùng để tạo liên kết
- Trong Properties inspector, tại mục link, nhập địa chỉ của trang cần liên kết đến



#### 3. Hiệu chỉnh bản đồ liên kết:

- Di chuyển vùng liên kết
- Click chọn công cụ  để chọn vùng cần di chuyển
- Drag chuột kéo đến vị trí mới

#### 4. Thay đổi kích thước vùng liên kết

- Chọn vùng liên kết
- Click vào một trong các nút chọn của vùng liên kết
- Drag chuột để thay đổi kích thước

#### 5. Xoá vùng liên kết

- Chọn vùng liên kết
- Nhấn Delete

### X.2.7. Nhóm ảnh động làm nút liên kết:

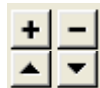
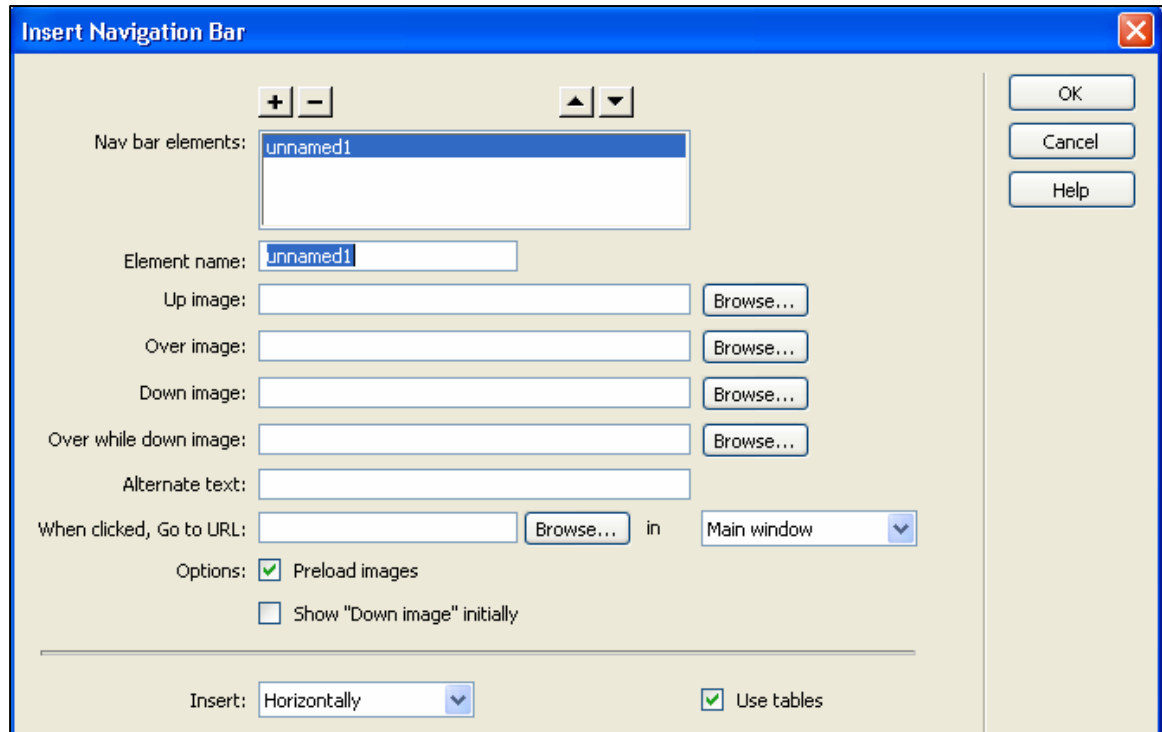
#### 1. Chèn nút biến đổi hình:

Trước hết phải có 2 tập tin ảnh, n1.gif màu cam, n2.gif màu xanh trong folder Images của Site

- Đặt dấu nháy tại vị trí cần chèn nút động
- Chọn Insert → Image Objects → Rollover Image. Hộp thoại Insert Rollover Image:
  - Image Name: Nhập tên ảnh
  - Original Image: tên tập tin ảnh gốc đại diện khi hiển thị (ví dụ n1.gif)
  - Rollover Image : tên tập tin ảnh hiển thị khi rê chuột vào (ví dụ n2.gif)
  - Preload Rollover Image: đưa ảnh sẽ thay thế vào bộ nhớ trước
  - Alternate Text: câu ghi chú kèm theo
  - When Click, go to URL: Địa chỉ của trang liên kết đến

#### 2. Chèn hệ thống nút biến đổi hình:

- Dreamweaver có thể giúp bạn cùng một lúc chèn vào cả một hệ thống nút biến hình
- Chọn Insert → Image Objects → Navigation bar



Thêm hoặc xóa nút trong hệ

Sắp xếp trình tự nút trong hệ nút

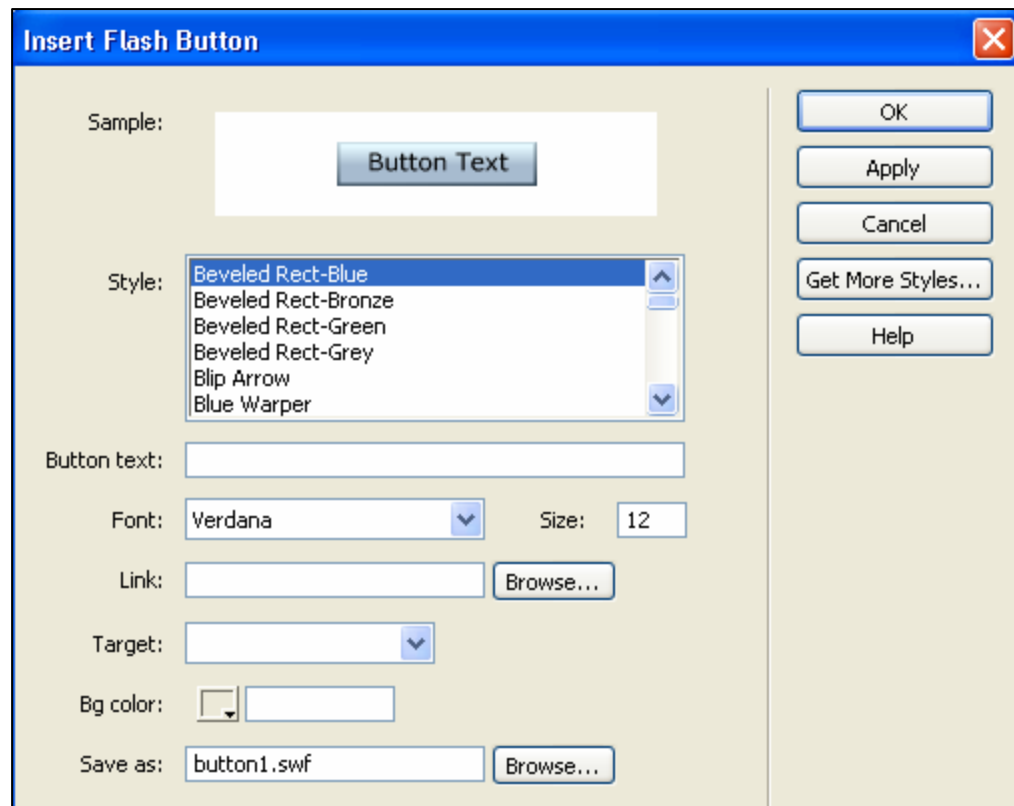
- *Nav bar Elements*: danh sách tên nút trong hệ nút
- *Element Name*: Nhập tên nút mới
- *Up Image*: Ảnh hiển thị ở trạng thái thường
- *Over Image*: Ảnh hiển thị khi rê chuột vào nút
- *Down Image*: Ảnh hiển thị khi Click chuột vào nút
- *Alternate Text*: câu ghi chú kèm theo ảnh
- *When Click, go to URL*: Địa chỉ trang liên kết đến
- *Preload Image*: Đưa ảnh sẽ thay thế vào bộ nhớ trước
- *ShowDownImageInitaly*: Khởi tạo chọn ảnh ở trạng thái ẩn
- *Insert*: Horizontally:hệ nút ngang
- *Insert Vertically*: hệ nút dọc
- *Use Table*: Hệ nút nằm trong bảng

### 3. Chèn nút Flash:

Macromedia Dreamweaver và Macromedia Flash là một bộ chương trình có thể sử dụng dữ liệu qua lại với nhau rất thuận tiện, do đó có thể dùng các nút Flash được thiết kế sẵn để làm nút liên kết trong Dreamweaver

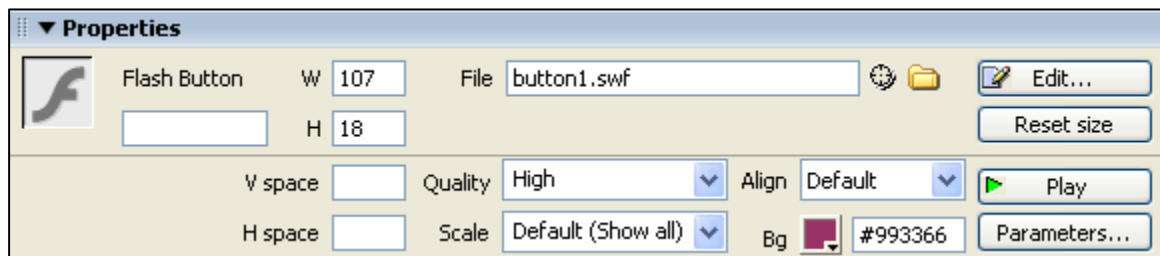
#### ❖ Cách tạo:

- Chọn Insert→Media→ Flash Button
  - Sample: Ví dụ mẫu nút Flash
  - Style: danh sách tên các nút mẫu Flash
  - Button Text: văn bản trên nút Flash
  - Font: kiểu chữ, Size: Cỡ chữ
  - Link: Địa chỉ liên kết đến
  - Target: Tên khung trang liên kết
  - Bg: Màu nền



❖ Hiệu chỉnh nút Flash:

- Chọn nút Flash cần hiệu chỉnh
- Trong Properties inspector, thay đổi giá trị của các thuộc tính:
  - Flash Button: tên nút Flash
  - H: chiều cao
  - File: tên tập tin SWF là nút
  - Edit: mở hộp thoại Insert flash Button để hiệu chỉnh
  - Reset Size: trở về kích thước ban đầu
  - Vspace, Hspace : khoảng cách giữa nút và nội dung văn bản
  - Quality: Chất lượng nút Flash
  - Scale: Kéo dẫn nút
  - Align: canh lề so với văn bản
  - Bg: màu nền của nút
  - Play: Xem hoạt động và các trạng thái của nút
  - Parameters: truyền tham số





**4. Chèn nút Flash Text:**


- Insert → Media → Flash Text
- Hộp thoại Insert Flash Text, nhập vào các thông số:
  - o Font, Size: Font và cỡ chữ
  - o Color: màu chữ
  - o Rollover Color: Màu chữ thay đổi khi rê chuột qua nút
  - o Text : Nội dung văn bản làm nút
  - o Link: Địa chỉ trang Web liên kết đến
  - o Target: Tên khung trang
  - o Bg Color: màu nền của văn bản nút

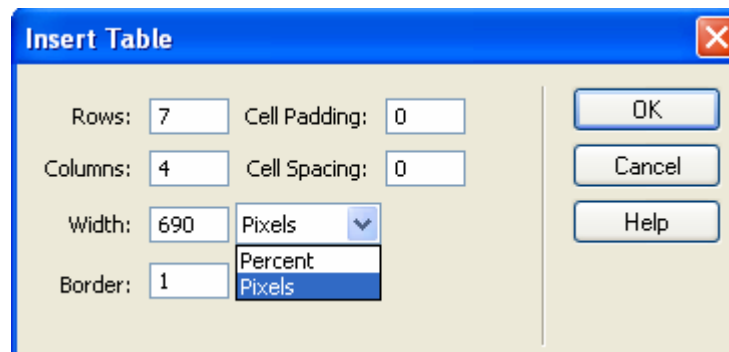
## CHƯƠNG XI: BẢNG VÀ TRÌNH BÀY TRANG

### XI.1. TABLE :

#### XI.1.1. Kẻ bảng:

Tùy thuộc vào từng bố cục cụ thể mà quyết định số dòng, cột của bảng, mỗi dòng có thể có số ô khác nhau. Thường chúng ta nên chọn dòng có số ô lớn nhất làm chuẩn khi kẻ bảng.

- Insert/Table, hoặc ấn trong nhóm Table. 
  - Rows: số dòng cần chèn.
  - Columns: số cột cần chèn.
  - Width: chiều rộng của bảng theo số điểm pixels hoặc phần trăm
  - Border: độ dày của đường viền bảng.
  - Cell Padding: khoảng cách nội dung ô và biên ô.
  - Cell Spacing: khoảng cách giữa các ô.



#### XI.1.2. Hiệu chỉnh bảng

##### 1. Chèn thêm dòng, cột vào bảng:

- ❖ Thêm dòng cuối bảng
  - Định vị con trỏ ở ô cuối cùng trong bảng.
  - Ấn phím Tab để thêm dòng.
 Cách khác:
  - Ấn vào viền bảng, hay ấn thẻ <Table> chọn bảng.
  - Trong properties inspector nhập số dòng mới.
- ❖ Chèn thêm cột vào bảng:
  - Trong properties inspector nhập số cột mới.

##### 2. Xoá dòng, cột, bảng

- Chọn dòng, cột, bảng
- Chọn thực đơn Edit/Cut. (Ctrl +X).

##### 3. Nối các ô trong bảng:

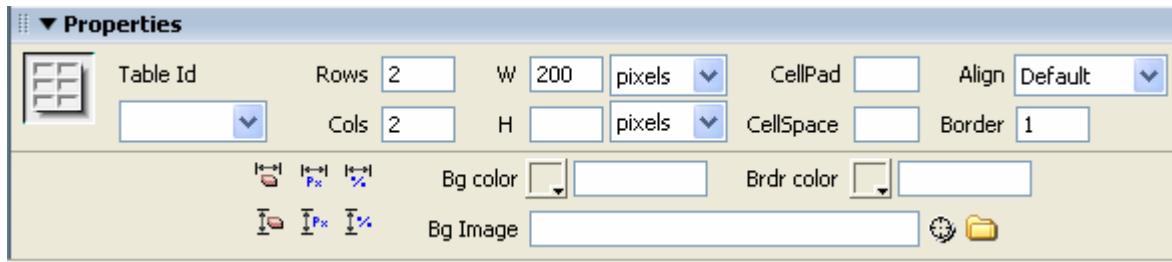
- Chọn các ô cần nối
- Modify → Table → Merge Cells.

##### 4. Tách các ô trong bảng:

- Chọn ô cần tách
- Modify → Table → Splits Cell
  - Split Cell into Columns: tách ô thành nhiều ô theo cột.
  - Split Cell into Rows: tách ô thành nhiều ô theo dòng.
  - Number of columns, Rows: xác định số ô cần tách theo cột, dòng.

### XI.1.3. Thuộc tính của bảng:

- Chọn table → mở Properties inspector.
- Nhập giá trị mới trong Properties inspector.



- Rows: số dòng.
- Cols : số cột.
- W(width) : chiều rộng theo điểm pixel, theo tỷ lệ màn hình.
- H (height) : chiều cao mặc định, tùy vào nội dung.
- Cellpad : khoảng cách văn bản đến ô trong bảng.
- Cellspace : khoảng cách giữa các ô trong bảng.
- Align : canh lề bảng, phải, trái, giữa.
- Border : độ dày nét đường viền bảng.
- Bg color : màu nền của bảng.
- Bg image : ảnh nền bảng.
- Brdr color : màu đường viền bảng.

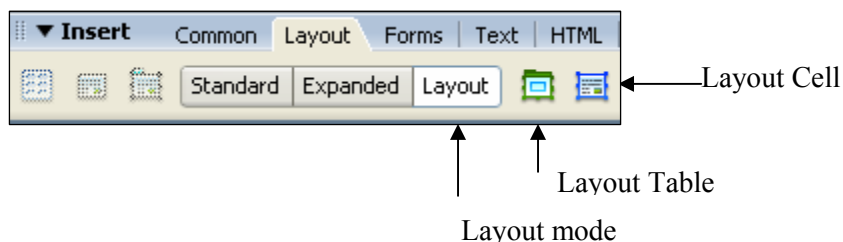
## XI.2. TRÌNH BÀY TRANG :

### XI.2.1. Layout Table và layout cell

Trình bày trang là một công đoạn quan trọng nhất trong thiết kế Web, đòi hỏi tính mỹ thuật và độ chính xác cao. Macromedia Dreamweaver MX 2004 cung cấp các công cụ rất tốt trong việc thiết kế và trình bày trang đó là Layout Tble và Layout Cell

#### 1. Layout table:

- Layout table là dạng biến thể của table với các thông số đi kèm như khung viền Border=0, khoảng cách giữa các ô CellSpace =0, khoảng cách giữa nội dung trong ô và viền ô CellPad=0
- Layout table dùng để phân vùng cho trang, nếu trong trang có nhiều nội dung với những chủ đề khác nhau hoặc cần nhập nội dung với dạng cột báo chí thì dùng layout table để bố cục trang theo chủ đề được chuẩn bị trước
- Layout Table dùng để bố cục trang, không dùng để chứa dữ liệu, viền khung của Layout Table có màu xanh lá cây
- Khi thiết kế dạng Layout, cần phải chuyển sang chế độ Layout view, trong Insert Inspector, chọn tab Layout, chọn Layout mode, hoặc chọn View → Table Mode → Layout Mode

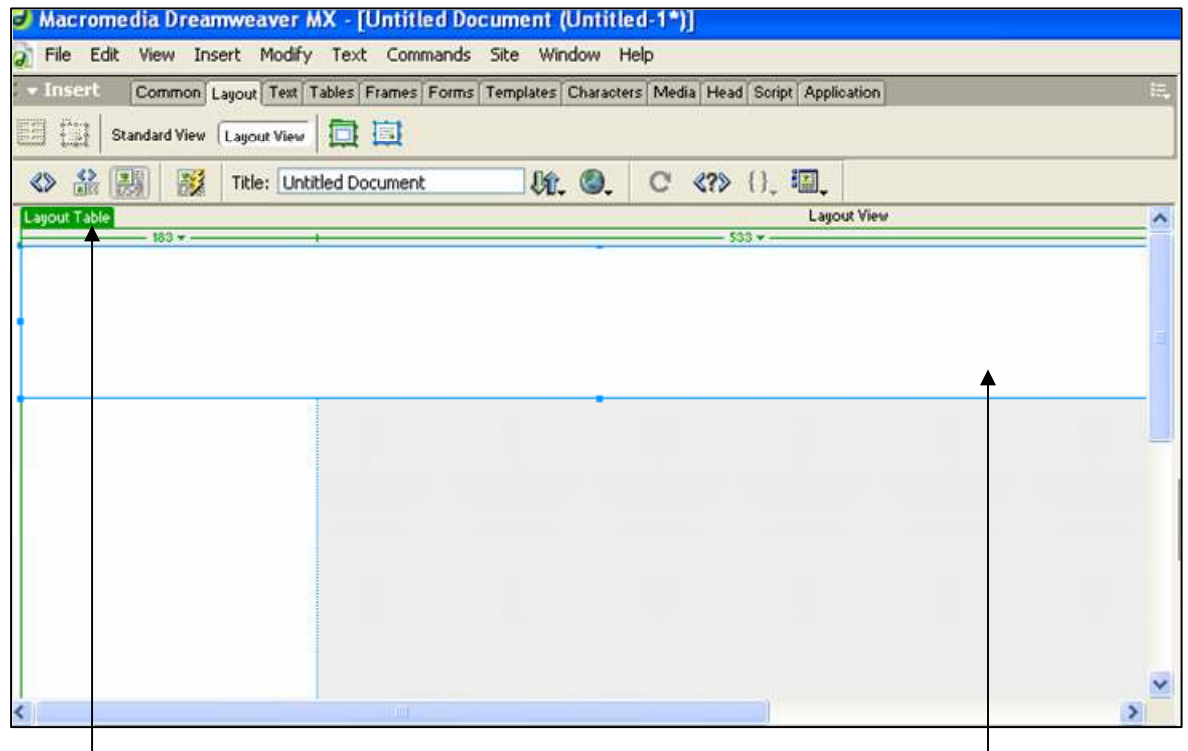


## 2. Layout cell:

Layout cell: Nằm trong Layout table, dùng để chứa dữ liệu, dữ liệu trong layout cell có thể là văn bản, hình ảnh, khi thiết kế dạng layout cần lưu ý các layout cell phải sát nhau để tránh trường hợp làm chi trang bị nát

Một layout Table có thể chứa nhiều layout table con Mỗi Layout Table gồm có nhiều dòng, mỗi dòng chứa nhiều Layout Cell, số Layout Cell trên mỗi dòng có thể khác nhau

### Ví dụ:



Layout table bố cục trang

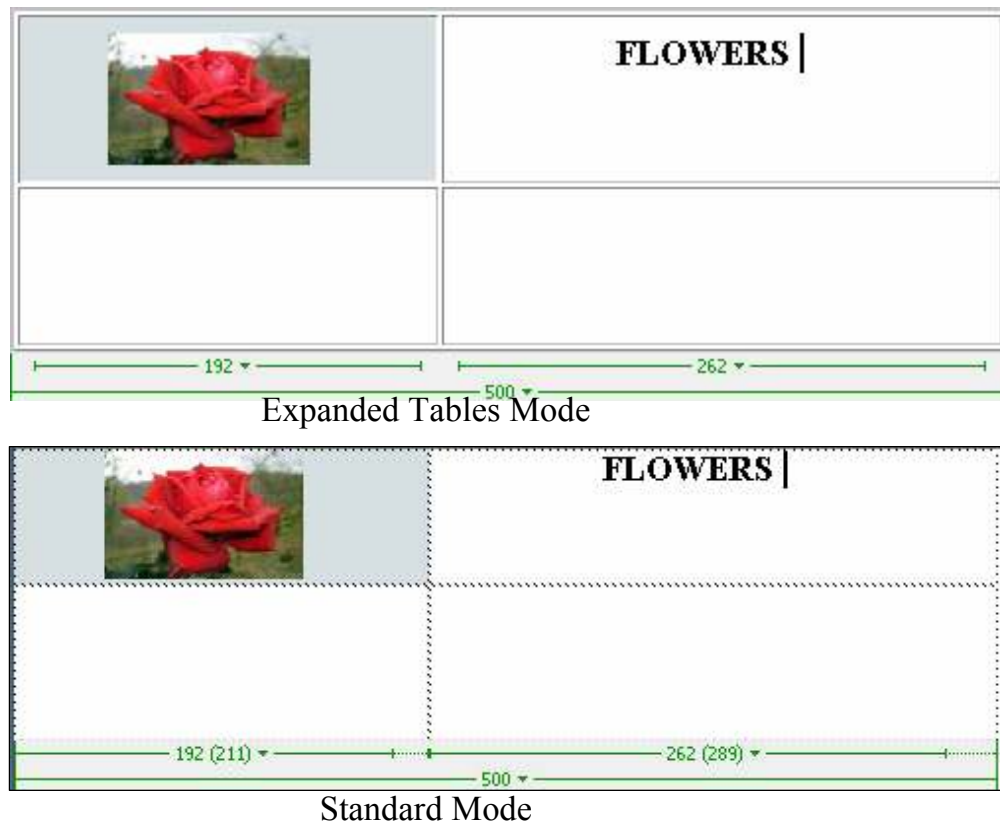
Layout cell chứa dữ liệu

## 3. Một số cách kết hợp Layout Table và Layout Cell:

- Vẽ một Layout Table có kích thước đầy trang, sau đó vẽ các Layout cell bên trong Layout Table theo đúng kích thước và yêu cầu của bố cục
- Vẽ nhiều Layout Table cùng cấp
  - o Layout Table trên chứa Logo, Banner, nút ngang.
  - o Layout Table ở giữa chứa nội dung văn bản, hình ảnh...
  - o Layout Table dưới chứa địa chỉ liên lạc, phone...
- Hoặc kết hợp cả 2 cách trên, dùng 2 Layout table ngang cấp:
  - o Layout Table trên chứa Logo, banner, nút ngang...
  - o Layout Table dưới chứa 2 Layout table con, một bên trái và một bên phải...

### Lưu ý :

- o Khi vẽ một Layout Cell bên ngoài Layout Table thì Dreamweaver tự phát sinh một Layout Table chứa Layout Cell đó
- o Chế độ Expanded Tables : cho hiển thị khoảng cách từ nội dung trong ô đến đường viền của Table, tiện cho việc hiệu chỉnh độ rộng của ô

**Ví dụ:**

1) Thiết kế trang với một Layout Table chính:

- Trong Document Window
- Chọn tab Layout, click nút Layout Table Drag chuột vẽ trong Document Window
- Khung viền Layout Table có 3 handle dùng để thay đổi kích thước của Layout Table, trên khung viền có số chỉ chiều rộng của Layout Table
- Thiết kế trang với 3 Layout Table ngang cấp:
- Thực hiện giống như trên, khi kẻ Layout Table phải bắt đầu từ biên trái trang

2) Thiết kế trang với các Layout Table lồng nhau:





Vì Macromedia Dreamweaver MX 2004 không cho phép kẻ các Layout Table phía bên phải, nên bắt buộc phải kẻ một Layout Table lớn canh trái, sau đó mới có thể kẻ lồng bên trong Layout Table lớn một Layout Table con.

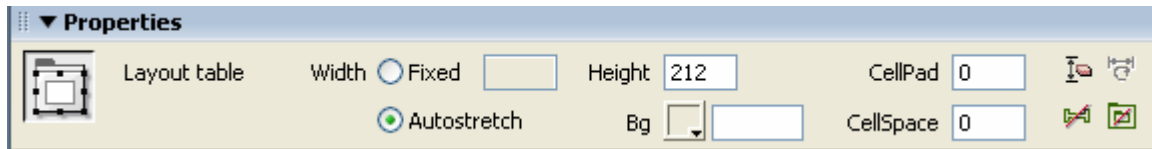
**XI.2.2. Thuộc tính của Layout Table và layout cell:**

Có thể hiệu chỉnh kích thước bằng cách drag chuột kéo các handle của khung, nhưng nếu cần kích thước chính xác thì phải nhập các thông số trong Properties Inspector của Layout Table

**1. Hiệu chỉnh thuộc tính của layout table:**

- Width:
  - o Fixed: số Pixel xác định chiều rộng
  - o AutoStretch: tự động kéo dãn ngang theo nội dung văn bản hoặc hình ảnh chèn vào Layout Table
- Height: Xác định số Pixel chiều cao, nhỏ nhất là 19 Pixel
- Bg: màu nền của Layout Table
- CellPad: khoảng cách từ nội dung đến biên của Layout Table
- CellSpace: Khoảng cách giữa các Layout Cell

- Clear Row Height:  tự động thay đổi chiều cao của các dòng cho vừa khít với nội dung, nếu không có nội dung thì chiều cao của dòng ít nhất là 19 Pixel
- Remove All Spacers:  Có hiệu lực khi chọn AutoStretch (xoá tất cả khoảng trống thừa)
- Make cells Width Consistent:  tạo các cell trong Layout Table có chiều rộng như nhau.
- Remove Nesting :  xoá Layout Table con trong các Layout Table cha.

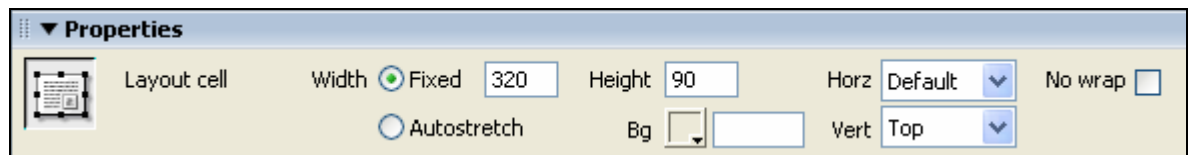


## 2. Xoá Layout Table:

- Chọn viền của khung Layout Table hoặc click thẻ <Table>
- Nhấn Delete

## 3. Hiệu chỉnh thuộc tính của Layout Cell:

- Chọn layout cell
- Mở thanh Properties Inspector của Layout Cell



- Width:
  - Fixed: Số Pixel xác định chiều rộng
  - AutoStretch: tự động kéo dãn ngang theo nội dung văn bản hoặc hình ảnh chèn vào Layout Cell
- Height: Xác định số Pixel chiều cao, nhỏ nhất là 19 Pixel
- Bg: màu nền của Layout Cell
- Horz: Canh lề cho nội dung trong Layout Cell theo chiều ngang (Left, Center, Right)
- Vert: Canh lề theo cho nội dung trong Layout Cell theo chiều dọc
- No wrap: khi nội dung dài hơn kích thước của Layout Cell, nếu chọn mục này thì văn bản không xuống dòng mà Layout Cell tự dẫn ra, nếu không chọn thì văn bản tự xuống dòng khi gặp lề phải của Layout Cell
- Di chuyển một Layout Cell:
  - Chọn Layout Cell (Ctrl+ Click chuột vào Layout Cell) cần di chuyển
  - Drag chuột kéo đến vị trí mới
- Xoá một Layout Cell
  - Chọn Layout Cell cần xoá
  - Nhấn phím Delete

### XI.2.3. Các lớp Layer:

Layer là một thành phần mới trong thiết kế Web, nó có thể chứa nội dung văn bản, hình ảnh xếp chồng lên nhau, nổi trên trang và chuyển động rất linh hoạt Layer thường được sử dụng để thiết kế trang có các hiệu ứng đặc biệt như chữ rơi, ảnh bay,...

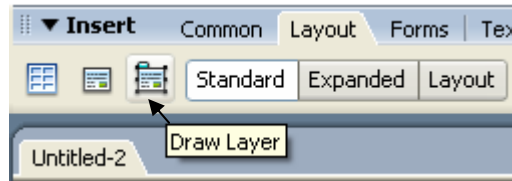
Điểm bất lợi của Layer là không hiển thị trên các trình duyệt cũ như IE4.0, Netscape 4.0, đối với các trình duyệt mới thì Layer hiển thị một cách trung thực

## 1. Cách tạo Layer trên trang:

Có thể tạo Layer bằng một trong các cách sau:

Cách 1:

- Chọn Standard Mode
- Click nút Draw Layer, drag chuột vẽ Layer



Cách 2:

- Chọn menu Insert → Layout Objects → Chọn Layer

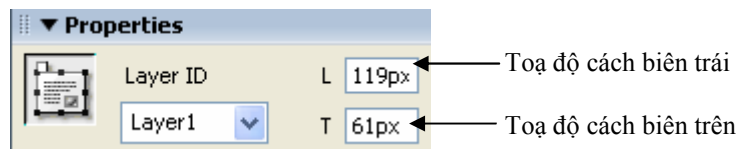
## 2. Hiệu chỉnh Layer:

- Chọn layer, Layer được chọn sẽ xuất hiện 8 Handle xung quanh, khi đó ta thực hiện các thao tác hiệu chỉnh trên Layer đó

a. *Di chuyển Layer:*

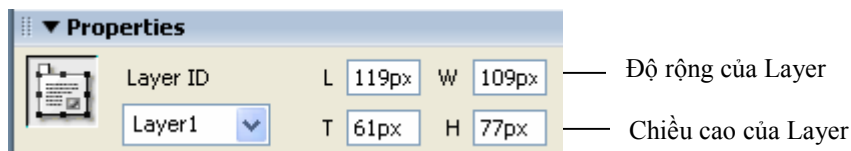
- Chọn Layer cần di chuyển
- Drag chuột kéo Layer đến vị trí mới

Hoặc nhập giá trị tọa độ trong Properties Inspector



b. *Thay đổi kích thước của Layer:*

- Chọn Layer cần hiệu chỉnh kích thước
- Click chuột vào một trong các Handle, Drag chuột để thay đổi kích thước
- Hoặc nhập thông số trực tiếp vào Properties Inspector



c. *Chèn nội dung vào Layer:*

- Nếu nội dung là văn bản thì nhập trực tiếp vào Layer
- Nếu nội dung là hình ảnh thì drag chuột chọn hình trong thư mục Image thả vào Layer (hoặc chọn Insert Image)

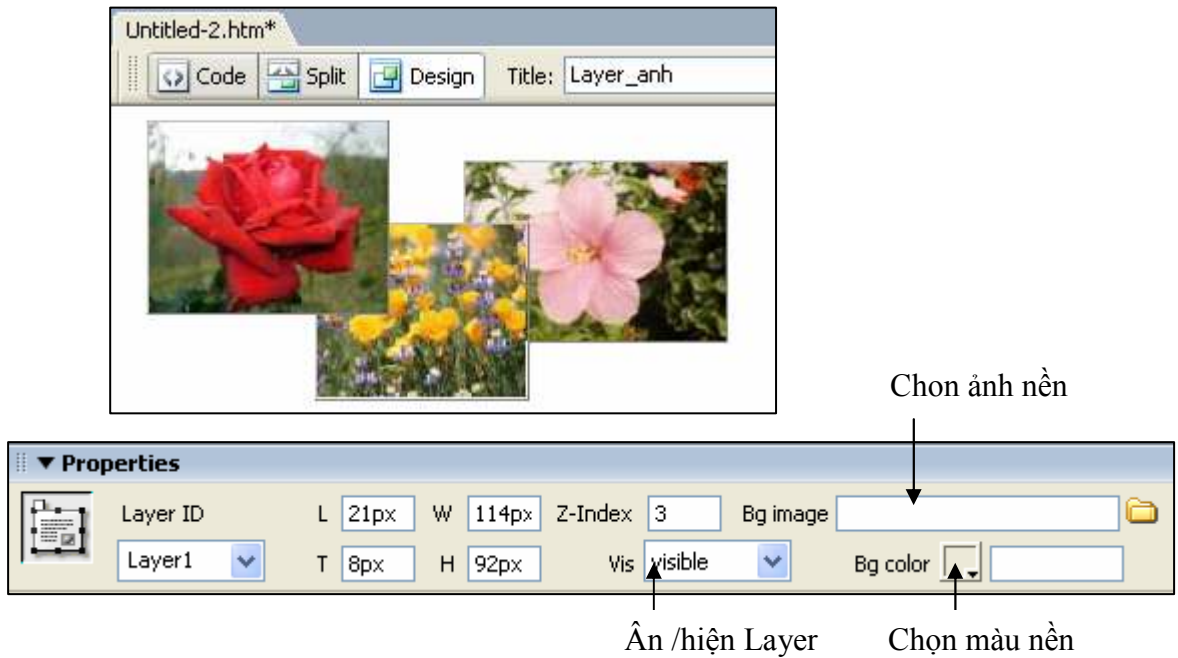
d. *Xếp chồng các Layer:*

Khi cần hiển thị nhiều ảnh trên trang, nhưng không đủ chỗ, ta có thể xếp chồng lên nhau, sau đó cho xuất hiện từng lớp một hoặc cho từng lớp Layer bay ra khỏi màn hình, điều này có thể thực hiện được khi kết hợp Layer, Timeline và Behaviors

e. *Thứ tự các Layer:*

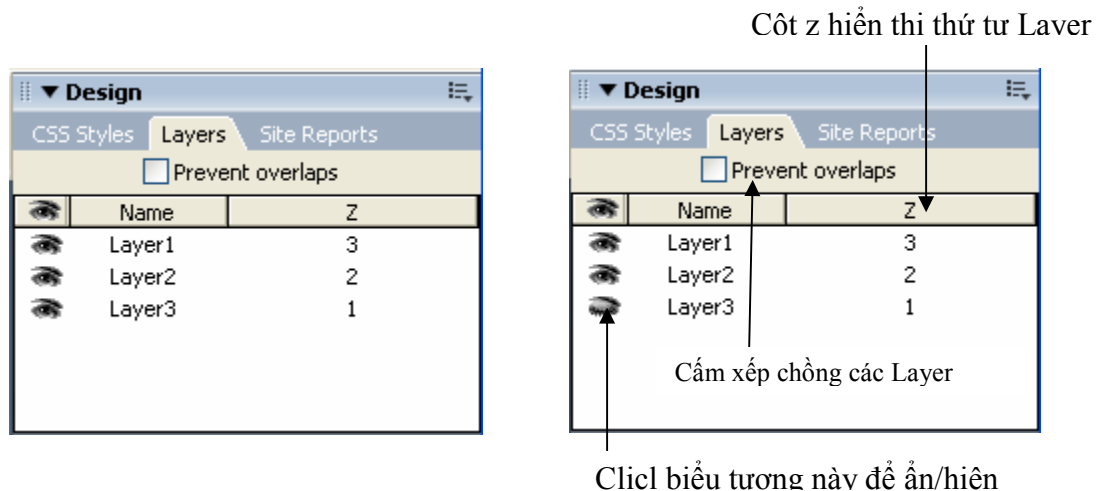
Mỗi lớp Layer đều có thuộc tính Z-Index hiển thị thứ tự trước, sau của các lớp Layer, lớp Layer sau sẽ che khuất lớp Layer trước nếu nó có cùng tọa độ, có thể thay đổi trình tự các lớp layer bằng cách:

- Chọn Lớp Layer cần thay đổi thứ tự
- Trong Properties Inspector, nhập thứ tự mới trong mục Z-index



### 3. Ẩn hiện một Layer:

Khi không muốn xem Layer nào thì ẩn Layer đó bằng một trong các cách sau: Trong Properties Inspector, tại thuộc tính Vis: chọn Hidden. Hoặc mở Layer Panel:

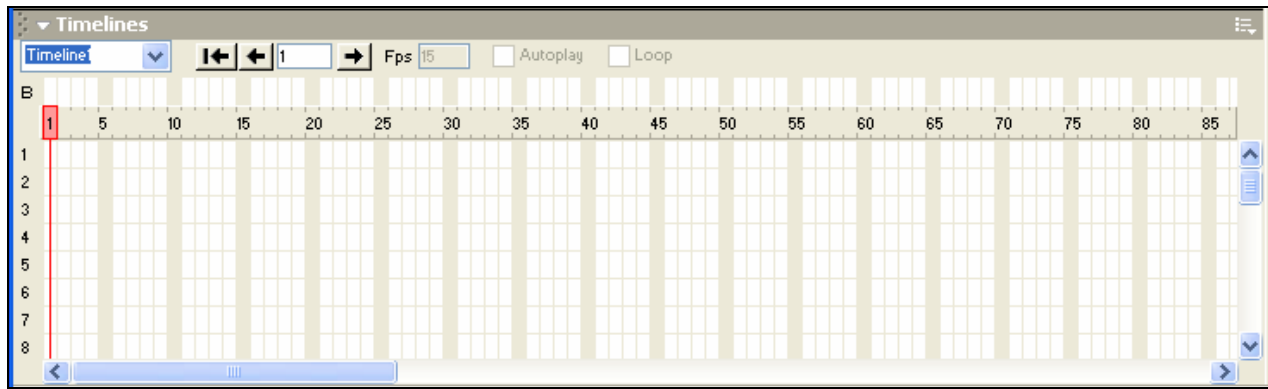


#### XI.2.4. Timeline Panel:

Timeline Panel là một bảng sắp xếp ảnh, lớp Layer theo một trình tự xuất hiện trên trục thời gian, nó giúp tạo các hình ảnh động.

- Mở Timeline Panel: Window → Others → Timeline





- Trực thời gian là trục hoành
- Trục không gian là trục tung
- Fps: (Frame per Second) tốc độ chạy đầu đọc theo số khung hình trên giây

### 1. Một số lưu ý khi sử dụng Timeline:

- Đối tượng trong Timeline là ảnh và lớp Layer
- Các đối tượng khác muốn sử dụng Timeline thì phải đưa nội dung vào Layer trước, sau đó mới đưa vào Timeline Panel

### 2. Cách hiệu chỉnh ảnh trong Timeline Panel:

- Chỉ có thể hiệu chỉnh thuộc tính SRC tên tập tin ảnh mà thôi
- Để tạo một ảnh chuyển động, cần chèn ảnh trong lớp Layer
- Sau đó chèn lớp layer này vào Timeline Panel

### 3. Hiệu chỉnh lớp Layer trong Timeline Panel:

Chỉ có thể hiệu chỉnh một số thuộc tính của lớp Layer như: vị trí (left, top) kích thước (width, Height), thứ tự lớp (Z-index) và thuộc tính có thể xem (Visibility)  
Không thể thay đổi thuộc tính chiều rộng và chiều cao trong trình duyệt Navigator4.0

## XI.3. TẠO ALBUM LẬT TỪNG HÌNH:

### Ví dụ 1:

- Tạo một album ảnh lật từng hình sau khoảng thời gian nhất định
- Chuẩn bị 3 hình có cùng kích thước có tên lần lượt là H1.gif, H2.gif, H3.gif
- Chèn tập tin ảnh H1.gif vào trang
- Chọn ảnh, trong Properties Inspector: nhập tên ảnh: H1, Src: H1.gif
- Kéo H1 trực tiếp vào Timeline Panel.
- H1 chiếm 15 khung hình từ 1 đến 15
- Kéo H1.gif vào Timeline thêm 1 lần nữa và sửa Src: H2.gif
- Lặp lại việc kéo hình vào Timeline Panel và sửa Src: H3.gif
- Trong TimeLine Panel, chọn Fps=15, nghĩa là đầu đọc chạy 15 khung hình/1s, mỗi hình chiếm 15 khung, do đó sau 1s sẽ chuyển sang hình mới
- Chọn AutoPlay để việc lật hình tự động diễn ra khi xem trang
- Chọn Loop để khi đến H3.gif thì sẽ lặp tiếp H1.gif
- Lưu tập tin, nhấn F12 xem lật hình trên trình duyệt

### Ví dụ 2:

- Tạo chuyển động cho máy bay từ trái sang phải của màn hình, từ tọa độ (0,0) đến tọa độ (600,0)
- Trong Standard view
- Chọn Insert → Layer, đặt tên Layer\_Maybay, T=0, L=0
- Chèn ảnh máy bay vào Layer

- Chọn Layer kéo vào TimeLine Panel, chiếm từ khung hình 1 đến 15
- Chọn khung hình 15 trong Layer\_Maybay
- Trong Properties Inspector, nhập L=600, T=0, Fps=15, chọn AutoPlay, chọn Loop
- Lưu tập tin, nhấn F12 xem kết quả trên trình duyệt

**Ví dụ 3:**

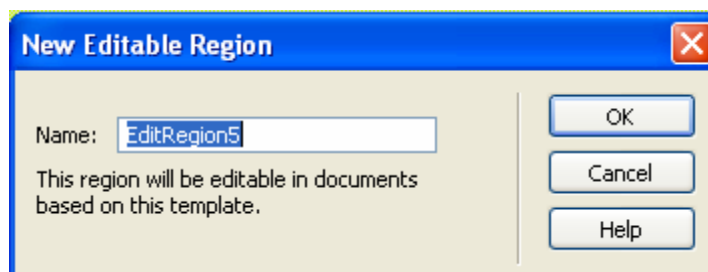
- Tạo 2 chuyển động xảy ra cùng một lúc, khi máy bay đang bay thì hoa tuyết rơi từ trên xuống
- Chèn thêm một lớp Layer 2, đặt tên Layer\_Tuyet, L=300, T=0
- Chèn ảnh hoa tuyết vào Layer này
- Kéo Layer\_Tuyết vào trong TimeLine Panel, ở kênh thứ 2, từ khung hình 5 đến 15
- Chọn khung 15 của lớp Layer\_Tuyết ở kênh 2
- Trong Properties Inspector, nhập L=300, T=200.
- Lưu tập tin, nhấn F12 xem kết quả, khi máy bay đang bay thì một hoa tuyết rơi xuống từ toạ độ (300,0)

**XI.4. TEMPLATE:**

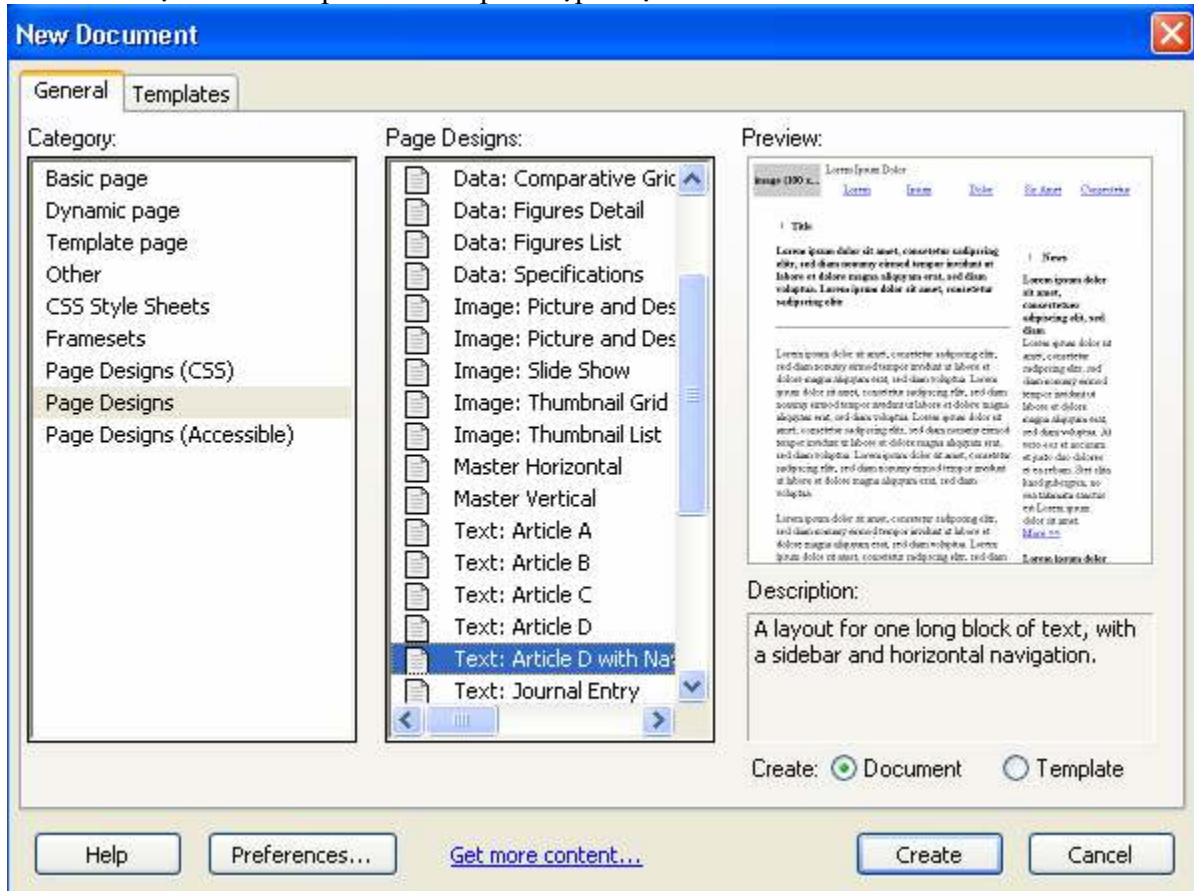
- Template không những giúp ta có thể sử dụng các thành phần dùng chung mà còn giữ quan hệ giữa các thành phần trong một mẫu trang được thiết kế, bố cục sẵn. Ta có thể căn cứ vào một mẫu template để tạo nhanh nhiều trang có cùng một bố cục thiết kế. Thao tác với template, ta cần phân biệt rõ giữa trang mẫu và trang sử dụng template.
- *Trang mẫu template*  
Là tập tin kiểu **.dwt** với phần thiết kế chuẩn cho một bố cục dùng chung, trong trang có 2 loại vùng: vùng được khoá và vùng không khoá.
- *Trang sử dụng template:*  
Là tập tin kiểu **.htm** nhưng có bố cục giống như trang mẫu template, ta chỉ được phép hiệu chỉnh, nhập nội dung mới cho các vùng không khoá. Khi có sự thay đổi trong trang mẫu template thì các vùng khoá của các trang sử dụng template cũng sẽ tự động cập nhật theo.

**XI.4.1. Tạo mới một Template:****1) Tạo trang template:**

- Tạo mới một trang HTML template (mẫu) như một trang bình thường Kê Layout table, Layout cell phù hợp, nhập nội dung, chèn hình cho các vùng dùng chung
- Lưu trang mẫu template: **File → Save as template**
- Khi lưu trang dưới dạng template (.dwt) mặc định tất cả các vùng của trang template đều bị khoá, do đó phải mở khoá cho các vùng không dùng chung
  - Chọn vùng cần mở khóa
  - **Insert → template objects → Editable Region →** đặt tên cho vùng mở khóa



- ❖ Các cách khác để tạo Template:
  - Tạo template theo mẫu có sẵn:
  - Chọn thực đơn File/New.....
  - Chọn Page Designs/Text: Article D with Navigation.
  - Chọn Creat template ở cuối phải hộp thoại.



## 2) Tạo trang theo mẫu template:

- Chọn File/New...
- Trong hộp thoại New Document, chọn tab template
- Chọn mẫu template đã tạo sẵn → create.

Những vùng dùng chung sẽ bị khóa, khi thiết kế người dùng chỉ có thể thay đổi nội dung trong phần đã được mở khóa

## XI.4.2. Hiệu chỉnh Template:

### 1. Hiệu chỉnh template

- Mở template cần hiệu chỉnh:
- Modify/ template / Open Attached template.
- Xuất hiện trang mẫu template, thực hiện hiệu chỉnh

### 2. Đổi tên template:

- Trong Asset Panel, nhóm template
- Chọn template cần đổi tên .

### 3. Xoá một template:

- Trong Asset panel, chọn nhóm template.
- Chọn template cần xóa
- Nhấn delete .

Khi xoá một template sẽ ảnh hưởng đến tất cả các trang có sử dụng template. Nếu thực sự muốn xoá, trước tiên nên mở các trang sử dụng template rồi chọn chức năng tách khỏi template

#### 4. Tách trang khỏi template

- Modify/ Template/ Detach from template.

#### 5. Sử dụng Template cho trang:

Sau khi tạo được các trang mẫu template, ta có thể dễ dàng sử dụng chúng. Áp dụng một mẫu template:

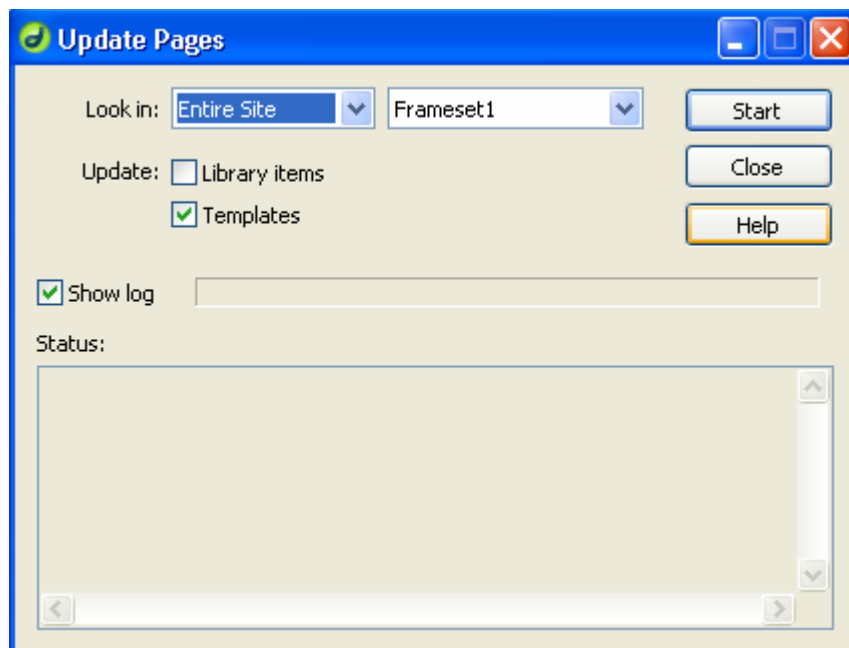
- File/ New/ HTML tạo trang mới.
- Modify/ Template/ Apply template to page...
- Chọn mẫu template.
- Nhập nội dung, hình ảnh vào những vùng không khoá.

#### ❖ Hoặc thực hiện cách khác:

- Mở Asset Panel, nhóm template.
- Chọn trong danh sách các mẫu template.
- Chọn nút Apply.

#### 6. Cập nhật trang sử dụng template:

- Modify/ Template/ Update current page, cập nhật trang hiện hành.
- Modify/ Template/ Update page/ Entire site trong list box look in.



## CHƯƠNG XII: BEHAVIORS - FORM

### XII.1. BEHAVIORS

#### XII.1.1. Tổng quan về Behaviors

Behaviors là các đoạn kịch bản (Script) được thiết kế sẵn bằng chương trình Dreamweaver, nó giúp thêm các điều khiển, hàm kiểm tra trình duyệt, Shockwave, thêm hệ thống liên kết Popup\_Menu, chèn âm thanh, kiểm tra form, làm phong phú hơn trang Web của bạn. Mỗi Behaviors gồm 3 yếu tố:

- Đối tượng chứa Behaviors:  
Đối tượng chứa biến cố rất đa dạng, có thể là trang, form, ảnh, nút, một dòng text... Phải chọn đối tượng phù hợp với biến cố và hành động.
- Biến cố (Event) xảy ra trên đối tượng: Là các sự kiện xảy ra trên đối tượng được chọn, có rất nhiều loại biến cố, các biến cố khác nhau trên các đối tượng khác nhau trên cùng một trang Web.
- Hành động đi kèm với biến cố: Là một đoạn mã lệnh thực hiện nhiệm vụ cho đối tượng. Hành động chỉ được gọi khi có biến cố tương ứng, điều đó nói lên mối quan hệ chặt chẽ giữa đối tượng, biến cố và hành động.

Một đối tượng trong trang thường đi kèm với một Behaviors, nhưng cũng có trường hợp một đối tượng có nhiều hơn một biến cố, khi đó tùy thuộc vào trình tự các Behaviors mà chương trình lần lượt kiểm tra từng biến cố.

Nếu một đối tượng có nhiều Behaviors mà các Behaviors này lại có cùng loại biến cố (nhưng khác hành động) thì khi biến cố xảy ra, lập tức các hành động tuân tự thực hiện

#### XII.1.2. Behaviors Panel

- Mở Behaviors Panel: Chọn **Window** → **Behaviors** → Xuất hiện Behaviors Panel
- Chọn đối tượng gắn Behaviors  
Tùy thuộc vào việc chọn các đối tượng khác nhau mà Behaviors tự chọn loại biến cố phù hợp và cho phép thực hiện một số hành động tương ứng trên đối tượng đó, bạn có thể điều chỉnh lại, do đó việc chọn đối tượng là công việc đầu tiên và rất quan trọng trong việc tạo một Behavior
- Đối tượng được chọn sẽ hiển thị trên Behaviors Panel dưới dạng <Tag> Actions, tên thẻ HTML kế bên từ Actions

##### 1. Thêm, xoá một Behaviors


- ❖ Thêm Behaviors:
  - Click nút (+) chọn hành động trong danh sách
  - Một biến cố tương ứng xuất hiện (có thể hiệu chỉnh lại)
- ❖ Xoá Behaviors:
  - Chọn dòng Behaviors cần xoá trong danh sách
  - Click nút (-)

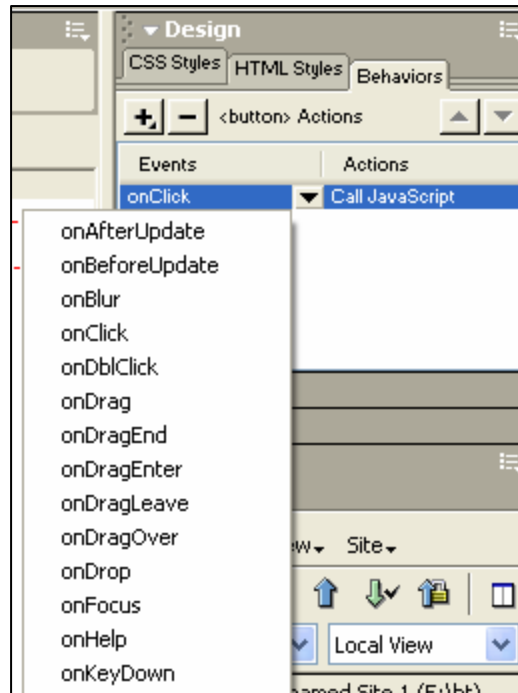
##### 2. Thay đổi trình tự Behaviors

Tùy thuộc vào trình tự các Behaviors mà các hành động theo trình tự thực hiện. Có những hành động ngang cấp thì không cần trình tự.

- ❖ Các Behaviors khác biến cố  
Ví dụ: một ảnh <img> có 2 biến cố khác nhau onMouseOut và onMouseOver, không cần quan tâm đến trình tự
- ❖ Các Behaviors có cùng biến cố:

Trong trường hợp này, phải sắp xếp đúng trình tự thì mới có kết quả như mong muốn. Cách sắp xếp:


- Chọn Behaviors cần sắp xếp
- Click nút  để sắp xếp



### 3. Thay đổi biến cố:

Khi hành động được chọn thì biến cố tương ứng cũng được gán cho Behavior.

Khi biến cố không phù hợp, ta có thể thay đổi biến cố khác bằng cách:

- Chọn dòng Behaviors chứa biến cố cần thay đổi,
- Click chuột vào mũi tên  trên dòng Behaviors đó

### XII.1.3. Tên và ý nghĩa các biến cố

- OnAbort: Khi ngưng tải một ảnh, hoặc nhấn nút Stop trong trình duyệt
- OnAfterUpdate: Khi trang thực hiện xong việc cập nhật nguồn dữ liệu
- OnBeforeUpdate: trước khi trang thực hiện cập nhật nguồn dữ liệu
- OnBlur: khi đối tượng được chọn không còn ở trạng thái hiện hành
- OnBounce: Khi nội dung trong Marquee đến biên của Marquee
- OnChange: Khi có sự thay đổi trong Textfield hay List/menu của form
- Onclick: Khi Click chuột vào đối tượng
- OnDbClick: Khi Double Click vào đối tượng
- OnError: Khi xảy ra lỗi hiện trang của trình duyệt
- OnFinish: khi nội dung Marquee đi hết một vòng
- OnFocus: Khi nhập nội dung vào TextField
- OnKeyDown: Khi nhấn phím xuống
- OnKeyUp: Khi thả phím ra
- OnKeyPress: Khi nhấn phím rồi thả ra
- OnLoad: Khi hoàn tất việc tải trang mới
- OnMouseDown: Khi nhấn chuột xuống
- OnMouseMove: Khi di chuyển chuột

- OnMouseOut: Khi di chuyển chuột ra ngoài nút
- OnMouseOver: Khi di chuyển ngang qua nút
- OnMouseUp: Khi thả chuột ra
- OnMove: khi cửa sổ hoặc khung di chuyển
- OnReadyStateChange: Trạng thái thành phần thay đổi, trạng thái gồm: Uninitialized, loading, complete
- Onreset: Khi Form trả về giá trị mặc định
- OnResize: khi cửa sổ hoặc khung thay đổi kích thước
- OnRowEnter: Khi nguồn dữ liệu có thêm dòng mới
- OnRowExit: khi dòng dữ liệu đã tồn tại
- OnScroll: Khi cuộn thanh cuộn
- OnSelect: Khi chọn Text, MenuItem của List/menu
- OnStart: Khi nội dung Marquee bắt đầu 1 vòng
- OnSubmit: Khi gửi Form
- OnUnload: Khi rời khỏi trang

### **XII.1.4. Hiệu chỉnh Behaviors**

#### **1. Cập nhật behaviour:**

Một Behavior sau khi được tạo sẽ xuất hiện trong danh sách các Behavior, ta có thể cập nhật, hoặc thay đổi các thông tin của Behavior bằng cách:

- Chọn đối tượng có Behavior
- Mở Behavior Panel
- Double Click trên Behaviour muốn cập nhật
- Thực hiện việc chỉnh sửa

#### **2. Thay đổi thuộc tính của behaviours:**

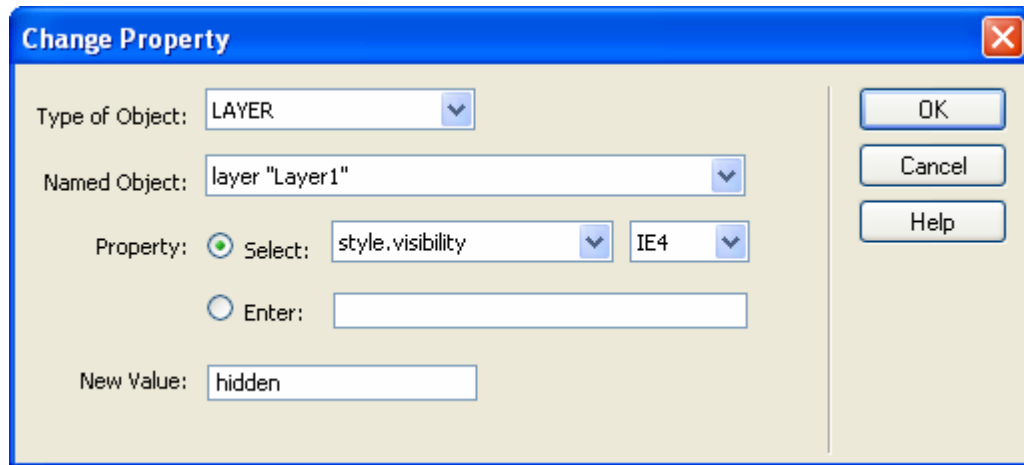
Thay đổi thuộc tính của đối tượng khi xem trang là một việc làm thú vị, thao tác rất đơn giản nhưng có thể tạo ra hiệu ứng lạ như sự ẩn hiện hoặc chuyển động của đối tượng

Ví dụ: Nút tạo sự ẩn hiện của đối tượng

Khi có một số lớp ảnh xếp chồng lên nhau, bạn có thể thay đổi thuộc tính ẩn hiện của chúng.

*Cách tạo:*

- Tạo một trang mới
  - o Chèn Layer 1 chứa ảnh thứ nhất, nhập Layer ID: Layer 1
  - o Chèn Layer 2 chứa ảnh thứ hai, hiệu chỉnh sau cho cùng tọa độ với Layer 1
- Tạo Behavior điều khiển:
  - o Chọn nút điều khiển ở góc trên màn hình
  - o Gán Behavior: chọn change properties
  - o Xuất hiện hộp thoại Change Properties: Trong danh sách Type of Object: chọn Layer
    - Name Object: nhập layer1
    - Property: Chọn Select: style.Visibility
    - New Value: Hidden → OK đóng hộp thoại
    - Trong Behavior Panel chọn tên biến cố là OnMouseOver
- Tạo thêm một Behavior cho nút này, trong hộp thoại Change Properties, nhập các thông số tương tự, nhưng tại ô New value: nhập visible
  - o Trong Behavior, thay tên biến cố là onMouseOut



**Ví dụ** : Tạo nút điều khiển tọa độ đối tượng. Bạn có thể thay đổi tọa độ của Banner về trái hay phải tùy thuộc vào click mũi tên theo hướng nào

*Cách tạo:*

- Tạo trang mới
- Chèn một Layer, chèn ảnh banner vào Layer
- Chọn Layer, thay đổi các thuộc tính như sau:
- LayerID: LayerLogo, L=0, T=0
- Chèn nút mũi tên điều khiển tọa độ ảnh
- Kéo 2 nút mũi tên ra giữa trang
- Chọn nút mũi tên trái, chèn Behavior, chọn Change Properties, nhập thông số:
- Properties: Style.left
- New Value=0 → OK
- Trong Behavior, chọn biến cố onClick.
- Làm tương tự cho mũi tên phải
- Properties= Style.left
- New Value=270 → OK
- Trong Behavior, chọn biến cố onClick.
- Check Browse- Check plugin
- Check Browse
- Trong site có thể chèn vào một số hiệu ứng đặc biệt không phải trình duyệt nào cũng xem được

**Ví dụ**

Bạn tạo 2 Website, Một Site mới nhất chứa các hiệu ứng đặc biệt, có trang chủ là tập tin có tên SiteMoi.htm, Một site thiết kế theo kiểu cũ, có trang chủ là tập tin có tên Sitecu.htm

Dùng Behavior để kiểm tra trình duyệt của người xem, nếu có trình duyệt mới thì xem Site mới bắt đầu với trang SiteMoi.htm, ngược lại thì xem Site cũ với trang SiteCu.htm

*Cách thực hiện:*

- Tạo trang mới (không cần nhập nội dung trang)
- Chọn thẻ <Body>
- Trong Behavior Panel, Click (+) chọn Check Browser
- Nhập URL: SiteMoi.htm
- Nhập Alt URL: SiteCu.htm → OK
- Một biến cố Onload kèm hành động Check Browser. Lưu tập tin → F12 xem kết quả



### XII.1.5. Check Plugin

Khi một trang chưa có thành phần Flash hay Shockwave thì người xem cần có thêm Plugin để xem được trang, mà bạn không biết người xem có cài các Plugin để xem hay không, khi đó bạn có thể dự phòng thêm một trang thường không sử dụng Flash, Shockwave

- Nếu người xem có Plugin thì cho xem trang “TrangFlash.htm”, ngược lại thì xem “trangthường.htm”. Cách thực hiện:
- Định hướng Site:
- Tạo trang mới ( Không cần nội dung)
- Chọn Tag <Body>, Click nút (+), chọn Check Plugin
- Select: Shockwave
- If found, Go to URL: TrangFlash.htm
- Otherwise, go to URL: Trangthuong.htm

### XII.1.6. Định hướng Page:

- Tương tự, chọn nút liên kết <Image>
- Trong Behavior, chọn biến cố Onclick, action Check Plugin

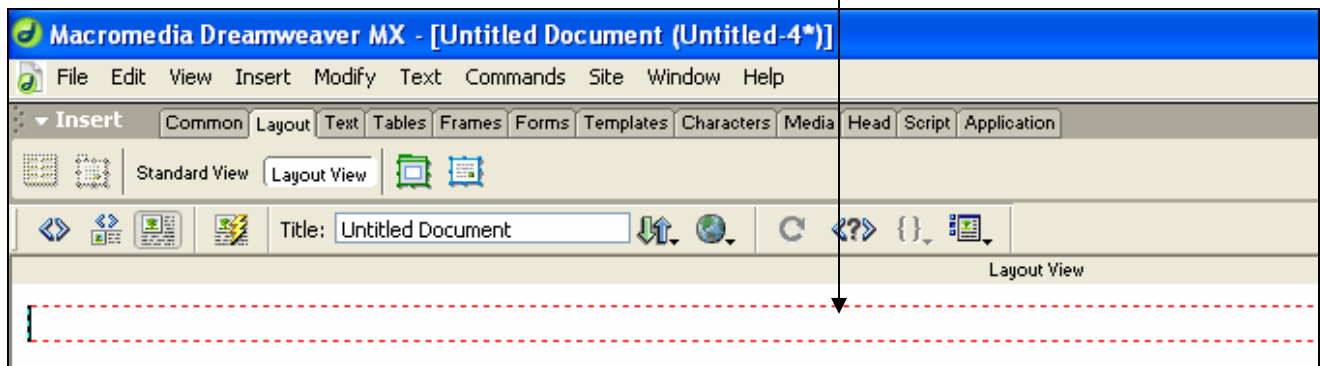
## XII.2. FORM:

Form để giao tiếp giữa người tham quan và người chủ web, có thể liên lạc thông qua Email. Form còn là màn hình nhập liệu, tập hợp những ô nhập liệu, nút chọn, nút kiểm tra hay dạng liệt kê danh sách... Đây là cách giao tiếp giữa người tham quan với chủ trang web. Dreamweaver giúp tạo các biểu mẫu rất dễ dàng và nhanh chóng. Các ô nhập liệu, các nút điều khiển...Ta có thể thay đổi các mặc định của form và xử lý thêm cho phù hợp với ý mình.

### XII.2.1. TẠO FORM:

- Chọn menu **Inert/ Form.**

Phạm vi của form



- Đường khuôn viền không liền nét màu đỏ đại diện cho giới hạn trong form. Mọi thành phần của form đều phải nằm trong khuôn viền này.
- Trong form cần thiết phải có 2 nút đặc biệt “SUBMIT” và “RESET”
  - Submit: chấp nhận nội dung của form, và cho phép người duyệt gửi thông tin, dữ liệu trong form về server
  - Reset: Thiết lập giá trị ban đầu của tất cả các điều khiển trên form

## XII.2.2. Các đối tượng trong form:

Có 2 cách để chèn các đối tượng vào form

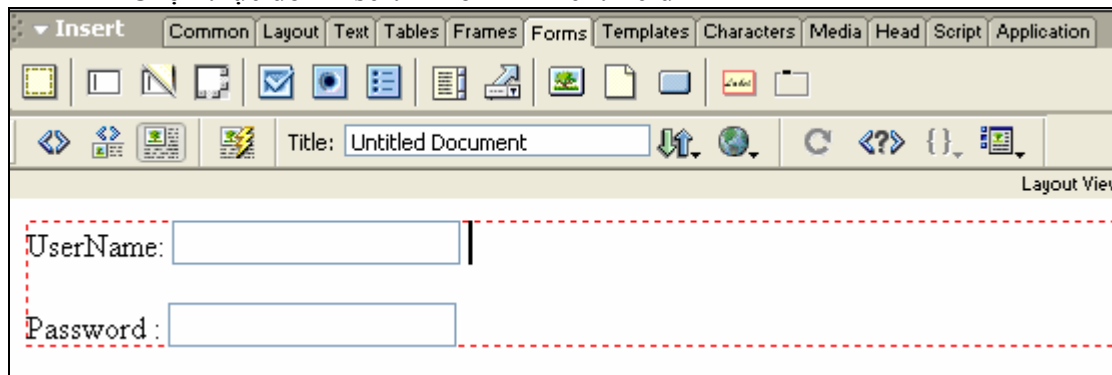
- Cách 1: Chọn tab form trên **Insert Panel**.
- Cách 2: Chọn menu **Insert** → **Form** → **Chọn đối tượng**

### 1. TEXT FIELD:

#### a) Cách tạo:

Dùng nhập các thông tin ngắn nằm gọn trong một dòng, loại ô này thường dùng nhập các thông tin về họ tên, địa chỉ, số điện thoại...

- Đặt con trỏ trong đường viền form.
- Chọn thực đơn **Insert** → **Form** → **Text field**



#### b) Thuộc tính của Textfield :

- Chart Width : Số ký tự cho chiều dài ô.
- Max Chars : Số ký tự nhiều nhất có thể nhận.
- Init Value : Nội dung khởi tạo của ô nhập liệu.
- Single Line : Ô nhập liệu một dòng.
- Multi line : Ô nhập liệu nhiều dòng.
- Password : Ô nhập các loại mật mã, khi nhập liệu từng ký tự nhập được thay bằng dấu “ \* ”.

### 2. RADIO BUTTON :

Trong 1 nhóm các nút chọn chỉ được chọn 1 tùy chọn. Các nút radio thuộc cùng nhóm phải có cùng tên nhóm (Group), nhưng khác nhau về giá trị(value).

#### a) Cách tạo:

- Đặt dấu nháy tại vị trí muốn chèn
- Chọn **insert** → **Form** → **radio Button**

Giới tính: Nam , Nữ

#### b) Thuộc tính của Radio button:

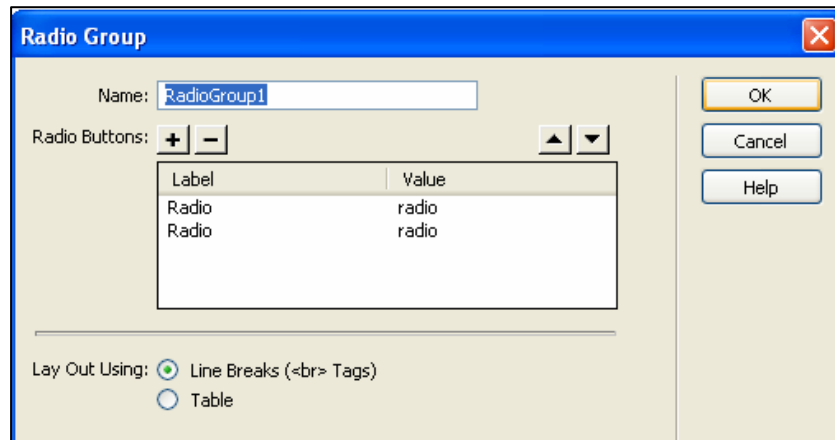
- Checked value : Giá trị từng nút đơn trong nhóm.
- Initial State : trạng thái ban đầu của nút.
- Checked: Nút được chọn.
- Unchecked: Nút chưa được chọn.

### 3. RADIO GROUP: :

Tạo một nhóm nút radio.

#### a) Cách tạo:

- Chọn **Insert** → **Form** → **Radio group**. Hộp thoại Radio Group xuất hiện.



### b) Thuộc tính của Radio group:

- Name: Nhập tên nhóm
- dấu "+": thêm nút, dấu "-": xoá nút.
- Label là tên nhãn nút.
- Value là giá trị gửi về cho chủ web.
- Layout Using: Bộ cục đang sử dụng
- Line break: Các nút xuống dòng
- Table: Các nút nằm trong một bảng.

## 4. CHECKBOX:

Tạo một nhóm các tùy chọn, cho phép chọn nhiều tùy chọn cùng một lúc hoặc không chọn gì cả.

**Cách tạo:** Chọn **Insert** → **Form Objects** → **Checkbox**.

## 5. LIST/ MENU (Dropdown menu)

Hộp liệt kê danh sách các nội dung theo từng dòng, được cuộn lại thành 1 dòng. Khi cần xem danh sách, Click nút mũi tên chỉ xuống bên phải hộp liệt kê để thả danh sách xuống.

- Chọn menu Insert/ Form object/ List/ Menu.
- Trong properties inspector ta nhập vào tên tùy ý trong list/menu.
- Chọn kiểu type: Menu.
- Ấn nút List Values...
- Ấn "+" để thêm mục, "-" để xoá mục chọn.
- Ấn nút OK, đóng hộp thoại List. Value
- Chuyển qua dạng xem Preview in browser, hay F12 xem kết quả.

## 6. TEXTAREA:

TextArea dùng nhập các thông tin, dữ liệu dài gồm nhiều dòng, thường dùng cho các ghi chú, ý kiến, cảm nghĩ, thắc mắc... của người duyệt web.

- Chọn menu Insert → Form object → Textarea.
- TextArea Properties:
  - o Text field : tên ô nhập liệu nhiều dòng.
  - o Char Width : Số ký tự cho chiều dài ô.
  - o Num lines : Số dòng cần hiển thị.
  - o Wrap : Văn bản tự xuống dòng.
  - o Init Value : Giá trị khởi tạo của ô

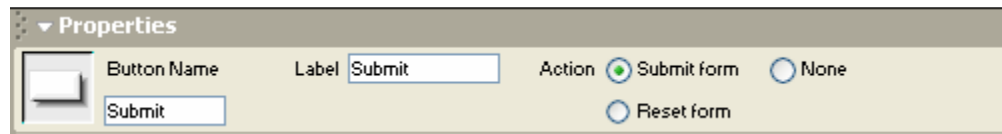
## 7. FILE FIELD: Gửi kèm tập tin:

Trong form, ngoài các thông tin ngắn gọn mà người tham quan có thể nhập trực tiếp, nếu họ muốn gửi một đoạn văn bản nhiều dòng, hay một hình ảnh minh họa thì ta nên chèn thêm file field. Đây là thành phần form giúp gửi kèm một tập tin theo các thông tin nhập.

- Chọn menu Insert → Form objects → File field
- Ấn F12 xem kết quả: Nếu muốn gửi kèm ảnh thì ấn nút Browser rồi tìm tập tin ảnh cần gửi hay cũng có thể nhập đường dẫn thư mục chứa hình ảnh.

## 8. BUTTON:

- Chọn menu **Insert** → **Form object** → **Button**.
- Các thuộc tính của Button:



- Button name. Đặt tên cho submit
- Gán nhãn cho button.
- Action:
  - Submit form.: nút submit
  - Reset form: nút reset
  - None: button do người sử dụng gán hành động

## 9. NÚT HÌNH ẢNH::

Chèn hình ảnh vào trang: **Insert** → **Image**... hiệu chỉnh thuộc tính để hình ảnh này thành các nút như Submit, Reset....

### a. Cách tạo:

- Định vị con trỏ nơi cần chèn hình.
- Insert → Form object → Images field. (hoặc click nút Image field trên thanh công cụ)

### b. Image field properties:

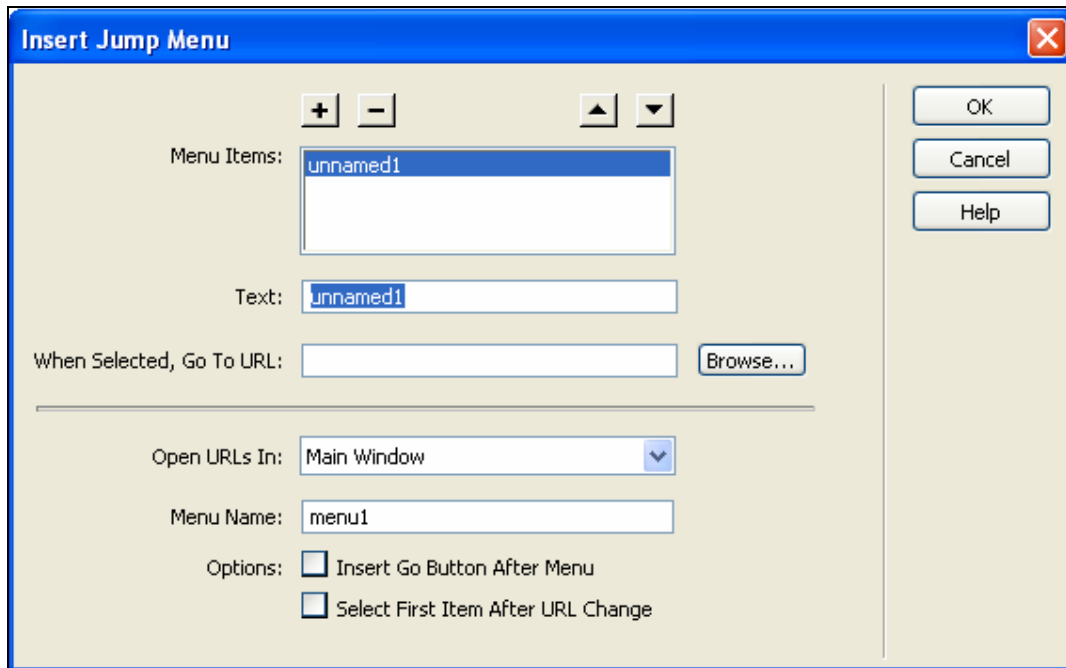
- ImageField: “Submit” chọn ảnh làm nút submit  
“Reset” chọn ảnh làm nút reset.
- W (Width): chiều rộng ảnh làm nút.
- H (Height): chiều cao ảnh làm nút.
- Src: tên tập tin ảnh làm nút.
- Alt: câu ghi chú khi chuột ngang qua ảnh.
- Align: canh vị trí ảnh so với các đối tượng khác.
- Edit Image: gọi chương trình Fire Works xử lý ảnh.

## 10. JUM MENU :

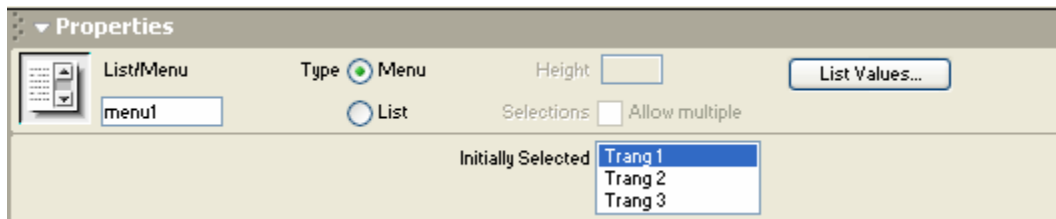
Đó là cách xếp gọn danh sách các liên kết lại trên một dòng, khi cần chọn liên kết nào, người tham quan thả danh sách liên kết xuống, rồi chọn trong số đó. Cách này giúp những trang web có quá nhiều liên kết có thể thu hẹp diện tích, tạo thông thoáng cho web hơn. Đôi khi việc chọn liên kết này rất nhay, nó sẽ nhảy nhanh đến các trang liên kết, để khắc phục, ta có thể chèn thêm nút GO kế bên. Chọn liên kết trong danh sách xong phải nhấn nút GO để xác nhận.

**a. Cách tạo:**

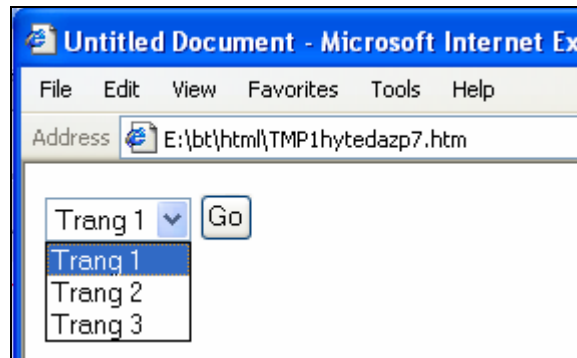
- Định vị con trỏ.
- Chọn thực đơn Insert → Form Objects → Jump Menu.
- Trong hộp thoại insert Jump Menu
  - o Text: Đặt tên cho jum menu
  - o When selected, Go To URL : Nhập tên trang liên kết
  - o Kết quả sẽ xuất hiện trên dòng Menu Items.
  - o Dấu “+” hoặc “-” để thêm hoặc xoá một Menu Item.
  - o Mũi tên lên hoặc xuống để thay đổi trình tự các Menu Items.
  - o Menuitem: danh sách các menuitem được nhập trong text.
  - o Open URL In: Target của khung, hay cửa sổ chính hiển thị trang.
  - o Menu name: Tên của Menu.
  - o Select Frist Item After URL Change: Chọn Item đầu tiên sau khi liên kết.
  - o Insert Go Button After Menu: chèn thêm nút GO.



Ví dụ: màn hình thiết kế: thuộc tính của jum menu



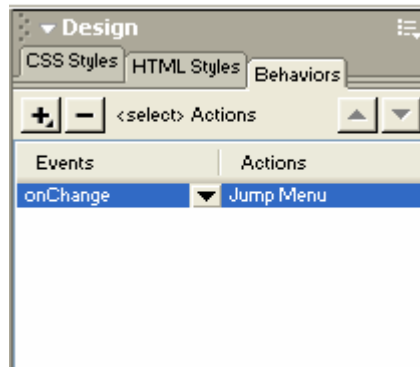
Kết quả trên trình duyệt:



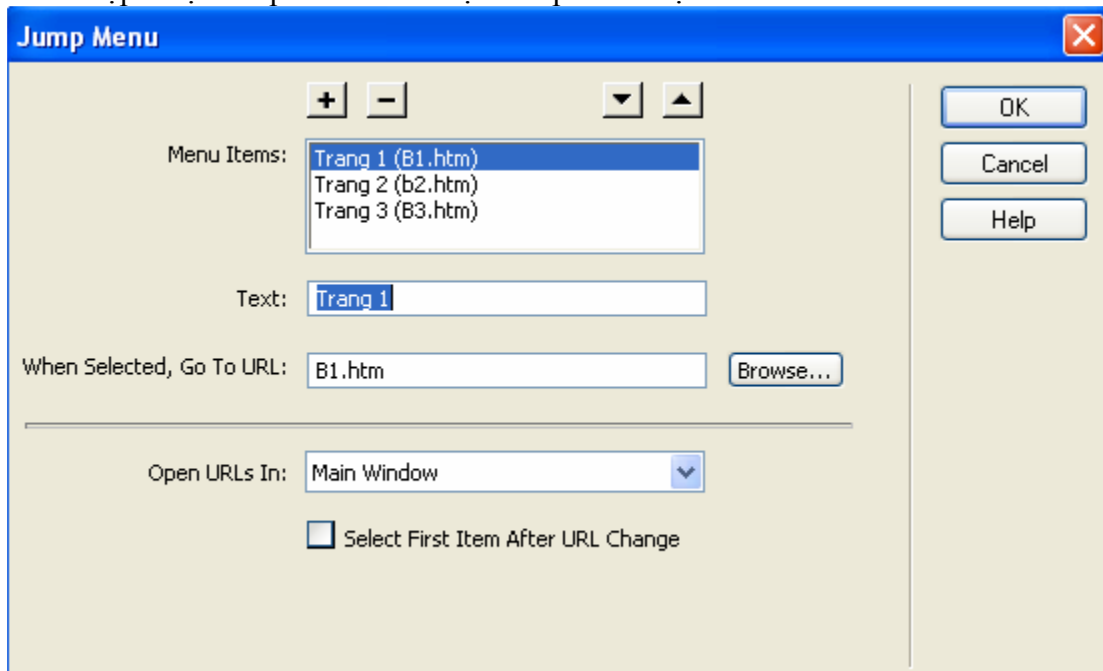
### b. Hiệu chỉnh Jump Menu:

Khác với các thành phần khác của form, khi cần hiệu chỉnh tên Menu Item, hay liên kết tương ứng, ta không xem ở Properties Inspector mà phải mở Behaviors.

- Chọn Jump Menu.
- Chọn menu Window → Behaviors → xuất hiện Design Panel của Behaviors.



- Double click vào tên Action: Jump Menu.
- Hộp thoại Jump menu xuất hiện cho phép ta hiệu chỉnh.



### c. Kiểm tra liên kết của Jump Menu:

Để kiểm tra liên kết của Jump Menu, ta cần xem trang trong trình duyệt, rồi ấn vào tên Menu Item trong Jump Menu để liên kết đi.

- File → Preview in browser, hoặc F12.
- Click vào dòng Menu Item, tên trang liên kết đến.

### XII.2.3. Sử dụng Behavior cho Textfield:

Textfield là ô nhập liệu của người tham quan, nhưng có trường hợp họ không nhập liệu cho ô, hoặc nhập giá trị không phù hợp, behavior của textfield có thể giúp kiểm tra các trường hợp này.

Ví dụ:

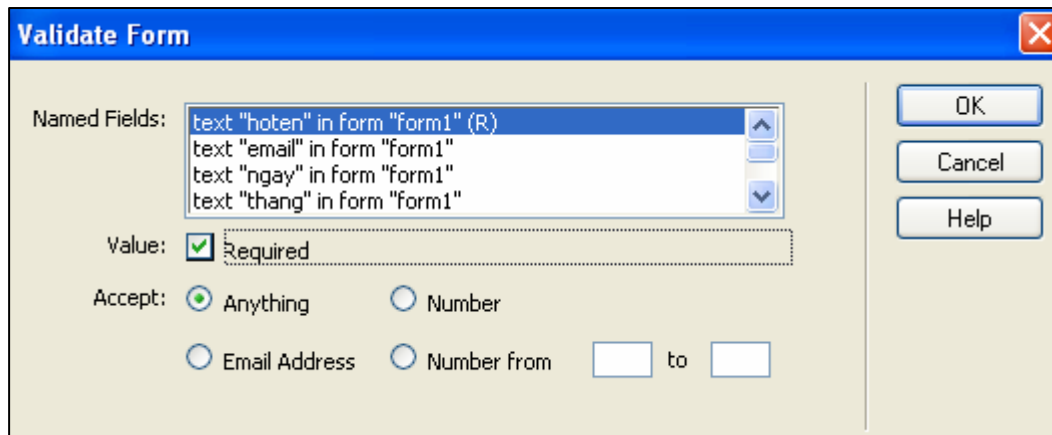
Ta có một form đặt phòng khách sạn, để dễ quản lý, ta cần thiết kế sao cho việc nhận thông tin là đầy đủ và chính xác nhất.

ĐẶT PHÒNG KHÁCH SẠN	
Họ và tên :	<input type="text"/>
Email :	<input type="text"/>
Ngày đến :	Ngày <input type="text"/> Tháng <input type="text"/> năm <input type="text"/>
Số người	<input type="text"/>
Ghi chú :	<input type="text"/>
<input type="button" value="datphong"/> <input type="button" value="vietlai"/>	

- Yêu cầu: Ta cần kiểm tra việc nhập liệu của người tham quan, cụ thể là:
  - Phải nhập dữ liệu cho ô “Họ và tên”.
  - Ô email phải đúng là địa chỉ Email, nghĩa là phải có ký tự @ trong địa chỉ email.
  - Ô “Ngày” phải nhập từ số 1 đến 31.
  - Ô “Tháng” phải nhập từ số 1 đến 12.
  - Ô “Năm” phải nhập từ số 2002 đến 2010.
  - Ô, “Số người” phải nhập dữ liệu số.
  - Ô “Ghi chú” là ô thông tin thêm có thể bỏ qua không nhập liệu cũng được.
- Cách thực hiện: Để kiểm tra các điều kiện này, phải dùng Behavior của form.
  - Chọn form “ĐẶT PHÒNG KHÁCH SẠN”. Hay chọn nút Submit, Reset làm đối tượng kiểm tra.
  - Trong Behavior Panel, click dấu (+).
  - Chọn chức năng Validate Form. Xuất hiện hộp thoại validate form chứa các thông tin giúp điều khiển Text field của form
  - Name field: danh sách tất cả các text field có trong form đó. Nếu muốn đặt điều kiện cho text nào thì ấn chọn cho nó hiện hành.
  - Value required: yêu cầu ô text đó phải được nhập dữ liệu, nếu không sẽ báo lỗi.
    - Anything: dữ liệu loại nào cũng được.
    - Email address: dữ liệu phải có ký tự @.
    - Number: dữ liệu dạng số.
    - Number from...to : dữ liệu số trong khoảng xác định.
- Điều kiện nhập TextField:
  - Cách nhập điều kiện cho Textfield: Trong hộp thoại Behavior, click dấu (+), chọn Validate Form, xuất hiện hộp thoại Valildate Form:
    - Named Fields: Chọn tên field muốn gắn điều kiện

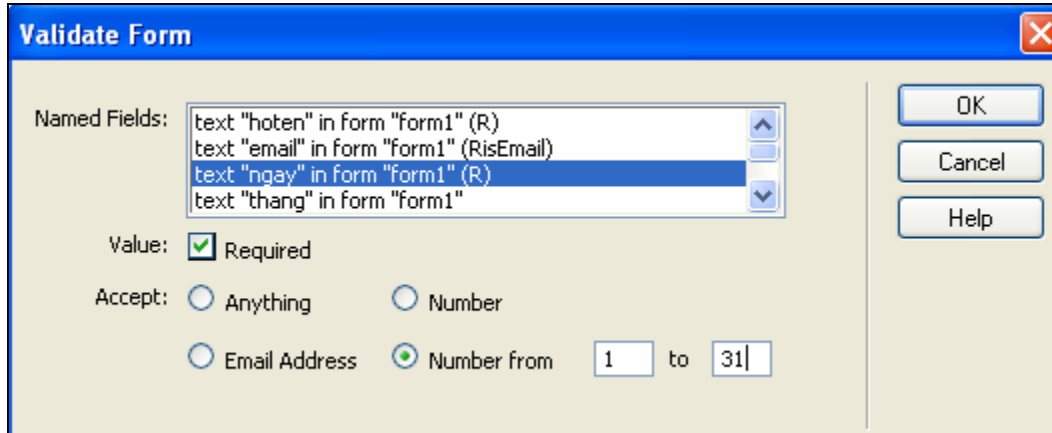
- Value: Nếu chọn Require thì yêu cầu phải nhập dữ liệu cho field đang chọn, nếu không sẽ báo lỗi
- Accept: Cho phép chọn kiểu dữ liệu, nếu nhập sai kiểu dữ liệu đã chọn thì xuất hiện thông báo lỗi
- Anything: mọi loại dữ liệu
- Number: dữ liệu kiểu số
- Email Address: kiểu địa chỉ Email
- Number Form: giá trị kiểu số nằm trong khoảng From ...to..

Ví dụ: điều kiện ô "Họ và tên", chọn Required



The screenshot shows the 'Validate Form' dialog box. The 'Named Fields' list contains four entries: 'text "hoten" in form "form1" (R)', 'text "email" in form "form1"', 'text "ngay" in form "form1"', and 'text "thang" in form "form1"'. The 'Value' section has 'Required' checked. The 'Accept' section has 'Anything' selected.

Điều kiện ô "Ngày": chọn Required, tương tự cho ô tháng, năm



The screenshot shows the 'Validate Form' dialog box. The 'Named Fields' list contains four entries: 'text "hoten" in form "form1" (R)', 'text "email" in form "form1" (RisEmail)', 'text "ngay" in form "form1" (R)', and 'text "thang" in form "form1"'. The 'Value' section has 'Required' checked. The 'Accept' section has 'Number from' selected with '1' in the 'From' field and '31' in the 'to' field.



## CHƯƠNG XIII: FRAMESETS – KIỂM TRA VÀ XUẤT BẢN

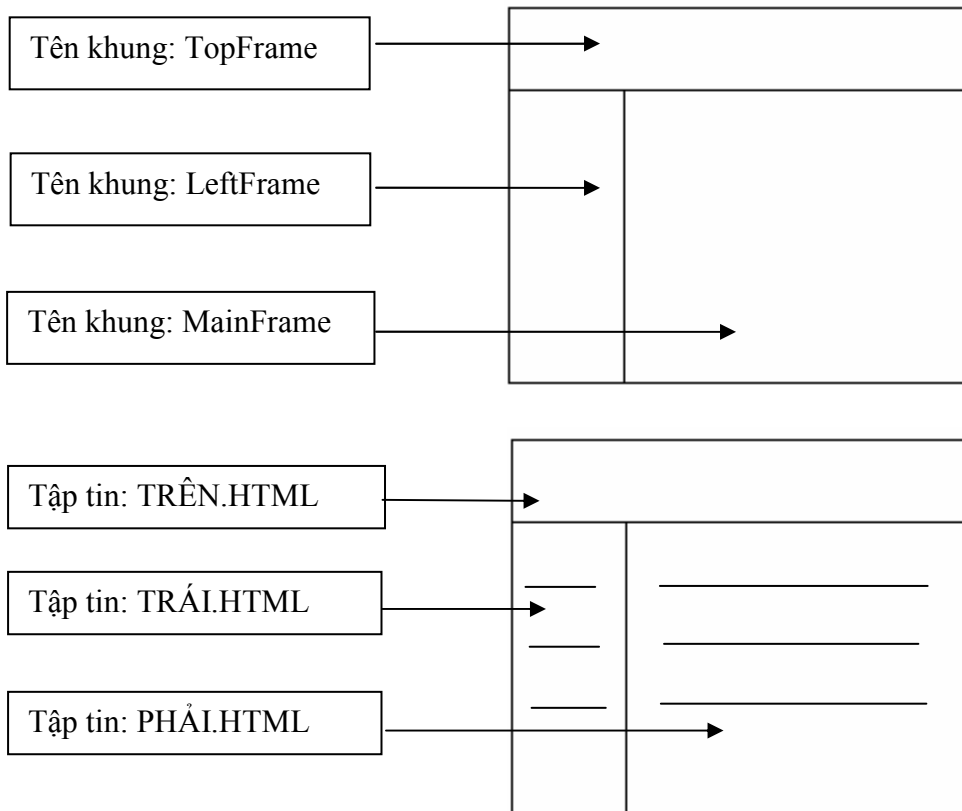
### XIII.1. GIỚI THIỆU

#### XIII.1.1. Trang khung (frameset):

Là trang HTML đặc biệt. Trang khung không mang nội dung, không có thẻ <BODY>. Trang khung chỉ giúp chia viền khung (frame) cho trang web. Muốn tạo trang khung ta phải xác định rõ các yêu cầu sau:

- Số khung (frame) trong một trang khung (frameset)
- Tên cho từng khung cụ thể.
- Các tập tin HTML làm nội dung cho từng khung.

Ví dụ:



tập tin: framepage.html

Trong ví dụ trên, tập tin trang khung có tên `framepage.html` được chia làm 3 khung: khung trên, khung trái và khung phải. Với tên cụ thể từng khung là:

- Khung trên : TopFrame.
- Khung trái : Leftframe.
- Khung phải : MainFrame.

Các tập tin HTML làm nội dung từng khung là:

- Tập tin TRÊN.HTML làm nội dung TopFrame.
- Tập tin TRÁI.HTML làm nội dung LeftFrame.
- Tập tin PHẢI.HTML làm nội dung MainFrame.

-

**❖ Ưu và khuyết điểm khi thiết kế một trang khung:****Ưu:**

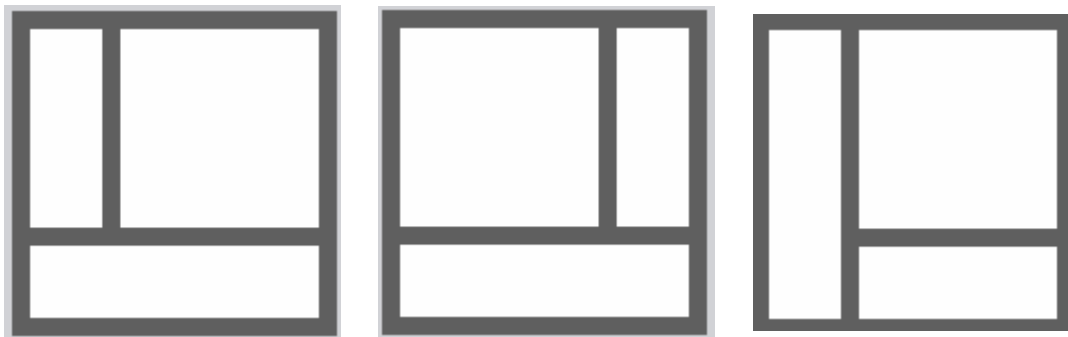
- Có thể hiển thị nhiều trang HTML cùng lúc trên màn hình duyệt.
- Phân chia tập tin rõ ràng cho từng khung cụ thể, nhóm tập tin chứa logo, banner riêng, nhóm tập tin chứa các nút liên kết riêng và nhóm tập tin chứa nội dung riêng.
- Phân nhóm nên dễ quản lý hệ nút liên kết, giúp điều khiển website thuận tiện hơn.
- Khi liên kết lật trang chỉ làm thay đổi nội dung trong một khung, các khung khác vẫn giữ nguyên, nên hạn chế được việc thiết kế các thành phần lặp lại như logo, banner, nút liên kết..., các thành phần dùng chung có thể để riêng trong khung chung.

**Khuyết:**

- Khá phức tạp trong thiết kế. Đòi hỏi người thiết kế phải hiểu cặn kẽ về khái niệm, thao tác thực hiện trên trang khung.
- Nếu không khéo thực hiện, các trang có thể bị bẻ gãy hoặc liên kết đến các trang không nằm đúng trong khung được chỉ định.

**Lưu ý trong thiết kế trang khung:**

- Phân biệt sự khác nhau giữa trang khung (Frameset) và khung (Frame), trang khung là tập hợp các khung đơn. Tập tin trang khung FramePage.html được chia làm 3 khung, khung trên (topframe), khung trái (leftframe) và khung chính (mainframe).
- Tên của từng khung như Topframe, Leftframe, Mainframe do Dreamweaver mặc định đặt tên, ta phải ghi nhận rõ từng tên khung để tiện thao tác liên kết trong trang khung, ta cũng có thể tự đặt tên cho trang khung.
- Phải phân biệt rõ sự khác biệt giữa tên khung và tên tập tin làm nội dung cho khung
- Vì các khung được chia ra từ trang khung nên kích thước từng khung bị hẹp, vùng hiển thị của từng khung nhỏ, nên khi thiết kế ta chỉ quan tâm những vùng được hiển thị như: Khung TRÊN, TRÁI, các tập tin TRÊN.HTML, TRÁI.HTML.
- Khi cần xem trang khung, phải mở tập tin khung framepage.html, nếu mở các trang trên.html, trái.html, phải.html riêng lẻ thì nội dung hiển thị không trọn vẹn.

**XIII.1.2. Các dạng trang khung và trình tự thiết kế:****1. Các dạng trang khung thường sử dụng:**



## 2. Trình tự thiết kế trang khung:

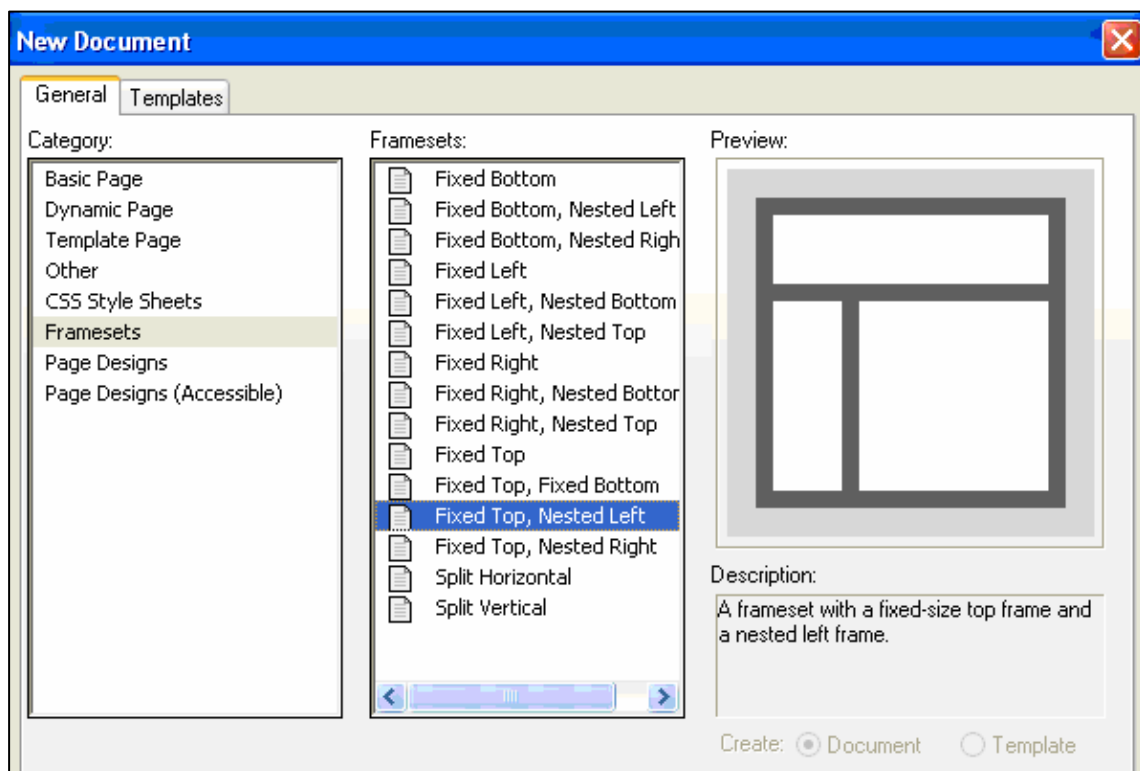
- Phác thảo bố cục dạng trang khung trên giấy nháp.
- Tạo website mới (site/ new site) chứa trang khung này.
- Phân nhóm tập tin nội dung và tạo nội dung từng nhóm.
- Chọn loại trang khung tương ứng và chỉ ra tập tin nội dung đại diện cho từng khung.
- Lưu trang khung.

## XIII.2. CÁCH TẠO TRANG KHUNG VÀ LIÊN KẾT TRANG

Dreamweaver tạo sẵn một số dạng trang khung chuẩn, ta có thể chọn và sử dụng chúng một cách dễ dàng. Nếu không có khung như ý thích, ta có thể chọn trang khung gần giống nhất rồi tự hiệu chỉnh hay thiết kế lại.

### XIII.2.1. Cách tạo:

- Chọn thực đơn File/ New...
- Chọn Frameset → Chọn dạng trang khung trong khung Framesets
- Xem mẫu trang khung bên cột Preview, click Create.



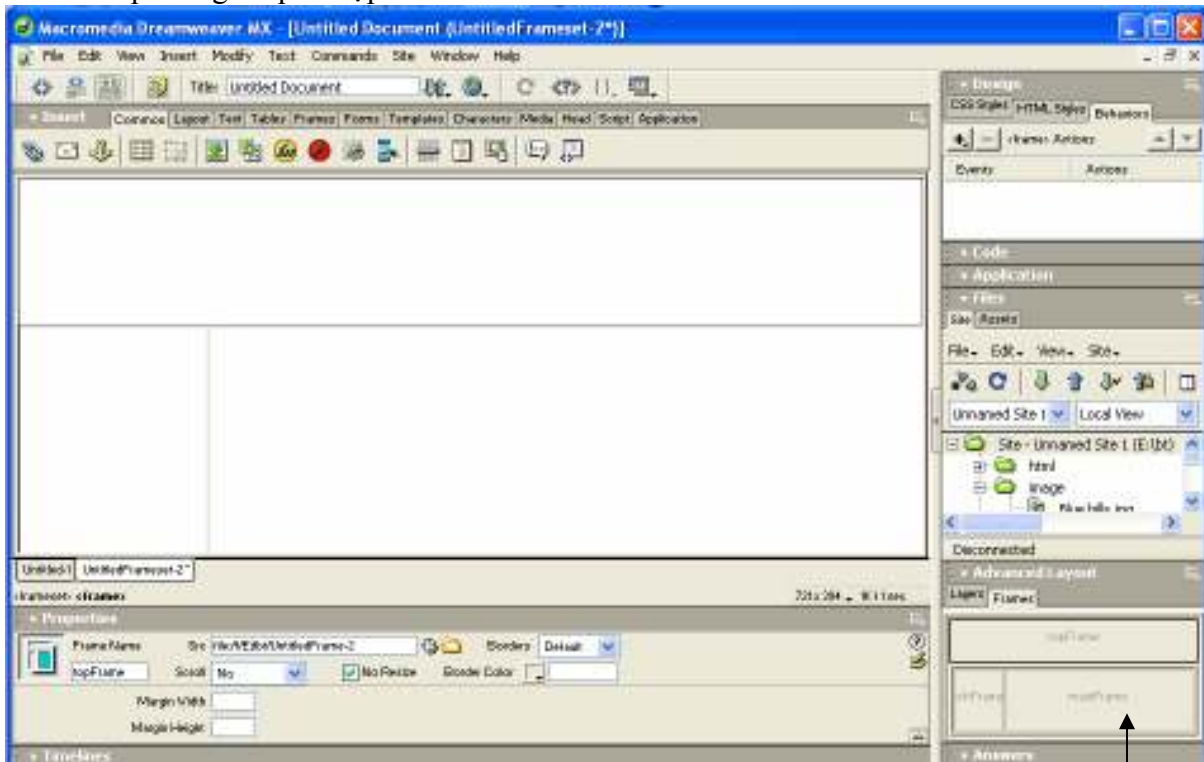
### XIII.2.2. Mở frames Panel:

Frames Panel giúp ta thao tác với từng khung một.

- Chọn thực đơn Windows → Others → Frames.
- Hoặc nhấn phím Shift + F2

### XIII.2.3. Thao tác trên Frames Panel:

Tuỳ thuộc vào thao tác trên Frames Panel mà Properties Inspector tương ứng cung cấp thông tin phù hợp



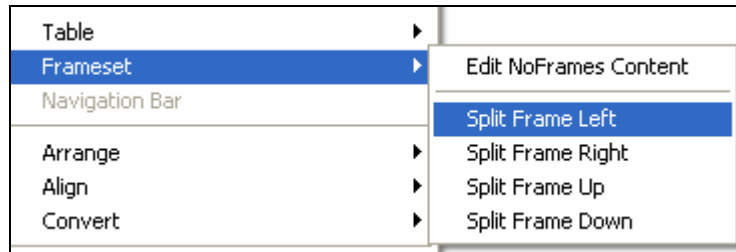
frames panel

- Click vào giữa khung, hay tên khung trong Frames panel để chọn khung cần làm việc, hiệu chỉnh như đổi tên khung, đổi nội dung tập tin khung, thanh cuộn, màu nền....
- Click đường viền ngoài cùng chọn cả trang khung, click vào đường viền giữa khung trên, khung dưới chọn dòng khung. Click đường viền giữa khung trái, phải chọn được cột khung. Sau thao tác chọn là thay đổi mối quan hệ giữa 2 khung, thường là kích thước, tỉ lệ khung, đường viền khung.

#### 1. Hiệu chỉnh trang khung:

##### a. Chia khung

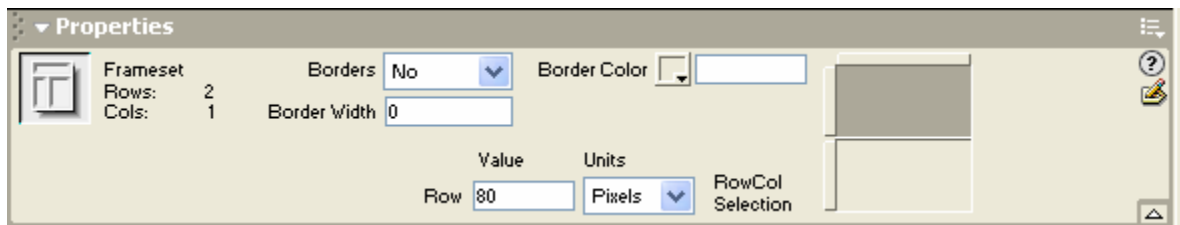
- Chọn tên khung trong Frame Panel cần chia.
- Chọn thực đơn Modify → Frameset.
- Chọn Split Frame Left/ Right / Up / Down để chia một khung thành 2 khung theo ý muốn.



b. Xoá khung:

- Đưa trỏ chuột đến biên khung cần xoá.
- Kéo biên khung đó ra khỏi màn hình, hoặc kéo sang hướng biên của khung cha.

c. Hiệu chỉnh thuộc tính trang khung:



- Thay đổi kích thước khung: Sau khi đặt nội dung vào khung, nếu khung không vừa với nội dung đã thiết kế, ta có thể thay đổi nhanh bằng cách
  - Đặt con trỏ chuột vào biên khung.
  - Giữ chuột và kéo biên đến vị trí mới.
- Thay đổi tên khung :
  - Chọn khung cần đổi tên.
  - Nhập tên mới trong ô Frame Name.
- Thay đổi nội dung đại diện trong khung:
  - Định con trỏ trong khung cần đổi nội dung.
  - Nhập tên tập tin .html mới vào Site Panel.
  - Hoặc ấn nút Browse to file... để tìm tập tin thay thế.
- Thay đổi biên cho khung:
  - Margin Width.
  - Margin hight
- Thay đổi kích thước khi xem:
  - Đánh dấu vào ô No Resize, không cho phép thay đổi kích thước trước khi duyệt trang khung.
- Không hiển thị thanh cuộn: Tại mục Scroll:
  - Chọn Yes: luôn luôn hiện thanh cuộn.
  - Chọn No: không hiển thị thanh cuộn, dù trang nội dung dài nhiều dòng.
  - Chọn Auto: Thanh cuộn tự xuất hiện khi nội dung dài hơn trang.
  - Chọn Default: Tùy thuộc vào cài đặt mặc định.
- Không viền nét đường khung:
  - Border = No không có đường viền
  - Border = yes: có đường viền
  - Width : Chọn kích thước nét viền
- Tô màu viền khung: Border Color

## 2. Lưu trang khung:

- a) Lưu cả khung trang:
  - Chọn viền ngoài cùng trang khung trong Frame Panel.
  - Chọn thực đơn File → Save Frameset As... để đặt tên trang khung.
  - Chọn File → Save Frameset để lưu cập nhật thông tin.
- b) Lưu trang đại diện khung :
  - Định con trỏ vào khung cần lưu.
  - Chọn thực đơn File → Save Frameset As....
  - Nếu chỉ cần cập nhật thông tin nội dung trong khung thì chọn File → save Frameset.

### XIII.3. KIỂM TRA VÀ XUẤT BẢN

#### XIII.3.1. Kiểm tra:

Trước khi xuất bản Website lên Server, cần phải thông qua một số thao tác kiểm tra kết quả của từng trang trong cả Website, bạn phải chắc chắn rằng phần trình bày bố cục, liên kết trang đã đáp ứng yêu cầu khi xuất hiện trên trình duyệt. Ngoài các kiểm tra thông thường như nội dung, hình ảnh, liên kết, các thẻ lệnh trên Site cục bộ, cần phải kiểm tra thêm một số các yếu tố sau đây:

**Trình duyệt Browse:** Kiểm tra các chức năng của Site phù hợp nhất với trình duyệt nào.  
**Màn hình:** Bố cục có thay đổi trên màn hình có kích thước 1024x768 pixel so với màn hình 800x600, phải cố gắng trung hoà trong trình bày, thiết kế ảnh, chọn kiểu font, với kích cỡ phù hợp.

**Liên kết Link:** không thể để trong Site những liên kết gãy, liên kết không đúng trang đích hoặc những trang mô côi không liên kết

**Thời gian tải trang:** là khoảng thời gian người xem chờ trang hiển thị, không nên đưa quá nhiều hình ảnh vào trang, có thể tạo trang Album.

**Tạo một Site thử nghiệm:** Nên chép Site đến một thư mục khác, hoặc một máy cục bộ khác để chạy thử nghiệm, giúp bạn hiệu chỉnh chính xác hơn.

Để giúp bạn dễ dàng hơn trong kiểm tra, Dreamweaver cung cấp lập bản báo cáo kết quả từng trang, cả Site rất chi tiết, dựa vào đó bạn có thể quản lý và hiệu chỉnh một cách hiệu quả nhất. Cách thực hiện như sau:

- Chọn **File → Check Page → Check Link**
- Xuất hiện cửa sổ Result Inspector . Với nhóm Search, Validation...
  - Nhóm Search:
 

Khi cần hiệu chỉnh một số thông tin như: tên công ty, số điện thoại, ..., nằm rải rác ở các trang khác nhau, để không sai sót trong quá trình tìm kiếm và thay thế, ta chọn nhóm Search: **File → Check Page → Check Link**

Trong nhóm Search, Click nút mũi tên, Chọn Find and Replace

    - Find What: Nhập nội dung cần tìm
    - Replace With: nhập nội dung thay thế, chọn Replace All
  - Nhóm kiểm tra tính hợp lệ (Validation):
 

Nếu thiết kế trang Web bằng các thẻ lệnh HTML thì không tránh khỏi những sai sót nhỏ, ta có thể mở trang và kiểm tra tính hợp lệ của các thẻ HTML

    - Click nút mũi tên, chọn validation Current Document
    - Xuất hiện hộp thoại chỉ rõ các thẻ không hợp lệ
  - Kiểm tra tính tương thích trên trình duyệt
 

Có những thẻ lệnh mà trình duyệt cũ không nhận diện được, để kiểm tra tính tương thích trình duyệt, chọn nhóm Check Target Browser Check

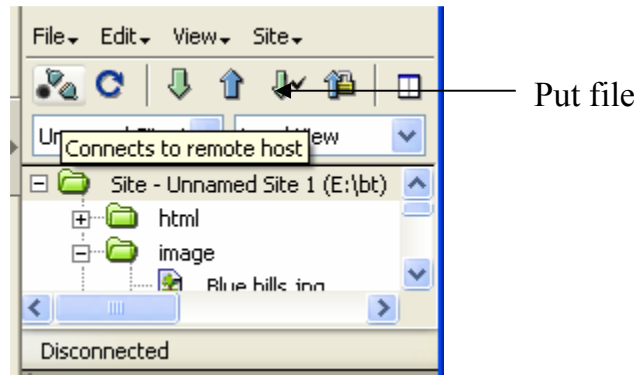
    - Click mũi tên, chọn Check Target Browser

- Chọn loại trình duyệt, version cần kiểm tra → Click check

### XIII.3.2. Kết nối và xuất bản

#### 1) Kết nối:

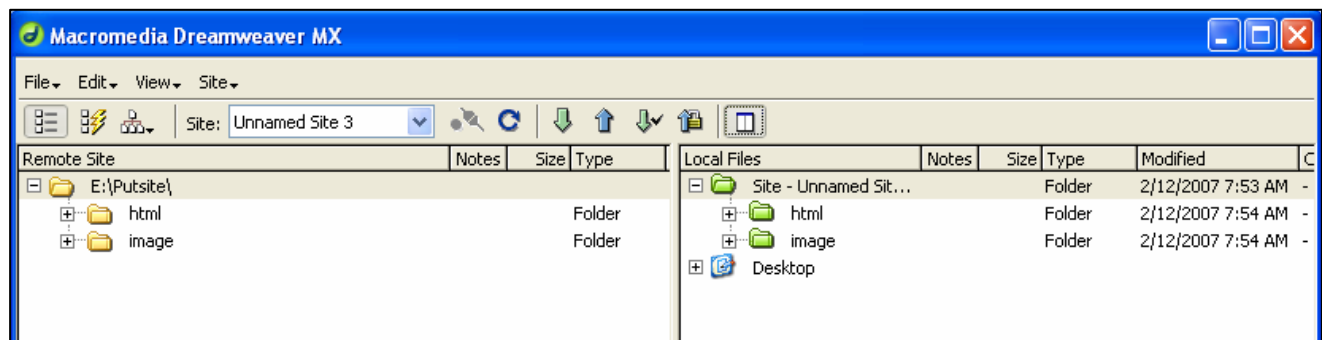
Sau khi hoàn tất việc kiểm tra, nếu có kết nối vào mạng, thì có thể xuất bản Site lên Server. Trong Site Panel click nút Connects to Remote host để kết nối, hoặc chọn Remote view.



- Nếu một site đã có kết nối thì sẽ hiển thị danh sách tập tin thư mục trên server.
- Nếu chưa kết nối thì phải thiết lập một kết nối lên server
  - Click dòng Define a remote Site
  - Trong Category: Chọn Remote Info
  - Access: Chọn Local/Network. Nếu bạn có quyền truy cập Server qua giao thức FTP thì có thể chọn FTP, nhập tên và mã số truy cập để kết nối
  - Chọn Remote Folder : Thư mục trên Server nơi sẽ chứa Site

#### 2) Xuất bản:

- Sau khi kết nối, thực hiện Put file lên server:
  - Click nút Put file
  - Xác nhận put toàn bộ website
  - Kiểm tra lại sau khi put file bằng cách click nút Expand Collapse để xem kết quả



## CHƯƠNG XIV: TỔNG QUAN VỀ JAVASCRIPT

### XIV.1. GIỚI THIỆU VỀ JAVASCRIPT:

Javascript ra đời với tên gọi LiveScript, sau đó Netscape đổi tên thành Javascript. Tuy nhiên giữa Java và Javascript có rất ít các điểm chung dù rằng cú pháp của chúng có thể có những điểm giống nhau.

Ngôn ngữ Javascript được tạo bởi Netscape vào năm 1996 và được đưa vào trong trình duyệt Netscape Navigator 2.0 của họ thông qua trình biên dịch để đọc và thực hiện các mã lệnh Javascript được kèm theo trong các trang HTML..

Javascript là một ngôn ngữ kịch bản (script) để viết kịch bản cho phía client. Client side là những yêu cầu của người sử dụng được xử lý tại máy khách. Thông thường những yêu cầu này là tính toán, kiểm tra tính hợp lệ của dữ liệu hay các hiệu ứng, các yêu cầu này thường không liên quan đến nguồn cơ sở dữ liệu trên server.

#### XIV.1.1. Đặc điểm của JAVASCRIPT:

- Javascript là một ngôn ngữ kịch bản được viết chung với HTML.
- Không biên dịch như các ngôn ngữ khác. Khi trang web load xuống nó được trình duyệt thông dịch.
- Javascript là ngôn ngữ thiết kế động vì các đối tượng có khả năng tương tác với nhau thông qua người sử dụng hoặc các sự kiện.
- Là ngôn ngữ hướng đối tượng. Phân biệt chữ hoa, chữ thường
- Được hỗ trợ bởi tất cả các trình duyệt như Netscape và Internet Explorer
- JavaScript có khả năng tạo và sử dụng các đối tượng(Object), các đối tượng gồm 2 nhóm:
  - Các Object do người sử dụng tạo ra gồm :
    - Định nghĩa thuộc tính cho đối tượng
      - Cú pháp: ***Object Name.Properties***
    - Thêm phương thức cho đối tượng
    - Tạo một instance của đối tượng
  - Các object có sẵn. JavaScript cung cấp một bộ các Built-in Object để cung cấp các thông tin về sự hiện hành của các đối tượng được load trong trang Web và nội dung của nó, các đối tượng này gồm phương pháp (method) làm việc với các thuộc tính (properties) của nó.

#### XIV.1.2. Cấu trúc của đoạn Javascript:

```
<Script language="JavaScript">
```

Các lệnh Javascript

```
</script>
```

#### XIV.1.3. JAVASCRIPT trong một trang HTML

- Đặt các dòng mã lệnh của Javascript giữa cặp tag <script></script>
- Có thể viết nhiều đoạn mã lệnh Javascript trong cùng một tập tin HTML. Các khối mã lệnh Javascript có thể đặt bất kỳ nơi nào của trang HTML. Có thể đặt trong cặp tag <head></head> hoặc trong cặp tag <body> </body> tuy nhiên ta nên đặt trong cặp tag <head> để dễ kiểm soát mã lệnh và cũng dễ sửa đổi chương trình.



- Có thể viết một tập tin Javascript riêng và sau đó kết nối với một hoặc nhiều tập tin trang web khác nhau.

❖ **Cách 1:** Viết đoạn mã script trong cùng trang HTML

Ví dụ 1:

```
<HTML>
  <HEAD>
    <script language="javascript" >
      document.write("What is your name? ");
    </script>
  </HEAD>
  <BODY>
    Nội dung của trang
  </BODY>
</HTML>
```

Ví dụ 2:

```
<HTML>
  <BODY>
    <script language="javascript">
      document.write("Hello World!")
    </script>
  </BODY>
</HTML>
```

Ví dụ 3:

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

❖ **Cách 2:**

Mở trình soạn thảo notepad, Viết đoạn chương trình Javascript. Lưu lại với phần mở rộng là **.js** ( lưu ý trong tập tin này không chứa bất kỳ một thẻ nào của ngôn ngữ HTML).

- Liên kết với một file **JavaScript.js** đã được xây dựng trước

Cú pháp:

```
<HTML>
  <BODY>
    <Script SRC="fileJavascript.js" Language="javascript" >
      JavaScript comments
    </Script>
  </BODY>
</HTML>
```

Lưu ý: trong thẻ JavaScript ta có thể bỏ thuộc tính *SRC* và *Language*, khi đó ngôn ngữ mặc định là JavaScript .

#### XIV.1.4. Môi trường viết JAVASCRIPT:

Có thể dùng chương trình soạn thảo: Frontpage, Notepad, Visual InterDev, Dreamweaver để viết mã Javascript, trong giáo trình này sẽ sử dụng môi trường Dreamweaver, chọn chế độ code, Dreamweaver hỗ trợ phân biệt từ khóa bằng màu chữ, hỗ trợ các hàm, thuộc tính của các tag, giúp người sử dụng thuận tiện trong việc thiết kế và viết chương trình

#### XIV.1.5. Lệnh đơn và khối lệnh:

##### a) Lệnh đơn:

Lệnh đơn là một câu lệnh được kết thúc bằng dấu chấm phẩy(;). Trong JavaScript cuối mỗi câu lệnh ta có thể dùng dấu (;) hoặc không dùng dấu gì cả .

##### b) Khối lệnh:

Khối lệnh là tập hợp nhiều câu lệnh đơn được bao bọc bởi cặp dấu {}

##### c) Lờì chú thích trong chương trình:

Lờì chú thích này trình duyệt sẽ bỏ qua khi thông dịch chương trình. JavaScript hỗ trợ 2 loại chú thích:

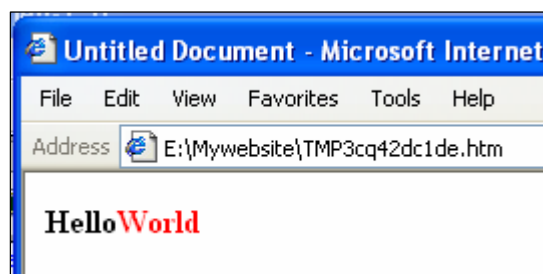
- Chú thích trên một dòng: dùng cặp dấu //
- Chú thích trên nhiều dòng: dùng cặp dấu /\*...\*/

#### XIV.1.6. Xuất dữ liệu ra trang Web

- JavaScript hỗ trợ 2 phương thức hiển thị dữ liệu ra trang Web là:
  - + **document.write("Text")**
  - + **document.writeln("Text")**
- Text là chuỗi dữ liệu muốn hiển thị ra trang Web, phải được đặt trong cặp nháy kép.
- Nếu xuất giá trị của biến thì không cần đặt trong nháy. Có thể dùng dấu + để nối các chuỗi và biến
- document.write("String " + variable );
- Nếu xuất tag HTML thì cặp tag đó cũng phải đặt trong cặp dấu nháy kép
- *document.writeln*: nếu đặt trong cặp tag <pre></pre> thì lệnh document.writeln xuất dữ liệu và xuống dòng. Nếu không có cặp tag <pre></pre> thì nó cách ra một khoảng trắng

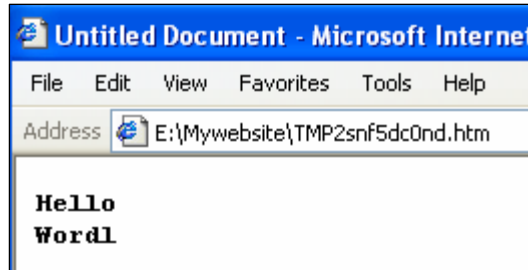
Ví dụ:

```
<BODY >
  <Script Language="JavaScript">
    document.write("<H1>Hello<H1>");
    document.write("<font color=red>World</font>");
  </Script>
</BODY>
```



Ví dụ:

```
<body>
  <pre>
  <script>
    document.writeln("<b>Hello</b>");
    document.writeln("<b>Wordl</b>");
  </script>
  </pre>
</body>
```



## XIV.2. BIẾN VÀ DỮ LIỆU TRONG JAVASCRIPT

### XIV.2.1. Biến

- 1. Khái niệm:** Biến là tên của một phần tử trong chương trình, được sử dụng để lưu trữ thông tin do người dùng nhập vào hoặc kết quả trung gian của quá trình tính toán, Khi khai báo biến trong Javascript không cần xác định kiểu dữ liệu cho biến cho nên khi một biến được khai báo xong nó có thể chứa bất kỳ kiểu dữ liệu nào.
- 2. Cách khai báo biến:** Trong JavaScript, để khai báo biến dùng từ khoá **var**, cũng có thể bỏ qua từ khóa var.

**var NameVariable ;**

- Một biến có thể được khai báo và khởi tạo hoặc không khởi tạo giá trị ban đầu
- Múôn khai báo nhiều biến cùng một lúc thì liệt kê tên biến kế tiếp nhau cách nhau bởi dấu (,)

Ví dụ: Var x = 7 ;

var y,z = "19" ;

- Trong JavaScript, 1 biến có thể chứa bất kỳ kiểu dữ liệu gì

Ví dụ:

var a="Hello World";

a=1999 ;

- 3. Cách xuất giá trị của biến:**

**document.write(NameVariable )**

- 4. Quy tắc đặt tên biến:**

Tên biến gồm các chữ cái và số, không dùng các ký tự đặc biệt như: ( , [ , { , # , & .... theo nguyên tắc sau:

- Tên biến phải bắt đầu bằng ký tự hoặc ký tự gạch dưới( \_ )
- Không bắt đầu bằng ký tự số.
- Không chứa khoảng trắng, tên biến phải gọi nhớ
- Không trùng với từ khoá của JavaScript
- Các từ khoá trong JavaScript

abstract	extends	Int	super
boolean	false	interface	switch
break	final	Long	synchronized
byte	finally	native	this
case	float	New	throw
catch	for	Null	throws
char	Function	package	transient
class	goto	private	true
const	if	protected	try
continue	implements	public	var
default	import	return	val
do	In	short	while
double	instanceof	static	with
else			

**5. Tầm vực của biến:** là tầm ảnh hưởng của biến trong chương trình. Có 2 loại biến:

- *Biến toàn cục* : được khai báo ngoài các hàm. Phạm vi hoạt động của biến là từ vị trí khai báo trở về sau trong chương trình.
- *Biến cục bộ*: được khai báo trong chương trình con. Phạm vi hoạt động của biến là từ vị trí khai báo đến kết thúc chương trình con.

Lưu ý: Nếu tên biến toàn cục và cục bộ trùng nhau thì biến được sử dụng trong hàm là biến cục bộ.

#### XIV.2.2. Dữ liệu: Có 4 loại dữ liệu

- *Kiểu số*: một biến kiểu số chứa bất kỳ giá trị số nào: số thập phân, số nguyên, số dạng chấm phẩy động.
- *Kiểu chuỗi*: một biến kiểu chuỗi có thể chứa một nhóm ký tự (Chữ cái, ký tự số, khoảng trắng, các ký tự đặc biệt, ...). Giá trị chuỗi phải đặt trong cặp dấu nháy đôi (“”) hoặc đơn (‘ ‘)

Ví dụ:

```
var s1, s2, s3 ;
s1="Hello World" ;
s2='Hello World' ;
```

- *Kiểu Boolean*: Là dữ liệu chỉ có 2 giá trị False hoặc True thường dùng trong trường hợp biến hoặc hàm chỉ nhận một trong 2 trạng thái đúng hoặc sai.

Ví dụ: var bl;

```
bl=true ;
```

- *Kiểu Null*: là biến không gán cho giá trị

#### XIV.2.3. Toán tử:

##### 1. Toán tử số học

T toán Tử	Chức Năng	Ví dụ	Kết quả
+	cộng	x=2 x+2	4
-	Trừ	x=2 5-x	3
*	Nhân	x=4 x*5	20

/	Chia	15/5 5/2	3 2.5
%	Lấy phần dư	5%2 10%8 10%2	1 2 0
++	Tăng giá trị lên 1	x=5 x++	x=6
--	Giảm giá trị xuống 1	x=5 x--	x=4

## 2. Toán Tử Gán

T toán Tử	Ví dụ	Tương đương
=	x = y	x = y
+=	x += y	x = x+y
-=	x -= y	x = x-y
*=	x *= y	x = x*y
/=	x /= y	x = x/y
%=	x %=y	x = x%y

## 3. Toán Tử so sánh

T toán Tử	Chức Năng	Ví dụ
==	bằng	5==8 returns false
!=	Không bằng	5!=8 returns true
>	lớn hơn	5>8 returns false
<	nhỏ hơn	5<8 returns true
>=	lớn hơn hoặc bằng	5>=8 returns false
<=	nhỏ hơn hoặc bằng	5<=8 returns true

## 4. Toán Tử logic

T toán Tử	Chức Năng	Ví dụ
&&	Và	x =6; y =3 ; (x < 10 && y > 1) returns true
	hoặc	x = 6 ; y =3 (x==5    y==5) returns false
!	not	x=6; y =3; !(x==y) returns true

## 5. Toán tử chuỗi

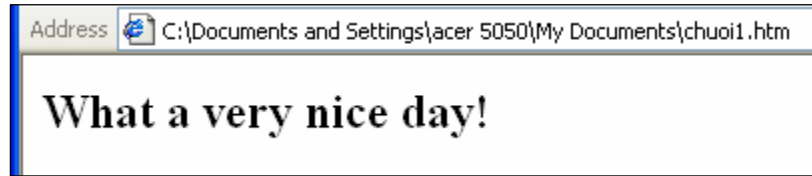
- Ký hiệu: + : Là phép toán nối hai chuỗi với nhau

Ví dụ:  
<html>

```

<script>
  txt1="What a very";
  txt2="nice day!";
  document.write('<h2>'+txt1+txt2+'</h2>');
</script>
</html>

```



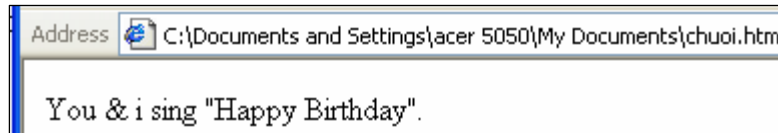
- Một số ký tự đặc biệt: \n ( new line), \t (tab), \b (BackSpace), \& (dấu &), \'(“)

Ví dụ:

```

<html>
  <script>
    document.write ("You \& i sing \"Happy Birthday\".")
  </script>
</html>

```



## 6. Toán tử Điều kiện:

**Cú pháp:**

**(Điều kiện) ? value1: value2**

- Nếu biểu thức điều kiện đúng thì trả về giá trị value 1
- Nếu biểu thức điều kiện sai thì trả về giá trị value 2

Ví dụ:

```

<html>
  <script>
    a=5;  b=6;
    document.write((a>b)? 'a lon hon b':'b lon hon a');
  </script>
</html>

```

## CHƯƠNG XV: HÀM TRONG JAVASCRIPT

### XV.1. ĐỊNH NGHĨA

Hàm là một đoạn chương trình có thể được sử dụng nhiều lần trong một chương trình để thực hiện một tác vụ nào đó.

**XV.1.1. Xây dựng hàm: Trong JavaScript, dùng từ khoá *function* để định nghĩa hàm. Một hàm có cấu trúc như sau:**

```
function NameFunction( List_Parameter )
{
    Khai báo các biến sử dụng trong hàm ;
    Các câu lệnh trong JavaScript thực hiện tác vụ;
    [return [giá trị /biểu thức] ];
}
```

- *NameFunction*: là tên hàm do người lập trình tự đặt. Quy tắc đặt tên hàm giống như tên biến. Sau *NameFunction* là cặp dấu ngoặc ( ) chứa danh sách tham số hình thức. Nếu hàm không có tham số thì cặp dấu ngoặc ( ) cũng phải viết sau *NameFunction*.
- *List\_Parameter*: là danh sách các tham số hình thức, nếu có nhiều tham số có thì các tham số phải cách nhau bởi dấu phẩy, các tham số này không chỉ ra kiểu dữ liệu cụ thể và cũng không cần từ khoá var.

Ví dụ:

```
function Display(user , pwd)
{
    document.write("UserName của bạn là:" + user) ;
    document.write("Password của bạn là:" + pwd) ;
    return ;
}
```

- **Câu lệnh *return***: là câu lệnh kết thúc hàm. Câu lệnh này là tùy chọn. Có thể bỏ qua, nếu hàm có giá trị trả về thì cần có câu lệnh Return để trả về giá trị. Sau Return có thể chứa hoặc không chứa một giá trị cụ thể hoặc một biểu thức tính toán.

Ví dụ:

```
Function total(a,b)
{ C=a+b;
    Return c;
}
```

### XV.1.2. Cách gọi hàm

- Hàm sẽ không thực hiện cho đến khi nó được gọi.
- Đối với hàm có đối số ta gọi tên hàm và danh sách các giá trị truyền cho đối số đó

**FunctionName(argument1,argument2,etc)**

- Đối với hàm không có đối số ta chỉ cần gọi tên hàm là được.

**FunctionName()**

- Đối với hàm không có giá trị trả về :

**NameFunction(parameter) .**

- Đối với hàm có giá trị trả về :

**variable= NameFunction(parameter) .**

Ví dụ:

```
<html>
  <head><title>Function</title></head>
  <body>
    <script>
      function Area(Width, Length)
        {
          size=Width*Length;
          return size;
        }
      x=eval(prompt("Nhập x: "));
      y= eval(prompt("Nhập y: "));
      document.write(Area(x,y))
    </script>
  </body>
</html>
```

## XV.2. CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

- 1) **Hàm alert():** dùng hiển thị một hộp thông báo có nút OK

Cú pháp:

***alert("nội dung thông báo")***

ví dụ:

```
<html>
  <head><title>Function</title></head>
  <body>
    <script>
      alert("Hello World")
    </script>
  </body>
</html>
```



- 2) **Hàm prompt():** tạo hộp thoại chứa 2 nút OK và Cancel, và một textbox để người sd nhập nội dung, giá trị trả về của hàm prompt là nội dung nhập trong textbox

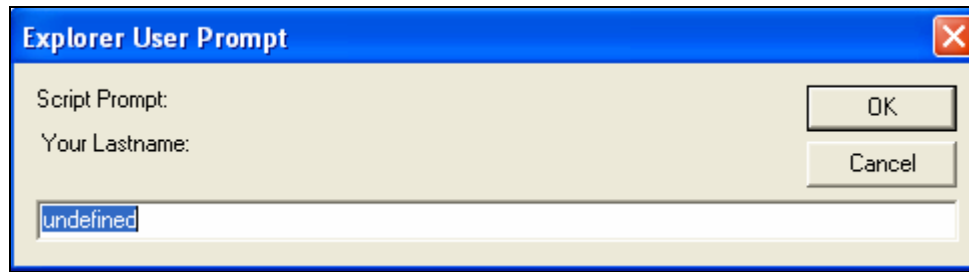
Cú pháp:

***variable= prompt("nội dung đối thoại", giá trị khởi tạo);***

ví dụ:

```
<html>
  <head><title>Function</title></head>
  <body>
    <script>
      a=prompt("Your Lastname:");
      b=prompt("Your FirstName");
      document.write("Your FullName is :"+ a + ' ' + b)
    </script>
  </body>
</html>
```





- 3) **Hàm confirm():** Hiện thị hộp thông báo có 2 nút OK và Cancel. Hàm trả về giá trị true nếu người sd click OK và ngược lại thì trả về giá trị false.

Cú pháp:

**variable=confirm("Chuoi thong bao");**

**Ví dụ:**

```
<html>
  <head><title>Function</title></head>
  <body>
    <script>
      a=prompt("nhap so a :");
      b=prompt("nhap so b");
      c=confirm( a +' lon hon '+ b+'?')
      if(c= =true)
        document.write( a +" > "+b )
      else
        document.write( a +" < "+b )
    </script>
  </body>
</html>
```



- 4) **Các hàm thông dụng của chuỗi và số:**

**a. Hàm eval():** Trả về giá trị số của một chuỗi số

Cú pháp:

**eval(chuoi số)**

Ví dụ:

```
<script>
  var str1="123", str2="456";
  str= str1+str2;
  document.write(str); → kết quả :123456
</script>
<script>
  var str1="123", str2="456";
  str=eval(str1)+eval(str2) ;
  document.write(str) → kết quả: 579
</script>
```

**b. Hàm parseInt(strNum)**

Trả về một số nguyên từ chuỗi strNum. Nếu strNum theo sau là ký tự chữ thì các ký tự này sẽ bị bỏ qua. Nếu strNum không bắt đầu bằng số thì hàm này trả về giá trị NaN (Not a Number)

Ví dụ :

```
var strNum="123.8" , kq;
kq=parseInt(strNum) =>kq=123
strNum="a123"
kq=parseInt(strNum) =>kq=NaN
strNum="123.8abc"
kq=parseInt(strNum)=>kq=123
```

### c. Hàm **parseFloat(strNum)**:

Hàm trả về một số thực từ chuỗi strNum. Nếu chuỗi strNum bắt đầu là số và theo sau là các ký tự chữ thì các ký tự này bị bỏ qua. Nếu chuỗi strNum bắt đầu từ ký tự chữ thì hàm trả về giá trị NaN.

Ví dụ:

```
var strNum="123.8" , kq;
kq=parseFloat(strNum) =>kq=123.8
strNum="a123.8"
kq=parseFloat(strNum) =>kq=NaN
strNum="123.8abc"
kq=parseFloat(strNum)=>kq=123.8
```

### d. Hàm **isNaN(str)**:

Hàm trả về giá trị True nếu str là chuỗi, ngược lại là False nếu str là chuỗi số.

Ví dụ :

```
Var str="123abc" , kq;
kq=isNaN(str) =>kq=true;
str="123.8"
kq=isNaN(str) =>kq=false ;
```

## 5) Các hàm thiết lập thời gian:

**a. Hàm Timeout( ):** Báo cho JavaScript thực hiện một lệnh JavaScript sau một khoảng thời gian nào đó. Hàm trả về một ID(duy nhất đối với mỗi hàm setTimeout thực hiện một lệnh) Giá trị ID này dùng để xoá khoảng thời gian đã thiết lập nếu không cần thực hiện hàm Timeout nữa .

Cú pháp:

**IdTime=setTimeout("Command JavaScript", delayTime);**

- *Command JavaScript* : có thể là lời gọi hàm hoặc là một câu lệnh đơn
- *delayTime* :là khoảng thời gian chờ để thi hành Command JavaScript, được tính bằng mili giây.

Ví dụ:

```
Idq=setTimeout("alert('Da het gio')",1000) ;
Cứ 1000 mili giây thì thông báo đã hết giờ một lần.
```

**b. Hàm clearTimeout():**Hủy thời gian đã thiết lập bởi setTimeout().

Cú pháp:

**clearTimeout(IdTime );**

Ví dụ:

```
clearTimeout(Idq);
```

**c. Hàm setInterval() và clearInterval()** với ý nghĩa và tham số giống như setTimeout() và clearTimeout() .

## CHƯƠNG XVI: CÁC CẤU TRÚC ĐIỀU KIỆN

### XVI.1. CẤU TRÚC LỰA CHỌN:

#### XVI.1.1. Câu lệnh if:

- **Mẫu 1:** Áp dụng cho trường hợp có 1 điều kiện và 1 công việc xử lý

**Cú pháp:**

**if (<Biểu thức điều kiện>)**

**Khối lệnh 1;**

**Khối lệnh 2;**

*Nguyên tắc hoạt động:* Nếu biểu thức điều kiện đúng thì thực hiện khối lệnh 1, sau đó thực hiện khối lệnh 2, ngược lại nếu biểu thức điều kiện sai thì bỏ qua khối lệnh 1 và thực hiện khối lệnh 2

- **Mẫu 2:** Áp dụng cho trường hợp có 1 điều kiện và 2 lựa chọn công việc xử lý

**Cú pháp:**

**if(<biểu thức điều kiện>)**

**Khối lệnh1;**

**else**

**Khối lệnh 2 ;**

**Khối lệnh 3;**

*Nguyên tắc hoạt động:* Nếu biểu thức điều kiện đúng thì thực hiện khối lệnh 1, sau đó thực hiện khối lệnh 3, ngược lại thì thực hiện khối lệnh 2, sau đó thực hiện khối lệnh 3

- **Mẫu 3 (if ...else lồng nhau):** Áp dụng cho trường hợp có nhiều chọn lựa khác nhau

**Cú pháp:**

**if(<biểu thức điều kiện1>)**

**Khối lệnh 1;**

**else**

**if (<biểu thức điều kiện 2>)**

**Khối lệnh 2 ;**

**else**

**...**

**khối lệnh 3**

Để áp dụng mẫu 3, cần phải xác định biểu thức điều kiện của bài toán rồi sắp xếp thứ tự lồng nhau cho hợp lý.

**Ví dụ:** Viết chương trình nhập 3 cạnh của tam giác sau đó xuất ra màn hình đó là tam giác gì?

```
<Body><script>
a=eval(prompt("Nhập cạnh a"));
b=eval(prompt("Nhập cạnh b"));
c=eval(prompt("Nhập cạnh c"));
if(a==b && b==c && c==a)
```

```

    Tam giác đều ;
else
    if(a= =b || b= = c || c= = a)
        Tam giác cân
    Else
        Tam giác thuong
</script></Body>

```

### XVI.1.2. Cấu trúc chọn lựa switch...case:

Áp dụng trong trường hợp muốn chọn một trong các giá trị của biểu thức để thực hiện lệnh. Giá trị của biểu thức có thể là một chuỗi hoặc một số

- **Mẫu 1:**

```

switch(Biểu thức)
{
    case value1:
        Khối lệnh 1;
        break;
    case value2:
        Khối lệnh 2 ;
        break;
    .....
    case valuek:
        Khối lệnh k ;
        break;
}

```

- **Mẫu 2:**

```

switch(biểu thức)
{
    case value1:
        khối lệnh 1 ;
        break;
    case value2:
        khối lệnh 2 ;
        break;
    .....
    case valuek:
        khối lệnh k ;
        break;
    default :
        khối lệnh k+1 ;}

```

Nguyên tắc hoạt động:

- Trình thông dịch sẽ tính giá trị của biểu thức rồi so sánh với các value, nếu bằng giá trị nào thì thực hiện khối lệnh đó.
- Sự khác nhau giữa mẫu 1 và 2 là: ở mẫu 2 khi so sánh giá trị của biểu thức với các value, nếu không khớp thì thực hiện lệnh trong default
- Trong trường hợp có nhiều value khác nhau mà cùng thực hiện một khối lệnh thì liệt kê các value liên tiếp nhau và cách nhau dấu phẩy.

**case** *valuej1* , *valuej2* , ..., *valuejk* : khởi lệnh; *break*;

Ví dụ:

```
<body>
  <script>
    t=prompt("nhap thang: ");
    switch(eval(t))
    {
      case 1: case 3: case 5: case 7: case 8 : case 10: case 12:
        alert("Thang "+ t+ " co 31 ngay");
        break;
      case 2:
        alert("Thang "+t + " co 28 ngay");
        break;
      case 4: case 6: case 9: case 11:
        alert("Thang "+t +" co 30 ngay");
        break;
      default:
        alert("Khong co thang nay");
    }
  </script>
</body>
```

## XVI.2. CẤU TRÚC LẶP:

Được áp dụng khi một công việc nào đó muốn thực hiện lặp đi lặp lại nhiều lần với một điều kiện nào đó. Có 2 loại cấu trúc lặp là : lặp với số lần lặp biết trước và lặp với số lần lặp không biết trước

### XVI.2.1. Vòng lặp For: Thường áp dụng cho số lần lặp biết trước

**Cú pháp:**

**for**(**biểu thức 1**; **biểu thức 2**; **biểu thức 3**)

```
{
  Khởi lệnh 1;
}
khởi lệnh 2;
```

Trong đó :

*biểu thức 1*: chứa giá trị khởi tạo của biến điều khiển

*biểu thức 2* : chứa biểu thức điều kiện lặp.

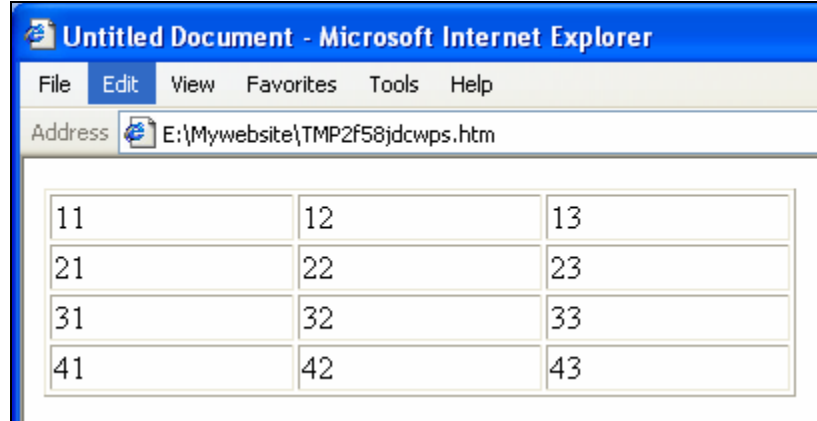
*biểu thức 3*: chứa biểu thức tăng hoặc giảm biến điều khiển .

Nguyên tắc hoạt động:

- Trình thông dịch gán giá trị khởi tạo cho biến điều khiển, Kiểm tra biểu thức 2, nếu đúng thì thực hiện khởi lệnh 1, chuyển lên thực hiện biểu thức 3, tiếp tục kiểm tra biểu thức 2, và tiếp tục ...
- Nếu biểu thức 2 có giá trị sai thì chương trình thoát khỏi vòng lặp và thực hiện khởi lệnh 2.
- Nếu khởi lệnh 1 có chứa câu lệnh Break thì chương trình sẽ thoát khỏi vòng lặp for và thực hiện khởi lệnh 2

Ví dụ: Viết chương trình tạo một table m dòng n cột.

```
<body>
  <Script language="javascript">
    var n, m, i, j;
    m=prompt("Nhập số dòng");
    n=prompt("Nhập số cột");
    document.write("<table width=50% border=1>");
    for(i=1;i<=m;i++)
    {
      document.write("<tr>");
      for(j=1;j<=n;j++)
        document.write("<td>" + i + j + "</td>");
      document.write("</tr>");
    }
    document.write("</table>");
  </Script>
</body>
```



## XVI.2.2. Vòng lặp while: thường áp dụng cho số lần lặp không xác định

### 1. Vòng lặp While: Kiểm tra điều kiện trước khi thực hiện lệnh

#### Cú pháp:

```
while(biểu thức điều kiện)
{
  khối lệnh 1;
}
khối lệnh 2;
```

#### Nguyên tắc hoạt động:

- Trình thông dịch kiểm tra biểu thức điều kiện, nếu đúng thì thực hiện khối lệnh 1, sau đó quay lại kiểm tra biểu thức điều kiện, và tiếp tục ..., nếu sai thì thực hiện khối lệnh 2.
- Như vậy khối lệnh 1 có thể không được thực hiện lần nào nếu ngay từ đầu biểu thức điều kiện sai
- Thường khối lệnh 1 chứa lệnh làm thay đổi giá trị của biểu thức điều kiện để có thể thoát ra khỏi vòng lặp, hoặc chứa lệnh break để thoát khỏi vòng lặp *while*

Ví dụ:

```
<script language="javascript">
  var userinput;
  while ((userinput!=99 )
  {
    userinput=prompt("Nhập vào một số bảy kỳ, nhập 99 để
thoát")
    if(isNaN(userinput)
    {
      document.write("Dữ liệu không hợp lệ, nhập số ");
      break;
    }
  }
}</script>
```

**2. Vòng lặp do ...while:** Thực hiện lệnh trước sau đó kiểm tra biểu thức điều kiện

**Cú pháp:**

```
do
{
  khối lệnh 1;
} While(biểu thức điều kiện);
khối lệnh 2;
```

*Nguyên tắc hoạt động:* trình thông dịch thực hiện khối lệnh 1, sau đó kiểm tra biểu thức điều kiện, nếu đúng thì thực hiện lại khối lệnh 1, nếu sai thì thoát khỏi vòng lặp và thực hiện khối lệnh 2

*Ví dụ:* Viết chương trình yêu cầu người dùng nhập vào một số, kiểm tra xem giá trị nhập có phải là số không, nếu không yêu cầu nhập lại.

```
<script language="javascript">
  var userinput;
  do
  {
    userinput=prompt("Nhập vào một số bảy kỳ, nhập 99 để thoát")
    if(isNaN(userinput)
    {
      document.write("Dữ liệu không hợp lệ, nhập số ");
      break;
    }
  }while ((userinput!=99 )
</script>
```

**3. Vòng lặp for ...in:** dùng để duyệt qua các thuộc tính của một đối tượng hay giá trị của các phần tử trong mảng

**Cú pháp:**

```
for ( variable in Object)
{
  khối lệnh 1 ;
}
```

**khối lệnh 2;**

Nguyên tắc hoạt động: trình thông dịch sẽ duyệt qua tất cả các phần tử trong Object.

Ví dụ:

```
<body>
  <script>
    obj= new Array() ;
    obj[0]="Hello";
    obj[1]="World" ;
    for(i in obj)
      document.write(obj[i]);
  </script>
</body>
```



### XVI.2.3. Câu lệnh try ...catch và throw: dùng xử lý lỗi trong các modul. Nó được dùng trong Internet Explorer 5 và trong IIS

#### Cú pháp:

```
try
{
    khối lệnh ;
}
catch(objErr)
{
    Xử lý lỗi ;
}
```

Nguyên tắc hoạt động:

- Trình thông dịch thực thi các lệnh trong khối lệnh, nếu trong quá trình thực thi có lỗi xảy ra thì trình thông dịch truyền đối tượng lỗi cho catch.
- Câu lệnh catch tự động gửi vào tham số có chứa đối tượng lỗi, đối tượng này có 2 thuộc tính number và description. mỗi dạng lỗi trong mã kịch bản sẽ được gán cho một con số lỗi duy nhất. thuộc tính Number chứa một số nguyên lỗi, thuộc tính description chứa một mô tả dạng văn bản về lỗi.

Ví dụ:

```
<head><title>Chương trình kiểm tra lỗi</title>
<Script language="JavaScript">
  var str ;
  try
  {
    document.write("Hello World");
    Math.r();
  }
  catch(objerr)
  {
    str="Lỗi thu " + objerr.number + "<br>";
    str="Va lỗi do la " + objerr.description;
    alert(str);
  }
</Script>
</head>
```



- Câu lệnh throw được dùng để truyền một thông báo lỗi đến một câu lệnh catch. Nó cũng có thể được dùng để truyền một lỗi lên
- Bộ xử lý lỗi mức cao hơn trong trường hợp có nhiều câu lệnh try...catch lồng nhau

Ví dụ:

```
<Html><head><title>Chuong trinh kiem tra loi</title>
<Script language="JavaScript">
    var str , m=4 ,kq;
    try
    {
        try
        {
            document.write("Hello World");
            kq=m/n;
        }
        catch(objerr)
        {
            str="Loi thu " + objerr.number + "<br>";
            str="Va loi do la " + objerr.description;
            if (kq==4)
                alert("n=1");
            else
                throw (objerr);
        }
    }
    catch (objerr)
    {
        alert(objerr.number + objerr.description);
    }
</Script></head></html>
```

## CHƯƠNG XVII: MÔ HÌNH ĐỐI TƯỢNG

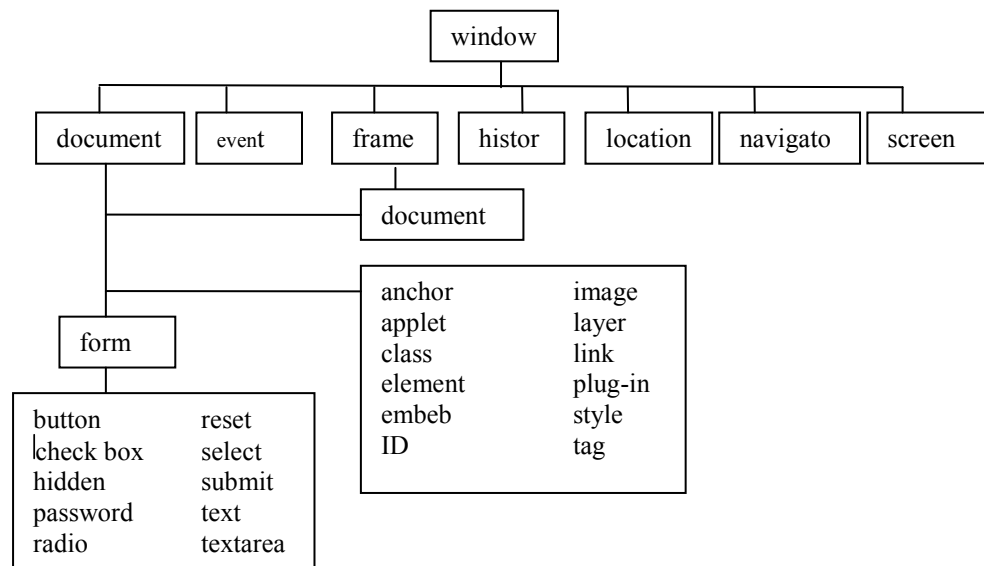
### XVII.1. MÔ HÌNH DOM ((Document Object Model)

#### 1. Đối tượng – Mô hình đối tượng:

- Mỗi thành phần trên trang web được xem như một đối tượng, mỗi đối tượng đều có các thuộc tính, phương thức và sự kiện của nó. Ví dụ hình ảnh, văn bản, button, một tag trong HTML cũng được xem như là một đối tượng và các thuộc tính của tag đó được xem như là đối tượng con của nó. 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
- Dùng thuộc tính ID để truy xuất đến một đối tượng trong IE và thay đổi dữ liệu cho chính phần tử đó, tất cả các đối tượng trên trang đều có một ID duy nhất

Ví dụ:

```
<tr><td Id=IdName1>Data</td></tr>
<img Id=Id Name2>
<Iframe Id=IdName3></Iframe>
```



- Muốn truy cập vào đối tượng nào thì phải truy cập vào đối tượng chứa nó trước, dùng toán tử dấu chấm (.) để phân cách giữa các đối tượng. Tuy nhiên ta có thể bỏ qua đối tượng window nếu đang thao tác trên cửa sổ hiện hành

Ví dụ: window.location

- Ngoài các đối tượng do chương trình xây dựng sẵn, có thể tạo thêm những đối tượng mới cần thiết cho nhu cầu sử dụng.
- Mỗi đối tượng đều có *thuộc tính*, *sự kiện* và *phương thức*, nhờ các thành phần này mà có thể truy cập và thay đổi nội dung của chúng.
- o Thuộc tính (Properties): là nơi chứa các mô tả thông tin của đối tượng.

Ví dụ: 

Trong tag Img có 3 thuộc tính src, width, height, để thay đổi kích thước của hình thì thay đổi giá trị của thuộc tính width, height hoặc đổi hình khác thì thay

đổi giá trị của thuộc tính, để thực hiện các việc trên ta dựa vào thuộc tính Id là Idh1.

```

Ví dụ: function ZoomIn()
{
    Idh1.width=Idh1.width +10;
    Idh1.height=Idh1.height + 5;
}
function ChangeImg(file)
{
    Idh1.src=file ;
}

```

- *Sự kiện (event)*: là các hành động, sự việc xảy ra trên trang web: click chuột, di chuyển chuột, ...gọi là sự kiện. Sự kiện được xử lý bởi các đoạn mã kịch gọi là bộ xử lý sự kiện

- *Các sự kiện thường sử dụng:*

Tên sự kiện	Ý nghĩa
Onmousedown	Phát sinh khi người sdụng nhấn chuột
Onmouseover	Phát sinh khi người sdụng d/chuyển chuột lên đối tượng
Onmouseout	Phát sinh khi người sdụng d/chuyển chuột ra ngoài đtượng
Onkeypress	Phát sinh khi người sử dụng nhấn một phím
Onfocus	Phát sinh khi đối tượng nhận tiêu điểm
Onblur	Phát sinh khi rời khỏi đối tượng
Onclick	- Người dùng click chuột vào đối tượng - Một đối tượng đang có tiêu điểm, người sd nhấn enter - Một checkbox hoặc nút chọn đang có tiêu điểm, người sd nhấn phím Spacebar
Onload	- Phát sinh khi đối tượng được tải xuống
OnUnload	- Phát sinh khi đối tượng được nạp trở lại hoặc chuyển trang
Onresize	- Phát sinh khi cửa sổ bị thay đổi kích thước
Onselect	- Phát sinh khi đối tượng được chọn
Onchange	- Phát sinh khi đối tượng thay đổi giá trị
Onsubmit	- Phát sinh khi Form được Submit

- *Cách sử dụng các sự kiện*: muốn điều khiển sự kiện, ta thêm sự kiện đó vào trong thẻ HTML.
- *Cú pháp*:< TagName event\_handler="JavaScript Command">
  - TagName: tên tag
  - event\_handler: tên sự kiện
  - JavaScript Command: gọi hàm xử lý sự kiện
- *Phương thức*: Là các hàm được xây dựng trước có tác dụng làm thay đổi thuộc tính của đối tượng.

## 2. Mục đích của mô hình DOM:

- Để định nghĩa 1 tổ chức phân cấp thể hiện các phần của 1 hồ sơ web.
- Cho phép thay đổi cấu trúc đó thông qua việc thêm bớt nội dung
- Cung cấp cách thức quan sát, thao tác các đặc tính của nội dung trên trang web
- Cung cấp thông tin về cách tương tác giữa các mục trên trang web với người dùng
- Nó cho phép thông báo các sự kiện gây ra do chuột và bàn phím

### XVII.1.1. Xây dựng một đối tượng mới:

Ngoài các đối tượng có sẵn trong Javascript, ta có thể tạo các đối tượng mới với các phương thức và thuộc tính riêng cho đối tượng đó.

#### 1. Cách xây dựng một đối tượng mới: Gồm 2 bước

a. **Bước 1:** Định nghĩa đối tượng bằng cách xây dựng hàm cho đối tượng đó gồm các phương thức và thuộc tính cho đối tượng đó.

```
function Object(List Parameter)
{
    this.property1= Parameter1;
    this.property2= Parameter2;
    ...
    this.method1=functionName1;
    this.method2=functionName2;
    ...
}
```

Trong đó

- Từ khoá **this** để tham chiếu đến đối tượng đang được tạo. Khi xây dựng đối tượng có bao nhiêu thuộc tính thì dùng từ khoá this để tham chiếu đến bấy nhiêu thuộc tính của nó
- Câu lệnh **this.property1= Parameter1**: gán giá trị Parameter1 cho thuộc tính property1
- Tương tự: muốn xây dựng phương thức cho đối tượng thì gán phương thức cho hàm đã định nghĩa sẵn

```
    this.method1=FunctionName1;
```

b. **Bước 2:** Tạo instance cho đối tượng, dùng từ khoá **new**

```
var obj=new Object();
```

- Truy cập hoặc thay đổi g/ trị của thuộc tính ta sử dụng: **obj.property**
- Muốn sử dụng phương thức method1 thì dùng **obj.method()**

Ví dụ:

Xây dựng một đối tượng Student gồm các thuộc tính IdStudent, Name, Address và phương thức Display() hiển thị thông tin của Student

Bước 1: Xây dựng đối tượng Student

```
function Student(masv,hten,dchi) // Đối tượng
{
    this.IdStudent=masv;
    this.Name=hten;
    this.Address=dchi;
    this.Display=Information;
}
function Display() //Phương thức
{
    document.write("Ma SV "+this.IdStudent + "<br>");
    document.write("Ho ten SV "+this.Name + "<br>");
    document.write("Dia chi SV "+this.Address + "<br>");
}
```

- Muốn xuất thông tin SV ta gọi phương thức Display()

- Sử dụng đối tượng Student: tạo instance cho đối tượng
 

```
var st=new Student() ; //Tạo thể thiện cho đối tượng
st.IdStudent="TH01";
st.Name="Truong Tam Phong" // Khởi tạo giá trị cho đtượng
st.Address="12 Nguyen Cuu Van"
```
- Hiện thị thông tin của đối tượng thì gọi đến phương thức Display()
 

```
st.Display()
```
- Có thể khai báo và khởi tạo đối tượng bằng cách:
 

```
var st=new("TH01","Truong Tam Phong","12 Nguyen Cuu Van")
```

## XVII.2. CÁC ĐỐI TƯỢNG CÓ SẴN TRONG JAVASRIPT

### XVII.2.1. Đối tượng Array()

Đối tượng Array dùng để lưu trữ nhiều giá trị với cùng một tên gọi. Trong Javascript đối tượng mảng có thể chứa các thành phần mang kiểu giá trị khác nhau. Một mảng có n phần tử được đánh chỉ số từ 0 đến n-1.

Mỗi phần tử mảng được phân biệt nhau qua chỉ số, dựa vào chỉ số này ta có thể truy cập hoặc thay đổi giá trị của từng phần tử trong mảng

#### 1. Khởi tạo một mảng:

Dùng từ khóa **new** để khởi tạo một mảng

```
var Variable = new Array(size)
```

Ví dụ: <script>

```
var arr= new Array()
arr[0]= "thu hai";
arr[1]= "Thu ba";
arr[2]= "Thu tu";
arr[3]= "Thu nam";
arr[4]= "Thu sau";
arr[5]= "Thu bay";
for(i=0; i<=5;i++)
document.write(arr[i]+ "<br>")
```

</script>

#### 2. Các thuộc tính của Array()

- **length** : để xác định số phần tử trong mảng

Ví dụ: <script>

```
var arr= new Array()
arr[0]= "thu hai";
arr[1]= "Thu ba";
arr[2]= "Thu tu";
arr[3]= "Thu nam";
arr[4]= "Thu sau";
arr[5]= "Thu bay";
document.write("So phan tu trong mang la: " +arr.length)
```

</script>

#### 3. Các phương thức của đối tượng Array()

Phương thức	Ý nghĩa	Ví dụ
concat()	Dùng để nối 2 mảng	a=a.concat(b)
join(separator)	để ghép các phần tử trong mảng lại với nhau cách nhau bởi dấu separator	a=a.join("+")
slice(start,end)	Dùng tách một mảng bắt đầu từ vtrí start đến vtrí end-1.	str=a.slice(i,j)
reverse()	Dùng để đảo ngược chuỗi	a.reverse()
valueOf()	Dùng để lấy tất cả các đối tượng trong chuỗi	a.valueOf()
pop()	Lấy phần tử cuối của mảng	
push()	Thêm 1 hoặc nhiều phần tử vào cuối mảng	
Shift()	lấy phần tử và trả về phần tử đầu tiên của mảng	
Sort()	sắp xếp các phần tử của mảng	
valueOf()	Trả về tất cả các giá trị ban đầu của mảng	

Ví dụ 1:

```
<script>
```

```
var a=new Array()
a[0]= "Thang gieng";
a[1]= "Thang hai";
a[2]= "Thang ba";
var b =new Array();
b[0]= "Thang tu";
b[1]= "Thang nam";
b[2]= "Thang sau";
a=a.concat(b);
document.write(a);
```

```
</script>
```

Ví dụ 2

```
<script type="text/javascript">
```

```
var arrName = new Array(3)
arrName [0] = "Jani"
arrName [1] = "Tove"
arrName [2] = "Hege"
document.write(arrName.length + "<br>")
document.write(arrName.join(".") + "<br>")
document.write(arrName.reverse() + "<br>")
document.write(arrName.sort() + "<br>")
document.write(arrName.push("Ola","Jon") + "<br>")
document.write(arrName.pop() + "<br>")
document.write(arrName.shift() + "<br>")
```

```
</script>
```

**XVII.2.2. Đối tượng Date() : Cung cấp thông tin về ngày, giờ trên môi trường client. Dùng thiết lập ngày tháng năm và giờ hiện hành trên trang web.**

1. **Cách khai báo:** Có 2 cách khai báo

**Cách 1:** Khai báo và khởi tạo

```
var variableName= new Date("month, day, year , hours : minutes : seconds")
```

hoặc:

```
var variableName= new Date(year,month,day,hours,minutes,seconds)
```

hoặc:

```
var variableName= new Date(year,month, day)
```

```
var variableName= new Date("Month dd, yyyy hh:mm:ss")
```

```
var variableName= new Date("Month dd, yyyy")
```

```
var variableName= new Date(yy,mm,dd,hh,mm,ss)
```

```
var variableName= new Date(yy,mm,dd)
```

```
var variableName= new Date(milliseconds)
```

- variableName là biến dùng để lưu trữ thông tin ngày tháng năm, giờ phút giây.
- Trường hợp 1: giá trị khởi tạo là 1 chuỗi. Trong trường hợp này month là chuỗi,
- Trường hợp 2 và 3, giá trị là một số.

Ví dụ:

```
var objday =new Date("November,1,2003,7:30:9") // Khai báo hợp lệ
```

```
var objday= new Date("10,1,2003,7:30:9") //Khai báo không hợp lệ
```

**Cách 2:** Khai báo ngày hiện hành ( Không khởi tạo)

```
var variableName=new Date()
```

- Trong trường hợp này giá trị trả về là ngày tháng năm giờ phút giây hiện hành của hệ thống.

**2. Các phương thức của đối tượng Date():**

- Để truy xuất phương thức của đối tượng dùng cú pháp

```
variableName.Method()
```

Phương thức	Mô tả
Date()	trả về đối tượng date
getDate()	Trả về giá trị ngày ( số nguyên từ 1-31) trong tháng
getDay()	Trả về giá trị ngày trong tuần ( số nguyên từ 0-6 Sunday=0)
getMonth()	Trả về tháng trong năm (from 0-11. 0=January, 1=February)
getFullYear()	Trả về giá trị năm (bốn số )
getYear()	Trả về giá trị năm (hai số )
getHours()	Trả về giờ của hệ thống (từ 0-23)
getMinutes()	Trả về phút của hệ thống (từ 0-59)
getSeconds()	Trả về giây của hệ thống (từ 0-59)
getMilliseconds()	Trả về giá trị millisecond from 0-999)
setFullYear( <i>years</i> )	Thiết lập lại năm cho ngày hệ thống ( 4 số)
setHours( <i>hours</i> )	Thiết lập lại giờ cho hệ thống ( từ 0-24)
setMinutes( <i>minutes</i> )	Thiết lập lại phút cho hệ thống ( từ 0-59)
setMonth( <i>months</i> )	Thiết lập lại tháng cho hệ thống ( từ 0-11)
setSeconds( <i>seconds</i> )	Thiết lập lại giây cho hệ thống (from 0-59)

toGMTString()	Chuyển ngày giờ hệ thống sang ngày giờ quốc tế.
toString()	Chuyển ngày giờ hệ thống sang chuỗi

Ví dụ : Hiển thị giờ trên thanh trạng thái

9:12:48 P.M.

<html>

```

<head><title>Digital Clock - Status Bar</title>
<script Language="JavaScript">
var timerID = null;
var timerRunning = false;
function stopclock ()
{
    if(timerRunning)
        clearTimeout(timerID);
        timerRunning = false;
}
function showtime ()
{
    var now = new Date();
    var hours = now.getHours();
    var minutes = now.getMinutes();
    var seconds = now.getSeconds();
    var timeValue = "" + ((hours >12) ? hours -12:hours);
    timeValue += ((minutes < 10) ? ":0" : ":") + minutes
    timeValue += ((seconds < 10) ? ":0" : ":") + seconds
    timeValue += (hours >= 12) ? " P.M.": " A.M."
    window.status = timeValue;
    timerID = setTimeout("showtime()",1000);
    timerRunning = true;
}
function startclock ()
{
    stopclock();
    showtime();
}
</script>
</head>
<body BGCOLOR="#FFFFFF" TEXT="#000000" LiNK="#FF0000"
VLiNK="#000080" ALiNK="#000080"
onLoad="startclock()">
</body>

```

</html>

Ví dụ:

<html>

```

<head><title>Hiển thị ngày giờ lên trang web </title>
<script language="javascript">
function Ngay()
{
    var day=new Date();
    var w,m,d,y;
    var arrday=new Array();

```



```

    arrday[0]= "chu nhật";
    arrday[1]= "Thứ hai ";
    arrday[2]= "Thứ ba ";
    arrday[3]= "Thứ tư";
    arrday[4]= "Thứ năm ";
    arrday[5]= "Thứ sáu";
    arrday[6]= "Thứ Bảy";
    w=day.getDay();
    d=day.getDate();
    m=day.getMonth()+1;
    y=day.getFullYear();
    document.write("Hôm nay:"+arrday[w]+" ngày "+d+" tháng "+m+ " năm "+y);
  }
</script>
</head>
<body>
<script language="Javascript">Ngày()</script>
</body>
</html>

```

### XVII.2.3. Đối tượng String

Mỗi chuỗi trong JavaScript là một đối tượng, gồm các thuộc tính và phương thức thực hiện trên chuỗi, đó là các phương thức tìm kiếm chuỗi, trích chuỗi con và áp dụng các thẻ HTML vào nội dung của chuỗi.

#### 1. Cách khai báo đối tượng String

```
var stringVariable=new String()
```

#### 2. Thuộc tính của String():

- **Length**: dùng để xác định chiều dài của chuỗi. Các ký tự trong chuỗi được đánh chỉ số từ 0 đến Length-1. Tất cả các thành phần có giá trị chuỗi đều dùng được thuộc tính length. Cách tham chiếu đến thuộc tính length của đối tượng String().
  - Cách 1: StringLength=stringVariable.length
  - Cách 2:
    - var st=new Student()
    - StringLength=stName.length
  - Cách 3: StringLength="This is a string".length

#### 3. Các phương thức của String: Các phương thức của String để thực hiện các thao tác trên nội dung của chuỗi:

Phương thức	Mô tả	Ví dụ
anchor("anchormame")	Trả về một chuỗi liên kết anchormame trở thành 1 liên kết	str.anchor("anchormame") <a name="namelink">This is a string </a>
big()	Trả về một chuỗi đặt trong cặp thẻ <big>	str.big() <big>This is a string </big>

Bold()	Trả về một chuỗi in đậm	str.bold() <b> This is a string </b>
charAt(index)	Trả về ký tự thứ index trong chuỗi. index từ 0 đến str.length-1	str.charAt(0)="T"
concat()	Trả về hai chuỗi nối nhau	
fontcolor()	Trả về một chuỗi với màu đã được xác lập.	str.fontcolor("red") <font color=red>This is a string</font>
fontsize()	Trả về một chuỗi với kích thước đã được xác lập.	str.fontSize("5") <font size=5>This is a string</font>
indexOf(searchvalue, [fromindex])	Trả về vị trí của đầu tiên được tìm thấy của chuỗi searchvalue bắt đầu tìm từ vị trí fromindex. Nếu không có fromindex thì tìm từ vị trí 0. Nếu không tìm thấy thì hàm trả về giá trị -1	Pos=str.indexOf("is") Pos=2
italics()	Trả về một chuỗi in nghiêng	
lastIndexOf(searchvalue)	Trả về vị trí của cuối cùng được tìm thấy của chuỗi searchvalue bắt đầu tìm từ phải qua trái. Nếu không tìm thấy thì hàm trả về giá trị -1	
link()	Trả về một chuỗi liên kết	
match()	Tương tự như hàm indexOf và lastIndexOf, nhưng phương thức này trả về một chuỗi cụ thể nếu không tìm thấy thì trả về giá trị "null".	
replace()	Thay thế một vài ký tự bằng một vào ký tự mới	
search()	Trả về giá trị là số chuỗi được tìm thấy trong chuỗi cha, nếu không tìm thấy thì trả về giá trị -1	
slice()	Trả về một chuỗi con được cắt từ chuỗi mẹ tại vị trí cắt	
small()	Trả về một chuỗi nhỏ hơn	
strike()	Trả về một chuỗi được gạch ngang qua thân chuỗi	
sub()	Trả về một chuỗi kiểu subscript	Str.sub() <sub>This is a string</sub>
substr(start,length)	Trả về chuỗi con bắt đầu từ vị trí start và có chiều dài length. nếu không có start xem như start=0	Str.substr(0,2)="Th"
substring(Start,end)	Tách ra một chuỗi con từ một chuỗi. Bắt đầu từ chỉ số start đến end. Nếu Start<end, chuỗi trả về từ start đến end-1 Nếu end<start, chuỗi trả về từ end đến	

	start Nếu start=end chuỗi trả về là null.	
sup()	Trả về chuỗi kiểu superscript	
toLowerCase()	Chuyển chuỗi thành chữ thường	
toUpperCase()	Chuyển chuỗi thành chữ hoa	

Ví dụ: Tính chiều dài chuỗi sử dụng phương thức length

```
<script type="text/javascript">
    var str="Nguyễn Thị Hoa Hồng !"
    document.write("<p>" + str + "</p>")
    document.write("Chiều dài của chuỗi là : "+ str.length)
</script>
```

Ví dụ: Phương thức fontcolor() dùng để định màu của chuỗi

```
<script type="text/javascript">
    var txt="Nguyễn Thị Bảo Nhi "
    document.write("<p>" + txt.fontcolor() + "</p>")
    document.write("<p>" + txt.fontcolor('red') + "</p>")
    document.write("<p>" + txt.fontcolor('blue') + "</p>")
    document.write("<p>" + txt.fontcolor('green') + "</p>")
</script>
```

Ví dụ Phương thức indexOf. Phương thức này trả về vị trí của chuỗi con được tìm thấy trong một chuỗi

```
<script type="text/javascript">
    var str="This is my Schools "
    var pos=str.indexOf("School")
    if (pos>=0)
    {
        document.write("School found at position: ")
        document.write(pos + "<br />")
    }
    Else
    {document.write("School not found!")}
</script>
```

#### XVII.2.4. Đối tượng Math()

Đối tượng math() cung cấp các hàm và các phương thức cần thiết để thực hiện các phép toán số học. Không cần phải tạo đối tượng Math() mà chúng ta có thể sử dụng trực tiếp đối tượng này

##### 1. Các phương thức của Math():

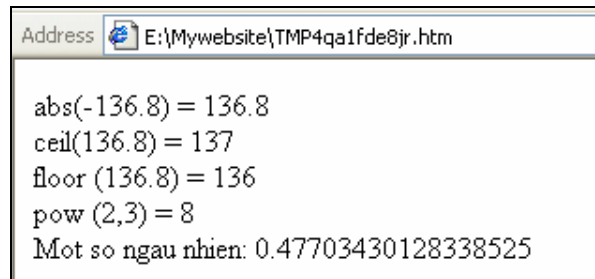
Cú pháp chung:

**Math.method([value])**

Ví dụ:

```
<script>
    var n= -136.8 , m=136.8
    Document.write("abs(-136.8) = " + Math.abs(n) + "<br>")
    Document.write("ceil(136.8) = " + Math.ceil(m) + "<br>")
    Document.write("floor (136.8) = " + Math.floor(m) + "<br>")
    Document.write("pow (2,3) = " + Math.pow(2,3) + "<br>")
```

```
Document.write("Mot so ngau nhien: " + Math.random()*5 + "<br>")
</script>
```



### – Danh sách các phương thức của Math()

Phương thức	Mô tả
abs(x)	Trả về giá trị tuyệt đối của biến x
acos(x)	Trả về giá trị arccosine của x
ceil(x)	Trả về số nguyên lớn hơn hoặc bằng x
floor(x)	Trả về số nguyên nhỏ hơn hoặc bằng x
log(x)	Trả về giá trị log của x
max(x,y)	Trả về giá trị lớn nhất trong hai số x và y
min(x,y)	Trả về giá trị nhỏ nhất trong hai số x và y
pow(x,y)	Trả về giá trị x lũy thừa y
random()	Trả về giá trị một số ngẫu nhiên từ 0 đến 1
round(x)	Làm tròn số x
sqrt(x)	Trả về giá trị căn bậc 2 của x

#### Ví dụ:

Viết chương trình tạo một nút đổi hình (play) và nút stop để ngưng  
<html><head>

```
<script>
var idq;
function play()
{
    var arrhinh= new Array();
    arrhinh[0]= "h1.jpg";
    arrhinh[1]= "h2.jpg";
    arrhinh[2]= "h3.jpg";
    arrhinh[3]= "h4.jpg";
    arrhinh[4]= "h5.jpg";
    arrhinh[5]= "h6.jpg";
    var i=Math.round(Math.random()*6)
    idhinh.src=arrhinh[i];
    idq=setTimeout("play()",1000);
}
function Stop()
{
```

```

        clearTimeout(idq);
    }
</script>
</head>
<body>

<form>
    <input type=button value= "Play" onClick= "Play()">
    <input type=button value= "Stop" onClick= "Stop()">
</form></body></html>

```

### XVII.2.5. Đối tượng 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

#### 1. Các thuộc tính của đối tượng document

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
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 xem của tài liệu

#### 2. Phương thức:

Phương thức	Mô tả
clear()	Xóa tài liệu
close()	Đóng một tài liệu
focus()	Đưa trở về một đối tượng trong trang
open("mimetype",replace)	
write("str")	viết một chuỗi vào một tài liệu
writeln("str")	viết một chuỗi vào một tài liệu và xuống dòng

#### 3. Sự kiện

- Cú pháp:

**document.event\_name="someJavaScriptCode"**

**– Danh sách các sự kiện tác động đối tượng document**

Event  
 OnBlur  
 OnClick  
 OnDblClick  
 OnFocus  
 OnKeyDown  
 OnKeyPress  
 OnKeyUp  
 OnMouseDown  
 OnMouseMove  
 OnMouseOut  
 OnMouseOver  
 OnMouseUp  
 OnMouseUp

**XVII.2.6. Đối tượng trình duyệt (Navigator Object)**

Đối tượng trình duyệt chứa đựng những thông tin về trình duyệt web của client Có hai trình duyệt web lớn là Netscape Navigator và internet Explorer. Mặc dù cả hai đều hỗ trợ mô hình đối tượng trên ngôn ngữ Javascript nhưng cũng có một số đối tượng và cách truy cập vào thành phần thuộc tính trên hai trình duyệt cũng có một cái khác nhau. Muốn cho ứng dụng chạy hoàn chỉnh trên mọi trình duyệt thì người lập trình phải xác định ra chương trình đang chạy trên trình duyệt nào và ở version nào để xử lý đoạn code tốt hơn

**1. Thuộc tính**

Thuộc tính	Mô tả
appName	Tên trình duyệt
appVersion	Phiên bản trình duyệt
cookieEnabled	
platform	Nền của hệ điều hành

**2. Phương thức**

Phương thức	Mô Tả
javaEnabled()	trả về giá trị true nếu trình duyệt có hỗ trợ Javascript
refresh()	
preference()	

**XVII.2.7. Đối tượng Window**

Là đối tượng cao nhất trong mô hình DOM, là nơi chứa tất cả các thành phần của trang web.

## 1. Thuộc tính của đối tượng Window:

Thuộc tính	Mô tả	Giá trị
defaultStatus	thiết lập chuỗi t/báo trên thanh trạng thái	Text
status	thiết lập thông báo tại thời điểm hiện hành	Text
location	Xác định vị trí trang hiện tại trong cửa sổ	URL
history	Xác định các phần tử trong history	
alwaysLowered	hiển thị cửa sổ bên dưới các cửa sổ khác	Yes/no
alwaysRaised	hiển thị cửa sổ trên tất cả các cửa sổ khác	Yes/no
Dependent	Cửa sổ này sẽ đóng khi cửa sổ cha bị đóng	Yes/no
directories	Hiển thị Button thư mục	Yes/no
fullscreen	hiển thị chế độ đầy màn hình	Yes/no
height	thiết lập chiều cao của cửa sổ	số nguyên
hotkeys	Cho phép dùng phím nóng	Yes/no
left	Thiết lập k/cách từ văn bản đến cạnh cửa sổ	số nguyên
location	hiển thị hộp location	Yes/no
menubar	hiển thị thanh menu bar	Yes/no
resizable	Cho phép thay đổi kích thước cửa sổ	Yes/no
scrollbars	xuất hiện /không xuất hiện thanh cuộn	Yes/no
status	Hiển thị thanh trạng thái	Yes/no
titlebar	hiển thị thanh tiêu đề	Yes/no
toolbar	hiển thị thanh công cụ	Yes/no
width	Xác định độ rộng của cửa sổ	số nguyên
closed	trả về giá trị true, false. True khi cửa sổ đóng	true, false

### **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"
```

## 2. Phương Thức

### **Cú pháp: window.method\_name()**

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

MoveTo(x,y)	Di chuyển cửa sổ qua trái và lên trên một đoạn pixel cụ thể so với cửa sổ hiện hành.
open(URL,"windowname","FeatureList")	URL : là địa chỉ trang web muốn nạp vào cửa sổ. WindowName: là tên cửa sổ . FeatureList : là danh sách các thuộc tính của cửa sổ: thanh công cụ, thanh menu, thanh status ....
print()	in nội dung trong cửa sổ hiện hành.
prompt("msg","reply")	Hiện thị hộp thoại nhập liệu
setTimeout(func,millisec)	Thiết lập thời gian mili giây để gọi một hàm
stop()	Hủy việc download một cửa sổ. Tương tự như việc đóng một cửa sổ trình duyệt.
resizeBy(dx,dy)	Thay đổi kích thước cửa sổ sang phải dx, dưới dy pixel
resizeTo(x,y)	Thay đổi kích thước x, y pixel
scrollBy(dx,dy)	cuộn nội dung sang phải dx, xuống dưới dy pixel
scrollTo(x,y)	cuộn nội dung trên trang đến vị trí x,y

Ví dụ: Objwindow.close()

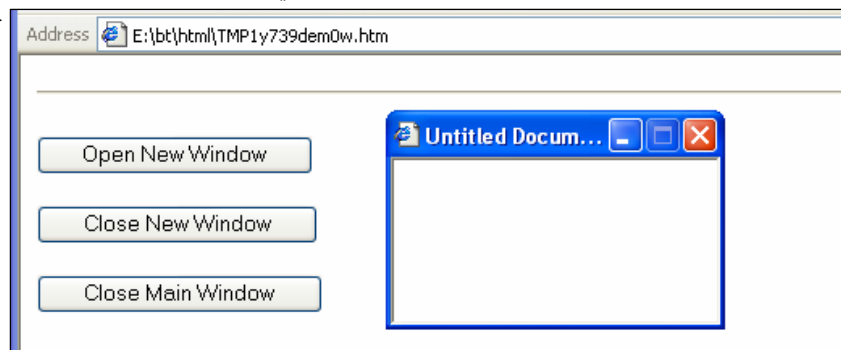
- ❖ **Từ khoá self**: trong trường hợp muốn thao tác trên cửa sổ hiện hành ta dùng từ khoá self thay thế cho đối tượng window

Ví dụ : đóng cửa sổ hiện hành:

**Self.close() hoặc window.close()**

Ví dụ: Đoạn Script đóng, mở cửa sổ trình duyệt

```
<html>
  <head></head>
  <body>
    <FORM NAME="winform">
      <INPUT TYPE="button" VALUE="Open New Window"
        onClick="NewWin=window.open('blank1.htm','NewWin',
        'toolbar=no,status=no,width=200,height=100'); ">
      <P><INPUT TYPE="button" VALUE="Close New Window"
        onClick="NewWin.close();" >
      <P><INPUT TYPE="button" VALUE="Close Main Window"
        onClick="window.close();">
    </FORM>
  </body>
</html>
```





Ví dụ: Viết hàm mở một cửa sổ mới khi click nút điều khiển

```
function openwindow()
{
    window.open("http://www.yahoo.com","my_new_window",
        "toolbar=yes, location=yes, directories=no, status=no, menubar=yes,
        scrollbars=yes, resizable=no, copyhistory=yes, width=400, height=400")
}
```

### 3. Sự kiện

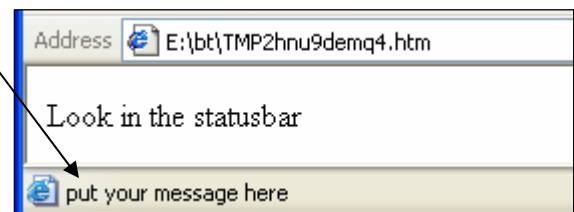
Sự kiện	Mô tả
onBlur	Thực thi đoạn mã lệnh khi sự kiện blur xảy ra
onError	Thực thi đoạn mã lệnh khi sự kiện Error xảy ra
onFocus	Thực thi đoạn mã lệnh khi sự kiện Focus xảy ra
onLoad	Thực thi đoạn mã lệnh khi sự kiện Load xảy ra
onResize	Thực thi đoạn mã lệnh khi sự kiện resizer xảy ra
onUnload	Thực thi đoạn mã lệnh khi sự kiện Unload xảy ra

Ví dụ: Viết hàm trở tới một trang web khác

```
function locate()
{
    location="http://www.yahoo.com/"
}
```

Ví dụ: Hiện thị một số thông tin lên thanh trạng thái

```
<head>
    <script type="text/javascript">
        function load()
        {window.status = "put your message here"}
    </script>
</head>
<body onload="load()">
    <p>Look in the statusbar</p>
</body>
```



Ví dụ: Viết hàm in trang web:

```
function printpage()
{
    window.print()
}
```

#### XVII.2.8. Đối tượng form:

Form là một thành phần trên trang web dùng để thu thập dữ liệu, thông tin từ người dùng. Mỗi phần tử trong form là một đối tượng trong DOM. Do đó mỗi phần tử trên form cũng có những sự kiện.

## 1. Các sự kiện của các phần tử trên form

Phần tử	Tên sự kiện
Button	onClick
Checkbox	onClick
Form	OnSubmit, onReset
Textbox	OnBlur,OnChange,OnFocus,Onselect
Radio	OnClick
Reset button	OnClick
Dropdown menu	OnBlur,onChange,onFocus,onSelect
Submit button	OnClick
Textarea	OnBlur,OnChange,OnFocus,Onselect

## 2. Truy cập giá trị các phần tử trên form

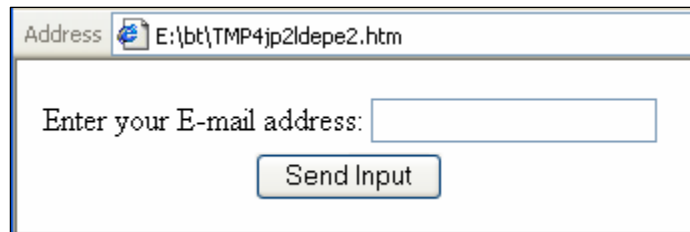
### Cú pháp:

**document.formName.formelement.properties**

**document.formName.formelement.method**

Ví dụ 1: Tạo một form chứa một field để nhập địa chỉ email. Kiểm tra dữ liệu nhập vào có phải là địa chỉ E-mail không bằng cách kiểm tra ký tự @ trong địa chỉ nhập vào

```
<html>
<head><Title> Kiem tra</title></head>
<body>
  <form name="form1" method="post" action="">
    <table>
      <tr>
        <td>Enter your E-mail address: </td>
        <td><input type="text" name="MyEmail"></td>
      </tr>
      <tr>
        <td align="center" colspan="2">
          <input type="submit" name="Submit" value="Send Input"
            onclick= "validate()">
        </td>
      </tr>
    </table>
  </form>
  <script>
    function validate()
    {
      ad= document.form1.MyEmail.value.indexOf("@")
      if (ad == -1)
      {
        alert("Not a valid e-mail")
        return false
      }
    }
  </script>
</body></html>
```



**Ví dụ 2:** Kiểm tra tính hợp lệ của giá trị nhập vào textfield

```
<html>
  <head>
    <script>
      function validate()
      {
        txt= document.myForm.myinput.value
        if (txt>=1 && txt<=5)
        {
          return true
        }
        else
        {
          alert("Must be between 1 and 5")
          return false
        }
      }
    </script>
  </head>
  <body>
    <form name="myForm" onsubmit="return validate()">
      Enter a value from 1 to 5:
      <input type="text" name="myinput">
      <input type="submit" value="Send input">
    </form>
  </body>
</html>
```

**Ví dụ 3:** Tạo một form chứa 1 field để nhập giá trị. Viết một hàm dùng để kiểm tra số ký tự nhập vào field này ( dùng thuộc tính length để kiểm tra số ký tự nhập vào)

```
<html>
  <head>
    <script language="javascript">
      function validateform()
      {
        input= document.myForm.myinput.value
        if (input.length>5)
        {
          alert("Do not insert more than 5 characters")
          return false
        }
        else { return true}
      }
    </script></head>
```

```

<body>
  <form name="myForm" onsubmit="return validateform()">
    in this input box you are not allowed to insert more than 5 characters:
    <input type="text" name="myinput">
    <input type="submit" value="Send input">
  </form>
</body>
</html>

```

### 3. Các thuộc tính trên đối tượng form:

Thuộc tính	Mô tả	Ví dụ
Action	Trả về đường dẫn (URL) đến tập tin xử lý của form thứ i	Document.forms[i].action
Length	Trả về số form trên trang web Hoặc trả về số phần tử trên form thứ i	Countform=document.forms.length Countfield=document.forms[i].length
Name	Trả về giá trị tên của form thứ i	Nameform=document.forms[i].name
Method	Các định phương thức của form thứ i	Methodform=document.forms[i].method
elements	mảng element chứa các phần tử trên form	document.form[i].elements[j].value

### 4. Các thuộc tính trên mảng element

Thuộc tính	Mô tả	Ví dụ
Name	Xác định tên của một phần tử trên form thứ i.	document.form[i].elements[j].name
Type	Xác định loại của đối tượng	document.form[i].elements[j].type
Value	Xác định giá trị của phần tử thứ j trong form i.	document.form[i].elements[j].value
Checked	Xác định phần tử thứ j có được checked không. Nếu có trả về giá trị true còn không trả về giá trị false	document.form[i].elements[j].checked
Disable	Thiết lập chế độ mờ ( gán giá trị true không cho phép người sử dụng chọn lựa và ngược lại	document.form[i].elements[j].disable
isDisable	Kiểm tra phần tử có mờ không (true là mờ và ngược lại)	document.form[i].elements[j].isDisable
readOnly	Cho phép/không thay đổi nội dung của phần tử	document.forms[i].elements[j].readOnly

### 5. Phương thức trên mảng element:

Phương thức	Mô tả	Ví dụ
Focus ()	Đưa con trỏ về lại text box hoặc dropdownmenu	document.form[i].elements[j].focus()

Lưu ý: Nếu ta đang làm việc trên document hiện hành, biết tên cụ thể của form và tên của thành phần trên form ta có thể truy cập trực tiếp mà không cần qua mảng form và element:

**nameForm.nameField.property**

hoặc

**nameForm.nameField.method**

Ví dụ 1: về set focus trên một field.

```
<html>
  <head>
    <script type="text/javascript">
      function setfocus()
      {
        document.forms[0].field.focus()
      }
    </script>
  </head>
  <body>
    <form>
      <input type="text" name="field" size="30">
      <input type="button" value="Get Focus" onclick="setfocus()">
    </form>
  </body>
</html>
```

Ví dụ 2: Viết một hàm để đưa con trỏ về textbox đã tạo trước ( sử dụng phương thức focus() )

```
<html>
  <head>
    <script language="javascript">
      function setfocus()
      {
        document.forms[0].field.select()
        document.forms[0].field.focus()
      }
    </script>
  </head>
  <body>
    <form>
      <input type="text" name="field" size="30" value="input text">
      <input type="button" value="Selected" onclick="setfocus()">
    </form>
  </body>
</html>
```

## XVII.2.9. Các phần tử trên form:

### 1. Thao tác trên trường radio

Muốn lấy giá trị của trường radio ta phải sử dụng đến mảng element. Duyệt qua tất cả các phần tử và kiểm tra phần tử đó có được checked không ?

**Cú pháp:**

```

Countfield=nameform.length
for(var i=0;i<Countfield;i++)
if(nameform.elements[i].type== "radio" &&nameform.elements[i].checked==true)
Giatri=nameform.elements[i].value

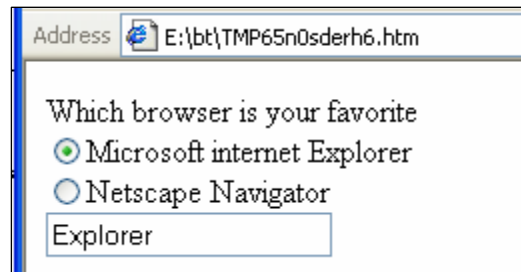
```

Ví dụ 1: Tạo form có chứa các radio:

```

<html>
  <head>
    <script type="text/javascript">
      function check(browser)
      {
        document.forms[0].answer.value=browser
      }
    </script>
  </head>
  <body>
    <form>
      Which browser is your favorite<br>
      <input type="radio" name="browser" onclick="check(this.value)"
      value="Explorer">Microsoft internet Explorer<br>
      <input type="radio" name="browser" onclick="check(this.value)"
      value="Netscape">Netscape Navigator<br>
      <input type="text" name="answer">
    </form></body></html>

```



Ví dụ 2:

```

<html>
  <head>
    <script type="text/javascript">
      function check()
      {
        coffee=document.forms[0].coffee
        answer=document.forms[0].answer
        txt=""
        for (i = 0; i<coffee.length; ++ i)
        {
          if (coffee[i].checked)
            {txt=txt + coffee[i].value + " "}
        }
        answer.value="You ordered a coffee with " + txt
      }
    </script>
  </head>
  <body>

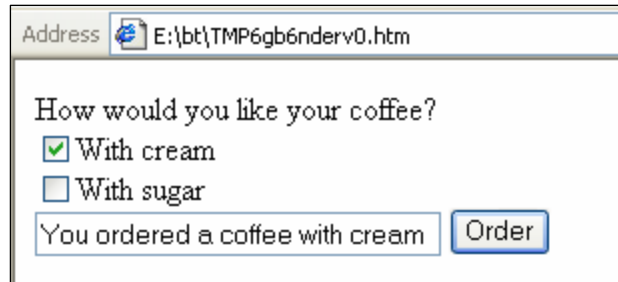
```

```

<form>
How would you like your coffee?<br>
<input type="checkbox" name="coffee" value="cream">With cream<br>
<input type="checkbox" name="coffee" value="sugar">With sugar<br>

<input type="text" name="answer" size="30">
<input type="button" name="test" onclick="check()" value="Order">
</form>
</body>
</html>

```



## 2. Thao tác trên dropdownmenu

Các thuộc tính và phương thức của dropdownmenu

### a) Thuộc tính

Thuộc tính	Mô tả	Ví dụ
length	Trả về số phần tử trong danh sách dropdownmenu.	spt=nameform.namefield.length
selectedIndex	trả về chỉ số của phần tử được chọn trong danh sách	spt=nameform.namefield.selectedIndex
options	mảng option chứa các phần tử trong danh sách được đánh chỉ số 0->spt-1	

### b) Phương thức

Phương thức	Mô tả	Ví dụ
Focus()	Đưa con trỏ về lại dropdownmenu	nameform.namefield.focus()

### c) Các thuộc tính của mảng option

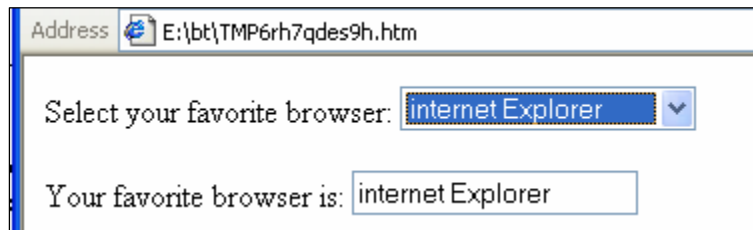
Thuộc tính	Mô tả	Ví dụ
DefaultSelected	Trả về giá trị true nếu phần tử thứ i được chọn	nameform.namefield.option[i].defaultSelectedfocus()
Selected	Trả về giá trị true nếu phần tử thứ i được chọn	nameform.namefield.option[i].selected
Value	Trả về giá trị value của option thứ i.	nameform.namefield.option[i].value
Text	Trả về giá trị text của option thứ i.	nameform.namefield.option[i].text

**Ví dụ 1:**

```

<html>
  <head>
    <script type="text/javascript">
      function put()
      {
        txt=document.forms[0].dropdown.options[document.forms[0].
        dropdown.selectedIndex].text
        document.forms[0].favorite.value=txt
      }
    </script>
  </head>
  <body>
    <form>
      Select your favorite browser:
      <select name="dropdown" onchange="put()">
        <option>internet Explorer
        <option>Netscape Navigator
      </select>
      <p>
      Your favorite browser is:
      <input type="text"
      name="favorite" value="internet Explorer">
    </form>
  </body>
</html>

```

**Ví dụ 2** Tạo form có dropdown menu

```

<html>
  <head>
    <script type="text/javascript">
      function put()
      {
        option=document.forms[0].dropdown.options[document.forms[0].
        dropdown.selectedIndex].text
        txt=document.forms[0].number.value
        txt=txt + option
        document.forms[0].number.value=txt
      }
    </script>
  </head>
  <body>
    <form>
      Select numbers:<br>

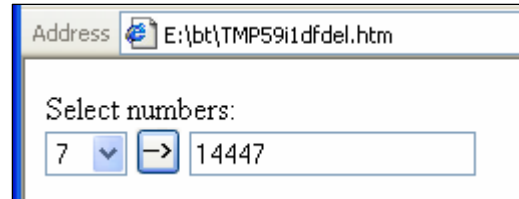
```



```

<select name="dropdown">
  <option>1
  <option>2
  <option>3
  <option>4
  <option>5
  <option>6
  <option>7
  <option>8
  <option>9
  <option>10
</select>

```



```

<input type="button" onclick="put()" value="-->" > <input type="text"
name="number">
</form>

```

```
</body>
```

```
</html>
```

### **Ví dụ 3:**

```
<html>
```

```
<head></head>
```

```
<body>
```

```
<script language="JavaScript">
```

```
var max=0;
```

```
function textlist()
```

```
{
```

```
max=textlist.arguments.length;
```

```
for (i=0; i<max; i++)
```

```
this[i]=textlist.arguments[i];
```

```
}
```

```
tl=new textlist("KHOA CONG NGHE THONG TIN", "TRUONG DAI HOC  
CONG NGHIEP TPHCM","SO 12 NGUYEN VAN BAO ","DIEN THOAI:  
8940390")
```

```
var x=0; pos=0;
```

```
var l=tl[0].length;
```

```
function textticker()
```

```
{
```

```
document.tickform.tickfield.value=tl[x].substring(0,pos)+"_";
```

```
if(pos++==l)
```

```
{
```

```
pos=0; setTimeout("textticker()",1000);
```

```
x++;
```

```
if(x==max) x=0; l=tl[x].length;
```

```
}
```

```
else
```

```
iq=setTimeout("textticker()",50);
```

```
}
```

```
function stop()
```

```
{
```

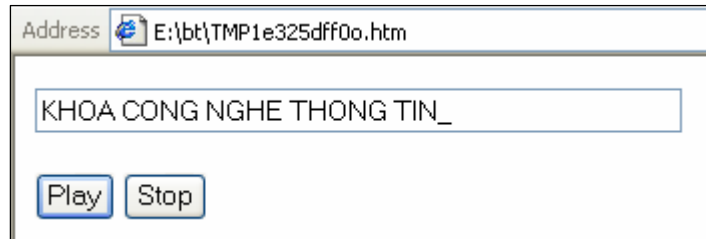
```
clearTimeout(iq)
```

```
}
```

```

</script>
<form action="" method="post" name="tickform" id="tickform">
  <input name="tickfield" type="text" id="tickfield" size="50">
  <p>
    <input type="button" name="Button" value="Play" onClick="textticker()">
    <input type="button" name="Button" value="Stop" onClick="stop()">
  </p>
</form>
</body>
</html>

```



### XVII.3. THAY ĐỔI NỘI DUNG ĐỘNG TRÊN TRANG

#### XVII.3.1. Trên Netscape:

Để thay đổi nội dung động trên trang dùng cặp tag <Layer> </Layer>

```

<Layer ID=IdName properties>
  Document content
</Layer>

```

*ID* : là tên của Layer, dựa vào IdName để thay đổi nội dung động trên trang,  
*Properties*: là danh sách các thuộc tính của Layer như xác định vị trí xuất hiện hay liên kết đến trang web nào đó.

#### Danh sách các thuộc tính:

Thuộc tính	Mô tả
Clip=top_x, left_y,bottom_x,right_y	Xác định tọa độ xuất hiện của layer trong cửa sổ
Height=value	Xác định chiều cao của layer
Left=value	Qui định khoảng trắng trái từ văn bản đến layer
PageX=value	Xác định khoảng cách của layer so với cạnh trên của cửa sổ
PageY=value	Xác định khoảng cách của layer so với cạnh bên của cửa sổ
SRC=URL	Xác định tập tin nạp vào layer
Top=value	Qui định khoảng trắng trên từ văn bản đến layer
Visibility=option(Hide show)	Xác định layer xuất hiện (show) hay không xuất hiện (hide)
Width=value	Xác định chiều rộng layer
Z-index=value	Vị trí tương đối của layer so với các phần tử khác

Ví dụ:

```
<layer ID=Idlayer SRC="filename.htm"></layer>
```

Trong một trang ta có thể viết một hoặc nhiều layer. Như vậy dựa vào thuộc tính SRC để có thể nạp vào một hoặc nhiều trang web khác nhau .

#### XVII.3.2. Trên Internet Explorer

Tag <Iframe></Iframe> trên Internet Explorer, có công dụng giống như Layer trên NetsCape.

```
<Iframe Id=IdName properties>
</Iframe>
```

### Các thuộc tính của Iframe

Thuộc tính	Mô tả
Align={left,canter,right}	Xác định vị trí xuất hiện của của iframe
frameborder	Xác định đường viền
Height=value	Xác định chiều cao
SRC	Xác định địa chỉ trang web nạp vào iframe
width	Xác định chiều rộng của iframe

- Nếu như nội dung trong iframe quá lớn thì tự nó tự động xuất hiện thanh trượt để cuộn nội dung.
- Viết nội dung vào iframe

**Idname.document.write(content)**

- Thay đổi nội dung trên trang iframe

**Document.all.idName.SRC="URL"**

Ví dụ :

```
<iframe id=page frameborder=0 src= "page2.htm"></iframe>
```

### XVII.3.3. Thay đổi nội dung trên trang dựa vào inner và outer

Ta dùng đặc tính inner và outer để thay đổi nội dung hoặc lấy giá trị của một vùng nào đó trên trang web.

#### 1. Phân biệt giữa inner và outer

- Inner là những gì chứa bên trong của đối tượng chứa ID. Inner gồm có
  - o InnerHTML lấy nội dung text và tag HTML bên trong đối tượng ID
  - o innerText: chỉ lấy nội dung text bên trong đối tượng ID
- Outer là phần inner và bản thân đối tượng chứa ID. Outer gồm có
  - o outerHTML lấy nội dung text và tag HTML của cả đối tượng ID
  - o outerText : lấy nội dung text

Ví dụ:

```
<Div ID=Intro>Monitor<B> SAMSUNG</B></Div>
```

#### 2. Lấy dữ liệu từ một đối tượng

- Nếu lấy dữ liệu từ một vùng nào đó thì innerText và outerText hoàn toàn giống nhau, chỉ lấy phần văn bản không lấy các thẻ HTML bao quanh chúng

**Ví dụ:**

```
Var s1,s2
s1=Intro.outerText
s2=Intro.innerText
thì s1 và s2 đều nhận giá trị Monitor SAMSUNG
```

**Ví dụ**

```
s1=Intro.outerHTML
s2=Intro.innerHTML
```

Thì s1=<Div Id=Intro>Monitor<B> SAMSUNG</B></Div>

Và s2=Monitor<B> SAMSUNG</B>

- Trong trường hợp ta viết nội dung nó ra ngoài thì nó sẽ định dạng những thẻ HTML chứa trong phần nội dung

**Ví dụ:**

**Intro.innerHTML="<I>CPU Pentium IV"</I>**

Khi đó vùng Intro sẽ được thay thế bằng chuỗi *CPU Pentium IV* Chứa trong thẻ <DIV>

**<Div Id=Intro>CPU Pentium IV</Div>**

**Intro.outerHTML="<I>CPU Pentium IV"</I>**

Khi đó vùng Intro sẽ được thay thế bằng chuỗi *CPU Pentium IV* chứa trong thẻ <DIV> đồng thời thẻ <DIV> không còn trên trang. Do đó nếu chương trình tiếp tục xử lý các lệnh còn liên quan đến ID có tên là Intro thì nó sẽ báo lỗi. Vì chuỗi *CPU Pentium IV* thay thế cho <Div Id=Intro>Monitor<B> SAMSUNG</B></Div>

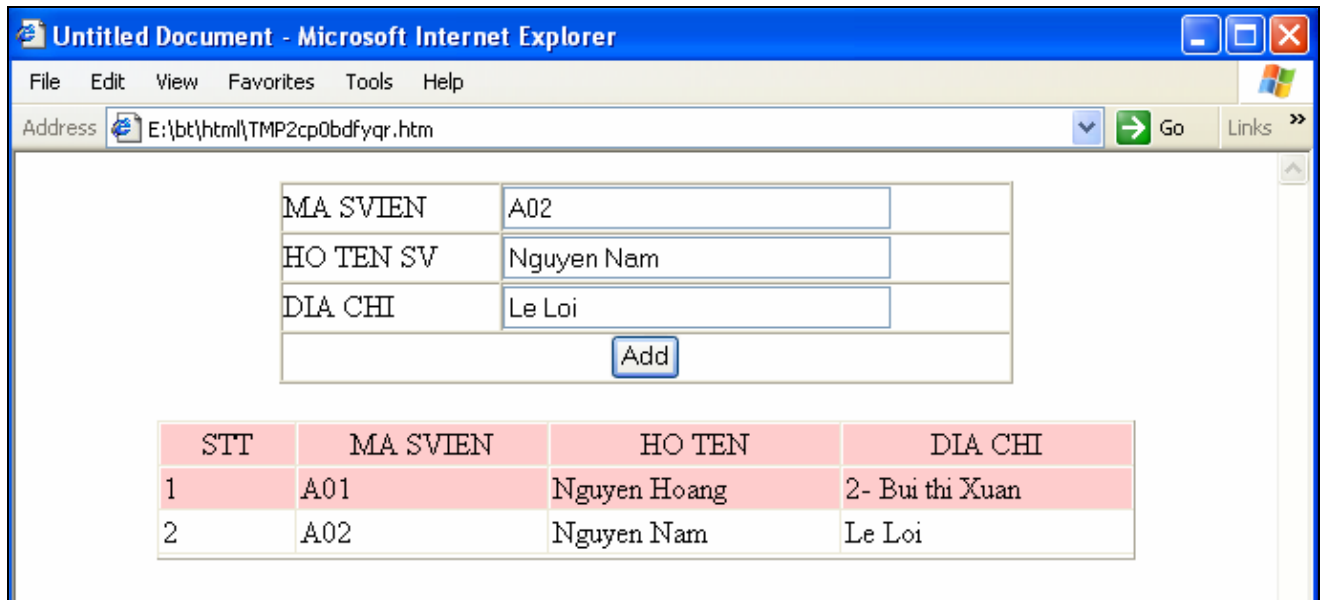
**Ví dụ:** Dùng Inner Outer

```
<html>
  <head><title>Untitled Document</title>
    <script language="JavaScript">
      var count;
      count=1;
      function Add()
      {
        stt.innerText=count;
        ma.innerText=form1.msv.value;
        ten.innerText=form1.tsv.value;
        dc.innerText=form1.dcsv.value;

        r.id="r" + count;
        stt.id="stt"+count;
        ma.id="ma"+count;
        ten.id="ten"+count;
        dc.id="dc"+count;

        pg=t.innerHTML
        pd='<table id=t border="1" cellspacing="1" style=" BORDER-
        COLLAPSE: collapse" align="center" width=80%>'
        pt='<tr id=r>'
        pt=pt+'<td id=stt width="14%"></td>'
        pt=pt+'<td id=ma width="26%"></td>'
        pt=pt+'<td id=ten width="30%"></td>'
        pt=pt+'<td id=dc width="30%"></td>'
        pt=pt+'</tr>'
        pc="</table>"
        t.outerHTML=pd+pg+pt+pc
        count++
      }
    </script>
  </head>
```

```
<body>
  <form name="form1" method="post" action="">
    <table width="60%" border="1" align="center" cellpadding="0"
      cellspacing="0">
      <tr>
        <td width="30%">MA SVIEN</td>
        <td width="70%"><input name="msv" type="text"
          id="msv" SIZE=30></td>
      </tr>
      <tr>
        <td>HO TEN SV</td>
        <td><input name="tsv" type="text" id="tsv"
          SIZE=30></td>
      </tr>
      <tr>
        <td>DIA CHI</td>
        <td><input name="dcsv" type="text" id="dcsv"
          SIZE=30></td>
      </tr>
      <tr>
        <td colspan="2"><div align="center">
          <input type="button" name="Button" value="Add"
            onClick="Add()"></div>
        </td>
      </tr>
    </table>
  </form>
  <table id="t" width="75%" border="1" align="center" cellpadding="0"
    cellspacing="0" align="center">
    <tr bgcolor="#FFCCCC">
      <td WIDTH=14%><div align="center">STT</div></td>
      <td WIDTH=26%><div align="center">MA SVIEN</div></td>
      <td WIDTH=30%><div align="center">HO TEN</div></td>
      <td WIDTH=50%><div align="center">DIA CHI</div></td>
    </tr>
    <tr id="r" bgcolor="#FFCCCC">
      <td id="stt" width="14%"></td>
      <td id="ma" width="26%"></td>
      <td id="ten" width="30%"></td>
      <td id="dc" width="50%"></td>
    </tr>
  </table>
</body>
</html>
```



**Ví dụ tham khảo :** Thiết kế form bán vé tàu

<HTML>

<HEAD><TITLE></TITLE>

<script>

```
var t1="",t2="",t3="",t4="";
```

```
var objw;
```

```
focus();
```

```
function nhap()
```

```
{
```

```
var i;
```

```
if(DK.T1.value=="")
```

```
{
```

```
    window.showModalDialog("massege.htm","Phai nhap vao ho ten","status=no;help=no;scrollbar=no")
```

```
    DK.T1.focus();
```

```
    return;
```

```
}
```

```
if(DK.D1.options[DK.D1.selectedIndex].text==DK.D2.options[DK.D2.selectedIndex].text)
```

```
{
```

```
    window.showModalDialog("massege.htm","Noi di khong duoc trung noi den","status=no;help=no;scrollbar=no");
```

```
    return;
```

```
}
```

```
if(DK.T2.value=="")
```

```
{
```

```
    window.showModalDialog("massege.htm","Phai nhap vao gia tien","status=no; help=no;scrollbar=no")
```

```
    DK.T2.focus();
```

```
    return;
```

```
}
```

```

if (isNaN(DK.T2.value )==true )
{
    window.showModalDialog("massege.htm","Gia tri phai co kieu so",
    "status=no; help=no;scrollbar=no");
    DK.T2.value="";
    DK.T2.focus();
    return;
}
objw=window.open("danh sach dang ky.htm","Danh Sach Dang Ky")
t1= t1 + DK.T1.value + "<br>" ;
objw.c1.innerHTML = t1
i=DK.D1.selectedIndex ;
t2=t2+DK.D1.options[i].text+"<br>";
objw.c2.innerHTML=t2;

i=DK.D2.selectedIndex ;
t3=t3+DK.D2.options[i].text+"<br>";
objw.c3.innerHTML=t3;
gia= eval(DK.T2.value);
if(DK.co.checked)
{
    t4 = t4 + gia*2*0.8 + "<br>"
    objw.c4.innerHTML= t4
}
else
{
    t4 = t4 + gia + "<br>"
    objw.c4.innerHTML= t4
}
blur();
objw.focus();
}
</script>
</HEAD>
<BODY>
<FORM method="post" name="DK" >
<TABLE border="1" width="79%">
<TR>
<TH colspan="2">
<font size="5" face="Arial, Helvetica, sans-serif">
DANG KY VE TAU
</font>
</TH>
</TR>
<TR>
<TD width="54%">Ho ten khách hàng: </TD>
<TD width="46%"><INPUT name="T1" ></TD>
</TR>

```





## CHƯƠNG XVIII: HIỆU ỨNG FILTER VÀ TRANSITION

### XVIII.1. FILTER :

Filter là hiệu ứng đặc biệt được áp dụng cho một đối tượng trong trang web để thay đổi sự thể hiện của nó. Hiệu ứng này chỉ được Internet Explorer hỗ trợ, Netscape nó không được hỗ trợ.

Filter chỉ được áp dụng cho các thành phần được tạo với các tag : <div>, <span>. Cần phải xác định độ rộng, độ cao hoặc có vị trí tuyệt đối (width, height, position). <img>, <marquee>, <button>, <input>, <textarea>, <table>, <tr>, <td>, <th>, <thead>, <tfoot>.

Một số thể hiện của một đối tượng là hình ảnh khi áp dụng các loại filter.

|   |  |  |  |   |
|---|--|--|--|---|
| <br>Hình gốc (1) | <br>alpha (2) | <br>Blur (3)      | <br>chroma | <br>dropshadow |
| <br>flipV      | <br>flipH   | <br>glow        | <br>gray | <br>invert   |
| <br>light (16) | <br>Mask    | <br>Shadow (18) | <br>Wave | <br>xray     |

BẢNG I

Để áp dụng một filter ta có thể có thể dùng CSS (cascading style sheet) hoặc dùng chương trình javascript.

#### XVIII.1.1. Áp dụng filter bằng CSS :

**Cú pháp:****Filter:filter\_name(parameters).**

- Filter\_name là tên một trong những filter.
- Parameters xác định những giá trị tham số của filter.

Một filter có thể không có hoặc có nhiều tham số, khi có nhiều tham số mà không chỉ định giá trị cụ thể thì coi như chấp nhận giá trị mặc định. Bảng sau đây mô tả tên filter và các tham số :

| Tên filter | Tham số   | Mô tả   |
|------------|---|---|
| Alpha      | Opacity : 0-100<br>Style : 0,1,2,3                                | Bộ lọc trong suốt, tham số opacity từ 0 (trong suốt) tới 100 (bình thường). Style là kiểu lọc   |
| Blur       | Direction : 0-360<br>Strength : 1-100<br>Add :0/1 (cộng hình góc) | Bộ lọc nhoè, direction là chiều làm nhoè tính theo góc, 0 là chiều hướng lên trên, 90 là chiều sang phải, strength chỉ độ nhoè mạnh hay yếu |
| Chroma     | Color : #rrggbb   | Làm cho màu chỉ định bởi color bị mất đi  |
| DropShadow | Color : #rrggbb<br>OffX, OffY                                     | Tạo bóng đổ xuống mặt phẳng phía dưới, OffX và OffY tính bằng pixel là chênh lệch theo toạ độ x và y của hình gốc và bóng đổ                |
| FlipH      | Không   | Lật hình theo chiều ngang   |
| FlipV      | Không   | Lật hình theo chiều đứng  |
| Glow       | Color : #rrggbb<br>Strength : 1-255                               | Tạo quang sáng quanh đối tượng, độ rộng chỉ số pixel được xác định bởi Strength, màu được xác định bởi Color.                               |
| Gray       | Không   | Thể hiện đối tượng theo thang độ xám  |
| Invert     | Không   | Đảo ngược màu   |
| Light      | Nhiều tham số   | Tạo các nguồn sáng để chiếu sáng đối tượng  |
| Mask       | Color : #rrggbb   | Tạo mặt nạ với màu xác định bởi Color   |
| Shadow     | Direction : 0-360<br>Color= #rrggbb                               | Tạo bóng đổ cho đối tượng như kiểu chữ bóng đổ thông thường.color định màu cho bóng đổ  |
| Wave       | Nhiều tham số   | Biến dạng đối tượng theo dạng sóng sin  |
| Xray       | Không   | Làm cho đối tượng có hình ảnh như ảnh của film X quang  |

Netscape không hỗ trợ filter, nên khi gặp những thuộc tính trong CSS mà nó không nhận biết, nó sẽ bỏ qua tất cả các thuộc tính trong CSS áp dụng cho đối tượng. Để giải quyết vấn đề này ta có thể áp dụng filter bằng cách sử dụng chương trình javascript.

Ví dụ: Hình 2 và 3 của bảng I được tạo bằng CSS như sau :

```
<img border = "0" src = "van.gif" width = "110" height= "120" style= "filter: alpha
(opacity = 90, style = 2)">

```

**XVIII.1.2. Áp dụng filter dùng javascript :****Cú pháp:**

**object.style.filter = “filter\_name(parameters)”**

- *object* : tên của của đối tượng trong trang.

Nếu muốn kết hợp nhiều filter trong trang, ta phải phải phân cách chúng với nhau bằng dấu chấm phẩy (điều này không cần thiết khi dùng CSS):

Ví dụ:

```
object.style.filter = "shadow ; alpha(opacity=30)" .
```

Ta có thể sử dụng cú pháp khác, xem filter như một phần của mô hình đối tượng hồ sơ (DOM: document object model), cho phép xem các tham số như là thuộc tính của đối tượng filter. Cú pháp tổng quát:

```
object.filters.filter_name.filter_parameter = value.
```

- *object* : là tên của đối tượng
- *filter\_name* : là tên của filter áp dụng cho đối tượng.
- *filter\_parameter* : là tên của một trong những tham số của đối tượng.

Ví dụ:

```
<div id=logo style="position:absolute; filter:dropShadow(color:#ff0000 offx=5 offy=5)"> TIN NHANH </div>
```

Có thể dùng script để biến đổi màu của bóng đổ từ màu đỏ sang màu xanh như sau :

```
logo.filters.DropShadow.Color = "#0000ff" ;
```

- Chú ý : để lệnh này không gây ra lỗi ta phải áp dụng filter Dropshadow cho đối tượng trước, nếu không sẽ trả về lỗi bởi vì filter không được xem như là một phần của đối tượng DOM.
- Một đối tượng có thể chứa nhiều filter. Cú pháp **logo.filters** truy xuất tới tập hợp filter của đối tượng, theo javascript để truy xuất tới phần tử (chẳng hạn DropShadow) trong đối tượng tập hợp ta có có nhiều cách viết tương đương :

```
logo.filters.DropShadow
logo.filters["DropShaDow"]
logo.filters[0]
```

- Như đã đề cập ở trên về việc không nhận biết filter của Netscape, để giải quyết vấn đề này ta tạo một biến boolean để xác định trình duyệt đang dùng

Ví dụ:

```
if (navigator.appName == "Microsoft Internet Explorer")
    ie=true
    else ie=false ;
if (ie)
{
    logo.style.filter = "apha(opacity=30)";
}
```

ie : là biến boolean nó có giá trị là true khi trình duyệt sử dụng là Internet Explorer.

Ví dụ: Hình 18 bảng I có thể được tạo bằng cách dùng javascript khi ta click chuột vào hình trong trang web :

```
<html>
<head>
<script language=javascript>
function imgfilter()
{
    img1.style.filter = "shadow";
    img1.filters.shadow.color="#ff0000";
```

```

    }
</script>
</head>
<body>
    
</body>
</html>

```

### XVIII.1.3. Một vài ứng dụng :

#### 1. Tạo hiệu ứng cuộn với filter (rollover) :

Ta có thể tạo hiệu ứng thay đổi thể hiện của đối tượng khi người dùng tương tác với trang web (chẳng hạn khi trỏ chuột ra lên đối tượng và khi trỏ chuột ra khỏi đối tượng), điều này có thể thực hiện bằng cách thay đổi hình ảnh dùng filter, cách làm này có lợi là không cần phải load nhiều hình ảnh.

Ví dụ:

```

<div style="position:absolute" onmouseover='this.style.filter="glow"'
onmouseout='this.style.filter=""' > TIN NHANH </div>

```

Khi trỏ chuột ra lên trên dòng chữ thì dòng chữ sẽ có hiệu ứng **glow**. Khi trỏ chuột ra khỏi dòng chữ thì không có hiệu ứng filter được áp dụng.

#### 2. Tạo hiệu ứng filter động :

Đối với những hiệu ứng có tham số khi ta thay đổi giá trị của tham số thì thể hiện của đối tượng trong trang web cũng thay đổi theo. Việc thay đổi tham số ta có thể thực hiện bằng chương trình javascript, nếu việc thay đổi này tiếp diễn liên tục theo thời gian đối tượng sẽ có hiệu ứng filter động.

Ví dụ:

```

var oplevel = 0;
objname.style.filter = "alpha()"
ids=setInterval ("dynobject(objname)",200);
function dynobject (object)
{
    if (oplevel <= 100)
    {
        object.filters.Alpha.Opacity = oplevel ;
        oplevel += 5;
    }
    else clearInterval (ids);
}

```

Khi thực thi, đối tượng sẽ áp dụng hiệu ứng filter với tham số opacity thay đổi từ 0-100, từ hoàn toàn trong suốt tới thể hiện bình thường hoàn toàn (người dùng sẽ thấy đối tượng sẽ hiện dần dần), tham số này cứ 200 milisecond thì tăng thêm 5, khi opacity >100 thì chương trình sẽ dừng bởi lệnh clearInterval();

### XVIII.1.4. Dùng filter Light :

Filter Light tạo được hiệu ứng rất lý thú, filter Light tạo hiệu ứng chiếu sáng đối tượng bằng những nguồn sáng khác nhau. Có thể tạo được đến 10 nguồn sáng có màu sắc, cường độ phát sáng, chiếu sáng đối tượng ở một tọa độ xác định (x,y,z với z là chiều cao), những nguồn sáng này có một số định danh từ 0-9, nguồn sáng ban đầu được định danh là 0.

Để điều khiển nguồn sáng, có 2 phương thức thông dụng là : addPoint() và MoveLight().

- **addPoint()** thiết lập một nguồn sáng chiếu sáng đối tượng trên trang web

**Cú pháp:**

**object.filters.light.addPoint (x ,z, y, r, g, b, strength).**

- *object* : tên của đối tượng.
  - *x,y,z* là những số tính bằng pixel xác định vị trí của nguồn sáng so với gốc toạ độ là vị trí trên bên trái của đối tượng.
  - *r,g,b* là giá trị màu theo hệ màu RGB để xác định màu tổng hợp của nguồn sáng.
  - *strength* : giá trị xác định độ mạnh của nguồn sáng.
- **MoveLight()** di chuyển nguồn sáng tới vị trí mới được xác định bởi các tham số.

**Cú pháp :**

**object.filters.light.MoveLight (light, x, y, z, absolute).**

- *light* : là số định danh của nguồn sáng.
- *x, y, z* : xác định toạ độ mới của nguồn sáng, toạ độ này phụ thuộc vào tham số absolute.
- *absolute* : có 2 giá trị là true và false. Nếu là true thì đối tượng sẽ di chuyển tới vị trí x,y, z. Nếu là false thì đối tượng sẽ dịch đi từ vị trí ban đầu với độ dời theo 3 chiều được xác định bởi tham số x, y, z.

Việc kết hợp với filter động sẽ cho ta những hiệu ứng vô cùng lý thú mà không cần phải “tiêu tốn đường truyền”.

Ví dụ: Hình 16 bảng I có thể được tạo bằng cách dùng filter light theo mã sau (phải chạy thực trong IE chứ không phải preview trong frontpage)

```
<body>
  
  <script language=javascript>
    img1.filters.light.addpoint(40,30,130,0,255,0,100)
    img1.filters.light.addpoint(40,30,20,0,255,0,100)
    img1.filters.light.addpoint(40,30,20,0,255,0,100)
  </script>
</body>
```

## XVIII.2. TRANSITION :

Transition là hiệu ứng áp dụng cho một đối tượng trong một khoảng thời gian, nó tương tự như filter động, thường dùng để tạo một hiệu ứng đặc biệt khi thay thế một đối tượng này với một đối tượng khác. Transition thường dùng để tạo một slide show kèm theo hiệu ứng. Nó chỉ được hỗ trợ bởi Internet Explorer, trong Netscape khi sử dụng transition bằng CSS, nó sẽ bỏ qua tất cả thuộc tính style của đối tượng.

Giống như filter, transition có thể được ứng dụng cho đối tượng bằng cách dùng CSS hay chương trình javascript. Có hai loại transition là : **blend transition** và **reveal transition**.

- **Blend Transition** : tạo hiệu ứng thay thế dần dần đối tượng thứ nhất bằng đối tượng thứ hai.
- **Reveal Transition** : khi thay thế đối tượng thứ nhất bằng đối tượng thứ hai, và có thể áp dụng một trong rất nhiều hiệu ứng.

**XVIII.2.1. Áp dụng transition bằng CSS :****1. Cú pháp blend transition trong CSS :****filter : blendTrans ( Duration=value).**

- *value* : lượng thời gian tính bằng giây ấn định thời gian cho hiệu ứng tiếp diễn.

**2. Cú pháp cho reveal transition****filter : revealTrans ( Duration = value, Transition = type)**

- *type* : một số từ 0 – 23, chỉ loại hiệu ứng transition theo bảng sau :

Transition	Type	Transition	Type
Box in	0	Random dissolve	12
Box out	1	Split vertical in	13
Circle in	2	Split vertical out	14
Circle out	3	Split horizontal in	15
Wipe up	4	Split horizontal out	16
Wipe down	5	Strips left down	17
Wipe right	6	Strips left up	18
Wipe left	7	Strips right down	19
Vertical blinds	8	Strips right up	20
Horizontal blinds	9	Random bars horizontal	21
Checkerboard Across	10	Random bars vertical	22
Ckeckerboard Down	11	Random (0--22)	23

**XVIII.2.2. Áp dụng transition bằng javascript :**

Cú pháp trong javascript cho tham chiếu tới transition tương tự như đối với filter. Để thiết lập thời gian diễn tiến cho transition ta có thể viết như sau :

```
object.style.filter = “blendTrans ( Duration=2)”;
```

Hoặc dùng tập hợp filter :

```
object.filters.bendTrans.Duration = 2;
```

- *object* : là tên của đối tượng trong trang mà ta muốn áp dụng transition.
- Chú ý khi xác định transition thì hiệu ứng transition vẫn chưa xảy ra, để làm điều này ta phải thực hiện thêm một bước nữa là viết chương trình javascript để thực thi .

**XVIII.2.3. Thực thi một transition :**

Khi transition được xác định, để thi hành transition ta phải chạy chương trình javascript thực thi bốn bước :

- Thiết lập trạng thái khởi đầu cho đối tượng (có thể bỏ qua).
- Dùng phương thức apply() cho đối tượng.
- Chỉ định trạng thái kết thúc.
- Dùng phương thức Play() để thi hành transition.

Trạng thái khởi đầu là hình ảnh của đối tượng trước khi thực thi transition. Có thể là trạng thái ẩn hay hiện của đối tượng (thuộc tính visiblility là hidden hay visible) hay

trong trường hợp một hình ảnh (<img>) đó là hình ảnh thể hiện của file được chỉ định bởi thuộc tính src.

### 1. Dùng phương thức apply() cho đối tượng :

**object.filters.transition\_type.apply();**

hoặc :

**object.filters[ i ].apply();**

- *object* : tên đối tượng .
- *transition\_type* : blendTrans hay revealTrans.
- *i* là chỉ số của filter tính từ 0.

Nếu chỉ có một filter cho đối tượng thì ta có thể viết **object.filters[0].apply()**.

Chú ý: tập hợp filter có thể xem là bao gồm cả filter và transition. Sử dụng phương thức apply () không thực sự chạy transition, ta còn cần phải chỉ định trạng thái kết thúc của đối tượng .

### 2. Chỉ định trạng thái kết thúc :

Nếu ta muốn transition chuyển một đối tượng từ trạng thái ẩn (hidden) ban đầu sang trạng thái khả kiến (visible) thì ta phải chỉ định trạng thái kết thúc là hiện đối tượng (thuộc tính visibility là visible).

Nếu transition dùng đổi một hình bằng một hình khác, thì chỉ định trạng thái kết thúc bằng lệnh javascript cho biết file hình mới cho đối tượng.

### 3. Dùng phương thức play() : để thực thi transition cú pháp như sau :

**object.filters.transition\_type.play();**

hoặc :

**object.filters[ i ].play();**

Ví dụ: tạo transition cho đối tượng hình ảnh :

```
<html>
  <head>
    <style>
      # img1 { filter : revealTrans( Duration=2, Transition=6 ) }
    </style>
    <script>
      function swapit ()
      {
        img1.filters.revealTrans.apply();
        img1.src = "image2.jpg";
        img1.filters.revealTrans.play();
      }
    </script>
  </head>
  <body>
    
  </body>
</html>
```

Trong ví dụ trên ta áp dụng style cho đối tượng img1, xác định reveal transition với hiệu ứng wipe right (ứng với transition=6) cho nó. Khi trang web được load thì đối tượng img1 thể hiện hình ảnh của file nguồn là image1.jpg .

Khi người dùng click vào hình ảnh thì hàm `swaptit()` được gọi : Trạng thái ban đầu của đối tượng `img1` là hình ảnh của file nguồn `image1.jpg`, áp dụng phương thức `apply()`, sau đó xác định trạng thái kết thúc của đối tượng là file hình `image2.jpg`, cuối cùng dùng phương thức `play()` để thi hành transition.

Trình duyệt sẽ áp dụng transition với hiệu ứng `wipe right` khi thay đổi hình từ file `image1.jpg` sang file `image2.jpg`.

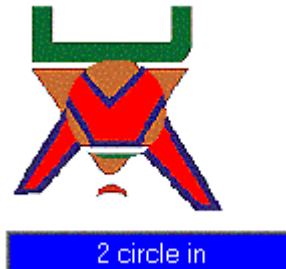
Ví dụ:

```
<html>
  <head>
    <title>Transition</title>
    <script language=javascript >
      // tạo dãy chứa các hình dùng cho transition
      var arrimg = new Array();
      arrimg[0] = "vana.gif";
      arrimg[1] = "vanb.gif";
      arrimg[2] = "vanc.gif";
      // tạo dãy chứa 23 loại transition
      var arname = new Array("box in","box out","circle in", "circle out","wipe
up","wipe down","wipe right","wipe left","vertical blind","horizon
blind","checkerboard Across","checkerboard down","random dissolse","split
vertical in", "split vertical out","split horizontal in","split horizontal out","strips
left down","strips left up","strips right down","strips right up","random bars
horizontal","random bars vertical","ran dom");
      // khởi động biến đếm i, biến dùng để truy xuất lần lượt file hình trong dãy arrimg
      var i=0;
      // khởi động biến đếm j, biến dùng để truy xuất lần lượt các loại transition trong
dãy arriname, bắt đầu bằng -1 để khi vào vòng lặp tăng lên 0 arname[0] sẽ truy
xuất tới hiệu ứng transition 1 (box in).
      var j = -1;
      function show()
      {
        i = i+1;
        j=j+1;
      // để tránh vượt chỉ số của dãy
        if (i > 2) i=0;
        if (j > 23) j=0;
      // gán string vào text box (texta)
        texta.value = j + " " + arname[j]
        imga.filters[0].transition = j;
        imga.filters[0].apply();
        imga.src= arrimg[i];
        imga.filters[0].play();
      // lặp đệ qui
        setTimeout ("show()",2000)
      }
    </script>
  </head>
  <body onload="show()">
    <br>
```



```
<input id=texta type=text style="text-align:center; color:white; background :  
blue">  
</body>  
</html>
```

Khi chương trình thực thi, hình sẽ thay đổi lần lượt từ vanb.gif sang vanc.gif sang vana.gif rồi quay về vana.gif ... cứ tiếp tục như thế. Mỗi lần chuyển hình sẽ kèm theo hiệu ứng transition khác nhau từ 0 đến 23, số và tên của hiệu ứng sẽ được hiện trong text box.



## MỤC LỤC

CHƯƠNG I: GIỚI THIỆU VỀ WEB.....	1
I.1. CÁC KHÁI NIỆM CƠ BẢN:.....	1
I.2. GIỚI THIỆU KHÁI QUÁT VỀ WEB.....	2
I.3. TAG HTML:.....	2
I.4. CẤU TRÚC CƠ BẢN CỦA TRANG WEB:.....	3
I.5. CÁC TAG HTML CƠ BẢN :.....	3
I.6. MỘT SỐ THAO TÁC TRONG CỬA SỔ TRÌNH DUYỆT.....	10
CHƯƠNG II: SIÊU LIÊN KẾT-HÌNH ẢNH.....	13
II.1. GIỚI THIỆU SIÊU LIÊN KẾT.....	13
II.2. TẠO SIÊU LIÊN KẾT.....	13
II.3. HÌNH ẢNH TRÊN TRANG WEB:.....	15
CHƯƠNG III: DANH SÁCH.....	19
III.1. DANH SÁCH KHÔNG CÓ THỨ TỰ (Unorder List -UL).....	19
III.2. DANH SÁCH CÓ THỨ TỰ (OrderList – OL).....	19
III.3. DANH SÁCH ĐỊNH NGHĨA:.....	21
CHƯƠNG IV: BẢNG VÀ TRÌNH BÀY TRANG.....	23
IV.1. BẢNG.....	23
IV.2. CÁC THUỘC TÍNH:.....	24
IV.3. TRÌNH BÀY TRANG.....	27
CHƯƠNG V: FRAME.....	30
V.1. KHÁI QUÁT VỀ FRAME.....	30
V.2. CÁCH TẠO MỘT FRAME LAYOUT.....	30
CHƯƠNG VI: FORM.....	37
VI.1. GIỚI THIỆU FORM.....	37
VI.2. CÁC PHẦN TỬ CỦA FORM:.....	37
CHƯƠNG VII: CASCADING STYLE SHEET-CSS.....	47
VII.1. GIỚI THIỆU.....	47
VII.2. CÁCH TẠO:.....	47
VII.3. ĐỊNH BẢNG MẪU CHO LỚP (CLASS):.....	49
CHƯƠNG VIII: GIỚI THIỆU DREAMWEAVER.....	53
VIII.1. GIỚI THIỆU.....	53
VIII.2. CÀI ĐẶT.....	53
VIII.3. MÀN HÌNH DREAMWEAVER:.....	53
VIII.4. KẾ HOẠCH THIẾT KẾ MỘT WEBSITE.....	55
CHƯƠNG IX: ĐỊNH DẠNG VĂN BẢN- SỬ DỤNG CSS.....	64
IX.1. ĐỊNH DẠNG VĂN BẢN.....	64
IX.2. Sử Dụng CSS (CASCADING STYLE SHEETS).....	66
CHƯƠNG X: HÌNH ẢNH VÀ LIÊN KẾT TRANG TRONG DREAMWEAVER.....	68
X.1. CHÈN HÌNH ẢNH VÀO TRANG WEB:.....	68
X.2. LIÊN KẾT TRANG TRONG DREAMWEAVER:.....	73
CHƯƠNG XI: BẢNG VÀ TRÌNH BÀY TRANG.....	81
XI.1. TABLE :.....	81
XI.2. TRÌNH BÀY TRANG :.....	82
XI.3. TẠO ALBUM LẬT TỪNG HÌNH:.....	88

XI.4. TEMPLATE:.....	89
CHƯƠNG XII: BEHAVIORS - FORM .....	92
XII.1. BEHAVIORS .....	92
XII.2. FORM:.....	96
CHƯƠNG XIII: FRAMESETS – KIỂM TRA VÀ XUẤT BẢN.....	104
XIII.1. GIỚI THIỆU.....	104
XIII.2. CÁCH TẠO TRANG KHUNG VÀ LIÊN KẾT TRANG .....	106
XIII.3. KIỂM TRA VÀ XUẤT BẢN .....	109
CHƯƠNG XIV: TỔNG QUAN VỀ JAVASCRIPT .....	111
XIV.1. GIỚI THIỆU VỀ JAVASCRIPT:.....	111
XIV.2. BIẾN VÀ DỮ LIỆU TRONG JAVASCRIPT .....	114
CHƯƠNG XV: HÀM TRONG JAVASCRIPT .....	118
XV.1. ĐỊNH NGHĨA .....	118
XV.2. CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT .....	119
CHƯƠNG XVI: CÁC CẤU TRÚC ĐIỀU KIỆN.....	122
XVI.1. CẤU TRÚC LỰA CHỌN: .....	122
XVI.2. CẤU TRÚC LẶP: .....	124
CHƯƠNG XVII: MÔ HÌNH ĐỐI TƯỢNG .....	129
XVII.1. MÔ HÌNH DOM ((Document Object Model).....	129
XVII.2. CÁC ĐỐI TƯỢNG CÓ SẴN TRONG JAVASRIPT .....	132
XVII.3. THAY ĐỔI NỘI DUNG ĐỘNG TRÊN TRANG .....	153
CHƯƠNG XVIII: HIỆU ỨNG FILTER VÀ TRANSITION.....	160
XVIII.1. FILTER :.....	160
XVIII.2. TRANSITION : .....	164