

THẠC BÌNH CƯỜNG - VŨ THỊ HẬU

# GIÁO TRÌNH

# THIẾT KẾ

# WEB



NHÀ XUẤT BẢN GIÁO DỤC

**THẠC BÌNH CƯỜNG – VŨ THỊ HẬU**

**Giáo trình**  
**THIẾT KẾ WEB**

*(Dùng cho các trường đào tạo hệ Cao đẳng và Trung cấp chuyên nghiệp)*

**NHÀ XUẤT BẢN GIÁO DỤC**

*Bản quyền thuộc HEVOCO - Nhà Xuất bản Giáo dục*

## *Lời nói đầu*

Ngày nay, khi Internet đã trở nên quen thuộc và gần gũi với mọi người, thậm chí là công cụ không thể thiếu trong đời sống hiện đại thì việc đưa thông tin quảng bá, tìm kiếm thông tin, giao dịch mua bán trên Internet... là những hoạt động thường xuyên của các cư dân trên mạng. Chính vì vậy nhu cầu thiết kế được những trang Web hấp dẫn, thu hút nhiều người truy cập là mục tiêu và mơ ước của nhiều người.

Xuất phát từ nhu cầu đó, cuốn sách trước hết sẽ cung cấp cho người đọc những kiến thức cơ bản nhất, hiểu được cấu tạo và cách thiết kế một trang Web đơn giản. Bên cạnh đó là những lời khuyên để giúp bạn đọc - những người mới bước chân vào thế giới Web chưa có kinh nghiệm, có được định hướng để tạo ra được những trang Web có giao diện đẹp, phù hợp với tiêu chí, mục đích của trang Web và thu hút được người xem.

Đồng thời cuốn sách cũng cung cấp cho các bạn sinh viên kiến thức về Thương mại điện tử, hiện đang là một trong những phương thức kinh doanh mang lại nhiều lợi ích cho nhân loại và là phương thức kinh doanh đại diện cho nền kinh tế tri thức. Sau khi đã nắm được những kiến thức về lập trình Web, bạn đọc có thể dùng công cụ hỗ trợ để làm việc hiệu quả hơn.

Giáo trình *Thiết kế Web* được sử dụng cho việc giảng dạy và học tập của giảng viên và sinh viên các trường Cao đẳng và Trung cấp chuyên nghiệp.

Chúng tôi xin chân thành cảm ơn các giảng viên Hoàng Anh Việt, Lương Thanh Bình, Nguyễn Thị Thu Trang đã đóng góp những ý kiến quý báu để cuốn giáo trình được chính xác và hoàn thiện hơn.

Mặc dù có nhiều cố gắng trong khi biên soạn nhưng giáo trình xuất bản lần đầu khó tránh khỏi thiếu sót. Vì vậy, rất mong nhận được những ý kiến góp ý của các đồng nghiệp và bạn đọc. Các ý kiến góp ý xin gửi về : Công ty cổ phần Sách Đại học - Dạy nghề, 25 Hân Thuyên, Hà Nội.

## CÁC TÁC GIẢ

# *Chương 1*

## **KHÁI QUÁT VỀ CÔNG NGHỆ World Wide Web**

### **1. KHÁI NIỆM SIÊU VĂN BẢN**

“Internet”, “World Wide Web”, và “Web page” không chỉ còn là các thuật ngữ. Giờ đây, các thuật ngữ này đã trở thành hiện thực. Internet là mạng máy tính lớn nhất trên thế giới, được xem như là mạng của các mạng. World Wide Web (WWW) là một tập con của Internet. WWW gồm các Web servers có mặt khắp mọi nơi trên thế giới. Các Web server chứa thông tin mà bất kỳ người dùng nào trên thế giới cũng có thể truy cập được. Các thông tin đó được lưu trữ dưới dạng các trang Web.

**Siêu văn bản** (hypertext) : là văn bản của một tài liệu có thể được truy tìm không theo tuần tự. Người đọc có thể tự do truy theo các dấu vết liên quan qua suốt tài liệu đó bằng các mối liên kết xác định sẵn do người sử dụng tự lập nên.

Trong một môi trường ứng dụng siêu văn bản thực sự, người đọc có thể trỏ vào chỗ tô sáng (highlight) bất kỳ từ nào của tài liệu để lập tức chuyển đến những tài liệu khác có văn bản liên quan đến nó. Cũng có những lệnh cho phép người đọc tự tạo cho riêng mình những dấu vết kết hợp qua suốt tài liệu. Các trình ứng dụng dùng siêu văn bản rất hữu ích trong trường hợp phải làm việc với số lượng văn bản lớn, như các bộ từ điển bách khoa và các bộ sách nhiều tập.

#### **Một số ngôn ngữ siêu văn bản tiêu biểu :**

– HTML. (Hypertext Markup Language) : Ngôn ngữ đánh dấu siêu văn bản. Đây là ngôn ngữ dùng để định dạng dữ liệu. Tất cả các trang Web đều được tạo nên từ các thẻ HTML. Dù Website của bạn được lập trình bằng ngôn ngữ nào đi nữa, thì khi được trả về trình duyệt Web, bạn sẽ chỉ nhận

được các mã HTML. Ví dụ, để định dạng một đoạn văn bản, bạn sử dụng cặp thẻ, để xuống dòng, bạn sử dụng thẻ. HTML cùng với HTTP tạo nên nền móng của một loại dịch vụ mà chúng ta gọi là Web như ngày nay.

– XML (Extensible Markup Language) : Ngôn ngữ đánh dấu có thể mở rộng. Khác với HTML, XML không định nghĩa bất cứ thẻ nào cả. Tất cả các thẻ và ý nghĩa của chúng đều do người sử dụng tạo ra. XML được tạo ra để xác định dữ liệu và do đó được dùng như một công cụ để truyền tải dữ liệu có cấu trúc giữa các ứng dụng, đặc biệt là ứng dụng Web.

## 2. MÁY CHỦ WEB

Mạng là một nhóm các máy tính kết nối với nhau. Internet là mạng của các mạng. Giao thức TCP/IP (Transmission Control Protocol/ Internet Protocol) cung cấp việc kết nối tất cả các máy tính trên thế giới.

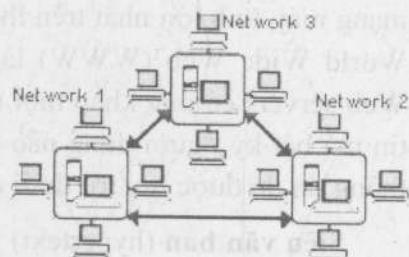
WWW là một tập con của Internet. Nó bắt đầu như là đề án nghiên cứu cấp quốc gia tại phòng nghiên cứu CERN ở Thụy Sĩ. Ngày nay, nó cung cấp thông tin cho người dùng trên toàn thế giới.

WWW hoạt động dựa trên ba cơ chế để đưa các tài nguyên có giá trị đến với người dùng. Đó là:

– Giao thức: Người dùng tuân theo các giao thức này để truy cập tài nguyên trên Web. Hyper Text Transfer Protocol (HTTP) là giao thức được WWW sử dụng.

– Địa chỉ: WWW tuân theo một cách thức đặt tên thống nhất để truy cập vào các tài nguyên trên Web. URL được sử dụng để nhận dạng các trang và các tài nguyên trên Web.

– HTML: Ngôn ngữ đánh dấu siêu văn bản (HTML) được sử dụng để tạo các tài liệu có thể truy cập trên Web.



Hình 1.1. Internet



Hình 1.2

Trình duyệt yêu cầu đến máy chủ

Tài liệu HTML được tạo ra bằng cách sử dụng các thẻ và các phần tử của HTML. File được lưu trên Web server với đuôi .htm hoặc .html.

Khi người dùng sử dụng trình duyệt để yêu cầu một số thông tin nào đó, Web server sẽ đáp ứng các yêu cầu đó. Nó gửi thông tin được yêu cầu đến trình duyệt dưới dạng các trang web. Trình duyệt định dạng thông tin do máy chủ gửi về và hiển thị chúng.

### **3. TRÌNH DUYỆT WEB**

Tài liệu HTML được hiển thị trên trình duyệt. Vậy trình duyệt là gì ? Trình duyệt là một ứng dụng được cài đặt trên máy khách. Trình duyệt đọc mã nguồn HTML và hiển thị trang theo các lệnh trong đó.

Trình duyệt được sử dụng để xem các trang Web và điều hướng. Trình duyệt được biết đến sớm nhất là Mosaic, được phát triển bởi Trung tâm ứng dụng siêu máy tính quốc gia Mỹ (NCSA). Ngày nay, có nhiều trình duyệt được sử dụng trên Internet. Netscape's Navigator và Microsoft's Internet Explorer là hai trình duyệt được sử dụng phổ biến. Đối với người dùng, trình duyệt dễ sử dụng vì nó có giao diện đồ họa với việc trỏ và kích chuột.

Để tạo một tài liệu nguồn, bạn phải dùng một trình soạn thảo HTML. Ngày nay, có nhiều trình soạn thảo đang được sử dụng: Microsoft FrontPage là một công cụ tổng hợp được dùng để tạo, thiết kế và hiệu chỉnh các trang Web. Chúng ta có thể thêm văn bản, hình ảnh, bảng và những thành phần HTML khác vào trang.Thêm vào đó, một biểu mẫu cũng có thể được tạo ra bằng FrontPage. Khi tạo xong giao diện cho trang Web, FrontPage sẽ tự động tạo mã HTML cần thiết. Chúng ta cũng có thể dùng Notepad để tạo tài liệu HTML. Để xem được tài liệu trên trình duyệt bạn phải lưu nó với đuôi là .htm hay .html.

# *Chương 2*

## **CƠ BẢN VỀ NGÔN NGỮ HTML**

### **1. GIỚI THIỆU HTML**

Ngôn ngữ đánh dấu siêu văn bản chỉ rõ một trang Web được hiển thị như thế nào trong một trình duyệt. Sử dụng các thẻ và các phần tử HTML, bạn có thể:

- Điều khiển hình thức và nội dung của trang.
- Xuất bản các tài liệu trực tuyến và truy xuất thông tin trực tuyến bằng cách sử dụng các liên kết được chèn vào tài liệu HTML.
- Tạo các biểu mẫu trực tuyến để thu thập thông tin về người dùng, quản lý các giao dịch...
- Chèn các đối tượng như audio clip, video clip, các thành phần ActiveX và các Java Applet vào tài liệu HTML.

Tài liệu HTML tạo thành mã nguồn của trang Web. Khi được xem trên trình soạn thảo, tài liệu này là một chuỗi các thẻ và các phần tử mà chúng xác định trang Web hiển thị như thế nào. Trình duyệt đọc các file có đuôi .htm hay .html và hiển thị trang Web đó theo các lệnh có trong đó.

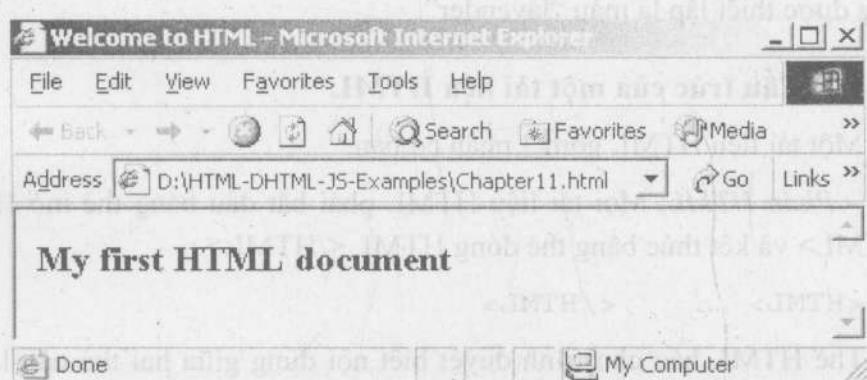
Ví dụ, theo cú pháp HTML dưới đây sẽ hiển thị thông điệp “My first HTML document”.

#### ***Ví dụ 1***

<HTML>

```
<HEAD>
    <TITLE>Welcome to HTML</TITLE>
</HEAD>
<BODY>
    <H3>My first HTML document</H3>
</BODY>
</HTML>
```

Trình duyệt thông dịch những lệnh này và hiển thị trang Web như hình 2.1.



Hình 2.1. Kết quả ví dụ 1

## 1.1. HTML Development

Các lệnh HTML được gọi là các thẻ. Các thẻ này được dùng để điều khiển nội dung và hình thức trình bày của tài liệu HTML. Thẻ mở ("<>") và thẻ đóng ("</>"), chỉ ra sự bắt đầu và kết thúc của một lệnh HTML.

Ví dụ, thẻ HTML được sử dụng để đánh dấu sự bắt đầu và kết thúc của tài liệu HTML.

```
<HTML>
```

```
  ...  
</HTML>
```

Chú ý rằng các thẻ không phân biệt chữ hoa và chữ thường, vì thế bạn có thể sử dụng <html> thay cho <HTML>

Thẻ HTML bao gồm:

```
<ELEMENT ATTRIBUTE = value>
```

Trong đó :

- Phần tử (element): nhận dạng thẻ
- Thuộc tính (attribute): Mô tả thẻ
- Value: giá trị được thiết lập cho thuộc tính

Ví dụ, <BODY BGCOLOR = lavender>

Trong ví dụ trên, BODY là phần tử, BGCOLOR (nền) là thuộc tính và “lavender” là giá trị. Khi cú pháp HTML được thực hiện, màu nền cho cả trang được thiết lập là màu “lavender”.

## 1.2. Cấu trúc của một tài liệu HTML

Một tài liệu HTML gồm 3 phần cơ bản:

– *Phần HTML*: Mọi tài liệu HTML phải bắt đầu bằng thẻ mở HTML <HTML> và kết thúc bằng thẻ đóng HTML </HTML> ;

<HTML> ... </HTML>

Thẻ HTML báo cho trình duyệt biết nội dung giữa hai thẻ này là một tài liệu HTML.

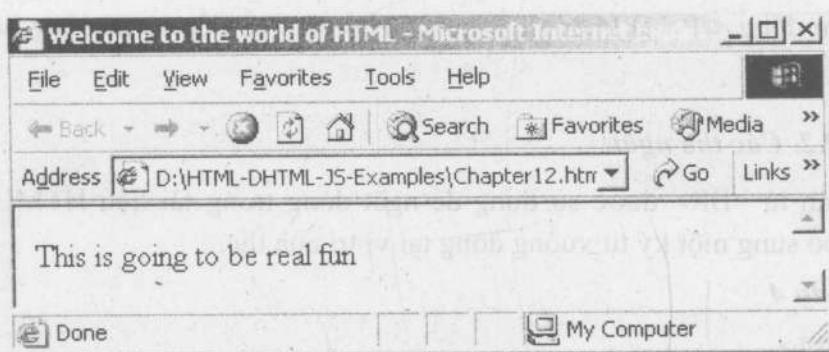
– *Phần tiêu đề*: Phần tiêu đề bắt đầu bằng thẻ <HEAD> và kết thúc bởi thẻ </HEAD>. Phần này chứa tiêu đề được hiển thị trên thanh điều hướng của trang Web. Tiêu đề nằm trong thẻ TITLE, bắt đầu bằng thẻ <TITLE> và kết thúc là thẻ </TITLE>.

Tiêu đề là phần khá quan trọng. Các mốc được dùng để đánh dấu một Website. Trình duyệt sử dụng tiêu đề để lưu trữ các mốc này. Do đó, khi người dùng tìm kiếm thông tin, tiêu đề của trang Web cung cấp từ khoá chính yếu cho việc tìm kiếm.

– *Phần thân*: Phần này nằm sau phần tiêu đề, bao gồm văn bản, hình ảnh và các liên kết mà người dùng muốn hiển thị trên trang web của mình. Phần thân bắt đầu bằng thẻ <BODY> và kết thúc bằng thẻ </BODY>.

### Ví dụ 2

```
<HTML>
    <HEAD>
        <TITLE>Welcome to the world of HTML</TITLE>
    </HEAD>
    <BODY>
        <P>This is going to be real fun</P>
    </BODY>
</HTML>
```



Hình 2.2. Kết quả của ví dụ 2

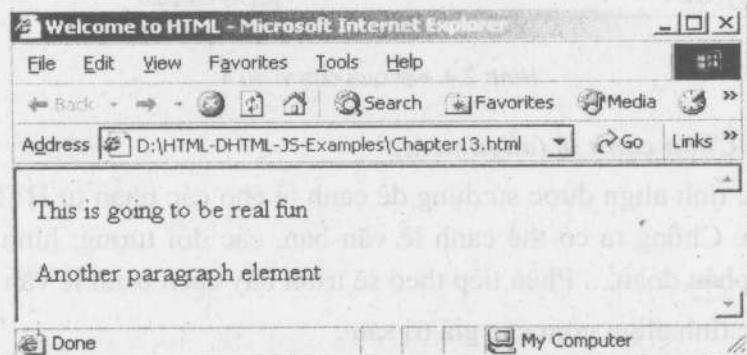
### 1.2.1. Đoạn

Ví dụ 2 sử dụng thẻ <P> để trình bày một đoạn.

Khi viết một bài báo hay một bài luận, thông thường nội dung sẽ được nhóm thành nhiều đoạn với mục đích là nhóm các ý tưởng logic lại với nhau và áp dụng một số định dạng cho nội dung. Trong một tài liệu HTML, nội dung có thể được nhóm thành các đoạn. Thẻ đoạn <P> được sử dụng để đánh dấu sự bắt đầu của một đoạn mới.

### Ví dụ 3

```
<HTML>
  <HEAD>
    <TITLE>Welcome to HTML.</TITLE>
  </HEAD>
  <BODY BGCOLOR = lavender>
    <P>This is going to be real fun
    <P> Another paragraph element
  </BODY>
</HTML>
```



Hình 2.3. Kết quả của ví dụ 3

Thẻ đóng </P> là không bắt buộc. Thẻ <P> kế tiếp sẽ tự động bắt đầu một đoạn mới.

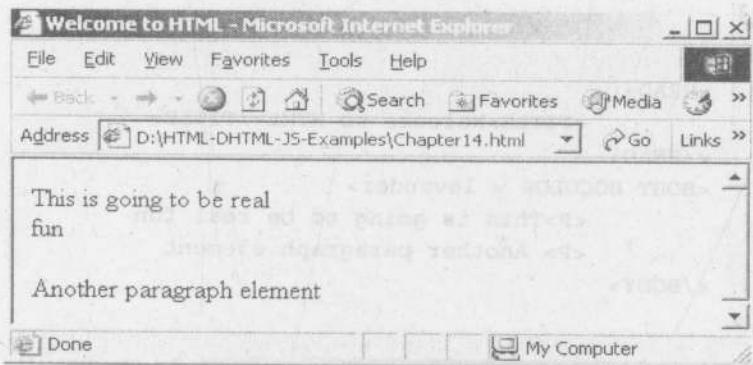
### 1.2.2. Các thẻ ngắt

Phần tử <BR> được sử dụng để ngắt dòng trong tài liệu HTML. Thẻ <BR> bổ sung một ký tự xuống dòng tại vị trí của thẻ.

#### Ví dụ 4

```
<HTML>
```

```
    <HEAD>
        <TITLE>Welcome to HTML</TITLE>
    </HEAD>
    <BODY BGCOLOR = lavender>
        <P>This is going to be real <BR>fun
        <P> Another paragraph element
    </BODY>
</HTML>
```



Hình 2.4. Kết quả của ví dụ 4

### 1.2.3. Chọn canh lề (alignment)

Thuộc tính align được sử dụng để canh lề cho các phần tử HTML trong trang Web. Chúng ta có thể canh lề văn bản, các đối tượng, hình ảnh, các đoạn, các phân đoạn,... Phân tiếp theo sẽ trình bày cách canh lề văn bản:

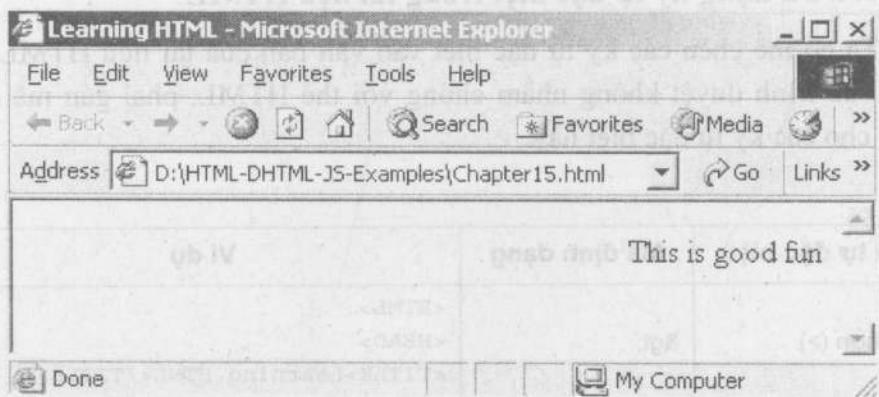
Thuộc tính align gồm các giá trị sau :

Giá trị	Mô tả
Left	Văn bản được canh lề trái
Center	Văn bản được canh giữa
Right	Văn bản được canh phải
Justify	Văn bản được canh đều hai bên

Canh lề được mặc định dựa vào hướng của văn bản. Nếu hướng văn bản là từ trái sang phải thì mặc định là canh lề trái.

### Ví dụ 5

```
<HTML>
  <HEAD>
    <TITLE>Learning HTML</TITLE>
  </HEAD>
  <BODY BGCOLOR=lavender>
    <P align = right>This is good fun</P>
  </BODY>
</HTML>
```



Hình 2.5. Kết quả của ví dụ 5

### 1.3. Sử dụng thẻ <META>

Phần tiêu đề cũng chứa phần tử META. Phần tử này cung cấp thông tin về trang web bao gồm tên tác giả, tên phần mềm dùng để viết trang đó, tên

công ty, thông tin liên lạc... Phần tử META sử dụng sự kết hợp giữa thuộc tính và giá trị.

Ví dụ, để chỉ Graham Browne là tác giả, người ta sử dụng phần tử META như sau:

```
<META name="Author" content="Graham Browne">
```

Tác giả của tài liệu là "Graham Browne"

Thuộc tính http-equiv có thể được sử dụng để thay thế thuộc tính name. Máy chủ HTTP sử dụng thuộc tính này để tạo ra một đầu đáp ứng HTTP (HTTP response header). Đầu đáp ứng được truyền đến trình duyệt để nhận dạng dữ liệu. Nếu trình duyệt hiểu được đầu đáp ứng này, nó sẽ tiến hành các hành động đặc biệt đối với đầu đáp ứng đó.

Ví dụ : <META http-equiv="Expires" content="Mon, 15 Sep 2003 14:25:27 GMT" > sẽ sinh ra một đầu đáp ứng HTTP như sau:

```
Expires: Mon, 15 Sep 2003 14:25:27 GMT
```

Do vậy, nếu tài liệu đã lưu lại, HTTP sẽ biết khi nào truy xuất một bản sao của tài liệu tương ứng.

#### 1.4. Sử dụng ký tự đặc biệt trong tài liệu HTML

Ta có thể chèn các ký tự đặc biệt vào văn bản của tài liệu HTML. Để đảm bảo trình duyệt không nhầm chúng với thẻ HTML, phải gán mã định dạng cho các ký tự đặc biệt này.

Ký tự đặc biệt	Mã định dạng	Ví dụ
Lớn hơn (>)	&gt;	<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;CODE&gt;If A &amp;gt; B Then &lt;BR&gt; A = A + 1 &lt;/CODE&gt; &lt;P&gt; The above statement used special characters &lt;/BODY&gt; &lt;/HTML&gt;</pre>

Nhỏ hơn (<)	&lt;	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;CODE&gt;If A &amp;lt; B Then &lt;BR&gt; A = A + 1 &lt;/CODE&gt; &lt;P&gt; The above statement used special characters &lt;/BODY&gt; &lt;/HTML&gt; </pre>
Trích dẫn ("")	&quot;	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &amp;quot; To be or not to be ? &amp;quot; That is the question &lt;/BODY&gt; &lt;/HTML&gt; </pre>
Ký tự "&"	&amp;	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt; William &amp;amp; Graham went to the fair &lt;/BODY&gt; &lt;/HTML&gt; </pre>

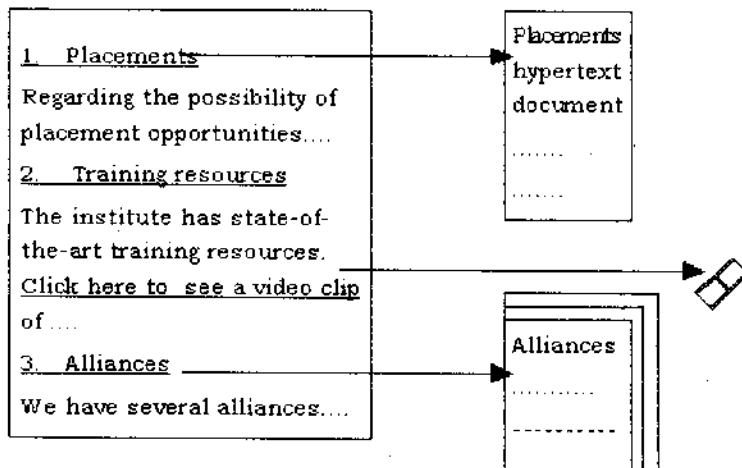
## 2. SỬ DỤNG CÁC SIÊU LIÊN KẾT

Siêu liên kết là sự liên kết giữa một phần tử bên trong tài liệu đến một vị trí khác trong cùng tài liệu đó hoặc đến một tài liệu hoàn toàn khác. *Các siêu liên kết là thành phần quan trọng nhất của hệ thống siêu văn bản*

### 2.1. Giới thiệu siêu liên kết và URL

Khả năng chính của HTML là hỗ trợ siêu liên kết. Một siêu liên kết, hay nói ngắn gọn là một liên kết, là sự kết nối đến tài liệu hay file khác (âm thanh, video) hoặc đến một phần khác trong cùng tài liệu đó. Khi kích vào siêu liên kết, người dùng được đưa đến địa chỉ URL đã chỉ rõ trong liên kết. Các địa chỉ đặt liên kết đến có thể là :

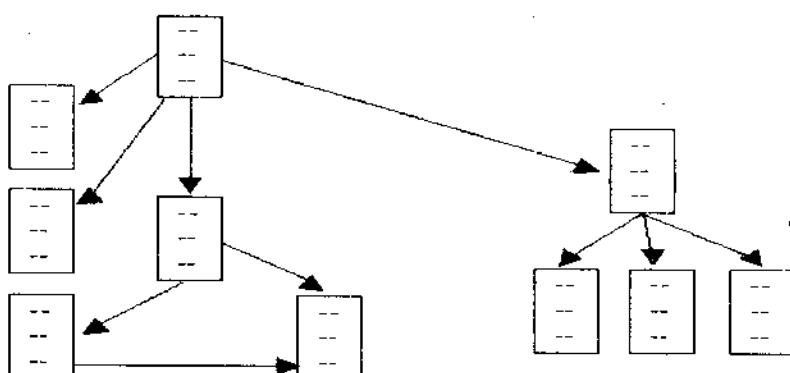
- Một phần khác trong cùng tài liệu.
- Một tài liệu khác.
- Một phần trong tài liệu khác.
- Các file khác – hình ảnh, âm thanh, trích đoạn video.
- Vị trí hoặc máy chủ khác.



Hình 2.6. Sử dụng liên kết

Các liên kết có thể là liên kết trong hoặc liên kết ngoài

- **Liên kết trong** là liên kết kết nối đến các phần trong cùng tài liệu hoặc cùng một Website.
- **Liên kết ngoài** là liên kết kết nối đến các trang trên các Website khác hoặc máy chủ khác.



Hình 2.7. Liên kết trong và liên kết ngoài

Để tạo siêu liên kết, chúng ta cần phải xác định hai thành phần:

1. Địa chỉ đầy đủ hoặc URL của file được kết nối.
2. Điểm nóng cung cấp cho liên kết. Điểm nóng này có thể là một dòng văn bản, thậm chí là một ảnh.

Khi người dùng kích vào điểm nóng, trình duyệt đọc địa chỉ được chỉ ra trong URL và “nhảy” đến vị trí mới.

Mỗi nguồn tài nguyên trên Web có một địa chỉ duy nhất. Ví dụ, 207.46.130.149 là địa chỉ Website của Microsoft. Việc nhớ các con số này là rất khó và dễ nhầm lẫn, vì vậy, người ta sử dụng các URL. URL là một chuỗi cung cấp địa chỉ Internet của Website hay tài nguyên trên World Wide Web.

Định dạng đặc trưng là www.nameofsite.typeofsite.contrycode

Trong đó :

- Nameofsite: tên của site.
- Typeofsite: kiểu của site.
- Contrycode: mã nước.

Ví dụ, 216.239.33.101 có thể được biểu diễn bằng URL là www.google.com

URL cũng nhận biết được giao thức mà site hay tài nguyên được truy nhập. Dạng URL thông thường nhất là “http”, nó cung cấp địa chỉ Internet của một trang Web. Một vài dạng URL khác là “gopher”, nó cung cấp địa chỉ Internet của một thư mục Gopher, và “ftp”, cung cấp vị trí của một tài nguyên FTP trên mạng.

URL cũng có thể tham chiếu đến một vị trí trong một tài nguyên. Ví dụ, bạn có thể tạo liên kết đến một chủ đề trong cùng một tài liệu. Trong trường hợp đó, định danh đoạn được sử dụng ở phần cuối của URL.

Định dạng là giao thức: tên của site / tài liệu chính #định danh đoạn

Có hai dạng URL:

**–URL tuyệt đối** – là địa chỉ Internet đầy đủ của trang hoặc file, bao gồm giao thức, vị trí mạng, đường dẫn tùy chọn và tên file. Ví dụ, http://www.microsoft.com là một địa chỉ URL tuyệt đối.

– **URL tương đối** – là một URL có một hoặc nhiều phần bị thiếu. Trình duyệt lấy thông tin bị thiếu từ trang chứa URL đó. Ví dụ, nếu giao thức bị thiếu, trình duyệt sử dụng giao thức của trang hiện thời.

## 2.2. Sử dụng siêu liên kết

Thẻ <A> được sử dụng để xác định văn bản hay ảnh nào sẽ dùng làm siêu liên kết trong tài liệu HTML. Thuộc tính HREF (tham chiếu siêu văn bản) được dùng để chỉ địa chỉ hay URL của tài liệu hoặc file được liên kết.

Cú pháp của HREF là:

```
<A HREF = protocol://host.domain:port/path/filename>  
                                Hypertext</A>
```

Trong đó :

- Protocol : Đây là loại giao thức. Một số giao thức thường dùng:
  - + http – giao thức truyền siêu văn bản.
  - + telnet – mở một phiên telnet.
  - + gopher – tìm kiếm file.
  - + ftp – giao thức truyền file.
  - + mailto – gửi thư điện tử.
- Host.domain – Đây là địa chỉ Internet của máy chủ.
- Port - Công phục vụ của máy chủ đích.
- HyperText – Đây là văn bản hay hình ảnh mà người dùng cần nhấp vào để kích hoạt liên kết.

### 2.2.1. Liên kết đến những tài liệu khác

Giả sử có hai tài liệu HTML trên đĩa cứng cục bộ, Doc1.html và Doc2.html. Để tạo một liên kết từ Doc1.html đến Doc2.html.

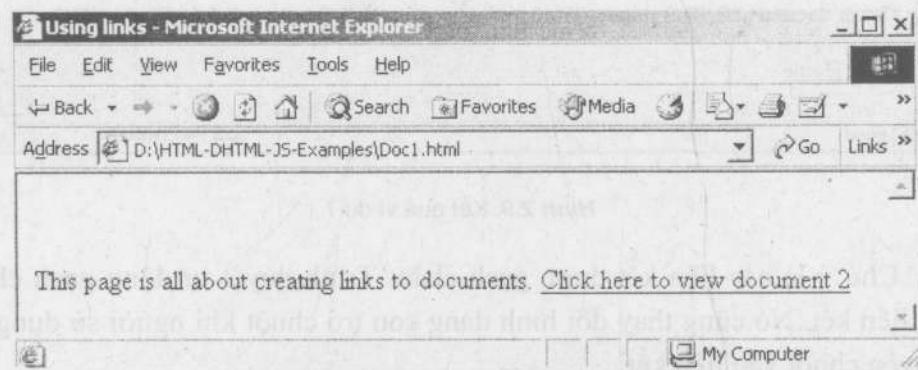
**Ví dụ 6**

```
<HTML>  
  <HEAD>  
    <TITLE> Using links</TITLE>  
  </HEAD>  
  <BODY>
```

```

<BR><BR>
<P> This page is all about creating links to documents.
<A HREF = Doc2.html>Click here to view document 2</A>
</BODY>
</HTML>

```



**Hình 2.8.** Kết quả của ví dụ 6

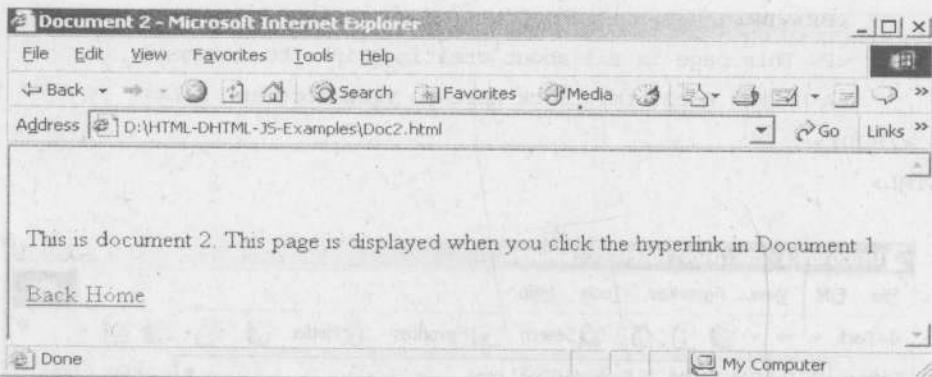
Khi người dùng “nhảy” đến một tài liệu khác, cần cung cấp cách để quay trở lại trang chủ hoặc định hướng đến một file khác.

### Ví dụ 7

```

<HTML>
    <HEAD>
        <TITLE> Document 2</TITLE>
    </HEAD>
    <BODY>
        <BR><BR>
        <P> This is document 2. This page is displayed
        when you click the hyperlink in Document 1
        <BR><BR>
        <A HREF= Doc1.html>Back Home</A>
    </BODY>
</HTML>

```



Hình 2.9. Kết quả ví dụ 7

Chú ý là các liên kết được gạch chân. Trình duyệt tự động gạch chân các liên kết. Nó cũng thay đổi hình dáng con trỏ chuột khi người sử dụng di chuyển chuột vào liên kết.

Ở ví dụ trên, các file nằm trong cùng một thư mục, vì vậy chỉ cần chỉ ra tên file trong thông số HREF là đủ. Tuy nhiên, để liên kết đến các file ở thư mục khác, cần phải cung cấp đường dẫn tuyệt đối hay đường dẫn tương đối.

**Đường dẫn tuyệt đối** chỉ ra đường dẫn đầy đủ từ thư mục gốc đến file. Ví dụ, C:\mydirectory\html examples\ Doc2.htm.

**Đường dẫn tương đối** chỉ ra vị trí liên quan của file với vị trí file hiện tại. Ví dụ, nếu thư mục hiện hành là mydirectory thì đường dẫn sẽ là :

```
<A HREF= "... \Doc3.htm" >Next page</A>
```

Vì vậy, nếu muốn liên kết các tài liệu không liên quan với nhau, ta nên dùng đường dẫn tuyệt đối. Tuy nhiên, nếu có một nhóm tài liệu có liên quan với nhau, ví dụ phần trợ giúp trong HTML, thì nên sử dụng đường dẫn tương đối cho các tài liệu trong nhóm và đường dẫn tuyệt đối cho các tài liệu không liên quan trực tiếp đến chủ đề. Khi đó người dùng có thể cài đặt phần trợ giúp này trong thư mục mình chọn và nó vẫn hoạt động.

### 2.2.2. Liên kết đến các phần trong cùng một tài liệu

Thẻ neo `<A>` (anchor) được sử dụng để người dùng có thể “nhảy” đến những phần khác nhau của một tài liệu. Ví dụ, bạn có thể hiển thị nội dung của trang Web như một loạt các liên kết. Khi người dùng kích vào một đề tài nào đó thì các chi tiết nằm ở một phần khác của tài liệu được hiển thị.

Kiểu liên kết này gọi là “named anchor” vì thuộc tính NAME được sử dụng để tạo các liên kết này.

```
<A NAME = "marker">Topic name</A>
```

Bạn không phải sử dụng bất kỳ văn bản nào để đánh dấu điểm neo. Để dùng, ta sử dụng vạch dấu (marker) trong thông số HREF như sau :

```
<A HREF= "#marker">Topic name</A>
```

Dấu # ở trước tên của siêu liên kết để báo cho trình duyệt biết liên kết này liên kết đến một điểm được đặt tên trong tài liệu. Khi không có tài liệu nào được chỉ ra trước ký tự #, trình duyệt hiểu rằng liên kết này nằm trong cùng tài liệu.

### Ví dụ 8

<HTML>

```
    <HEAD>
```

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

```
    </HEAD>
```

```
    <BODY>
```

```
        <A HREF = #Internet>Internet</A>
```

```
        <BR><BR>
```

```
        <A HREF = #HTML>Introduction to HTML</A>
```

```
        <BR><BR>
```

```
        <A HREF = #Consistency>Unity and Variety</A>
```

```
        <BR><BR>
```

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

```
        <BR>
```

<P> The Internet is a network of networks. That is, computer networks are linked to other networks, spanning countries and today the globe. The TCP/IP transfer protocol provides the bindings that connect all these computers the world over.

```
    </P>
```

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

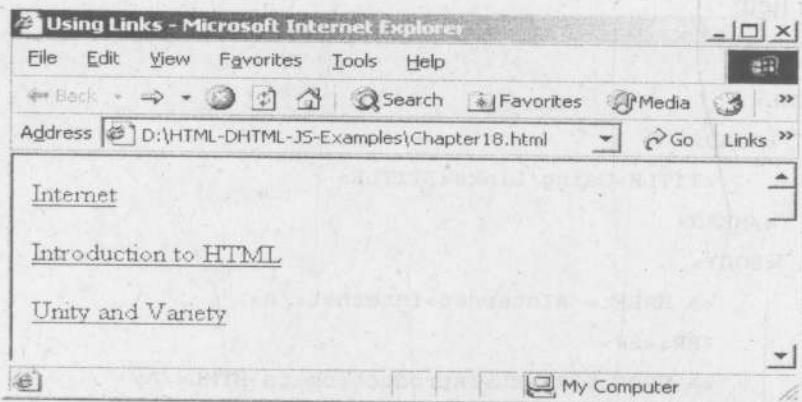
```
    <BR>
```

<P> Hyper Text Markup Language is the standard language that the Web uses for creating and recognizing documents. Although not a subset of, it is loosely related to the Standard Generalized Markup Language (SGML). SGML is a method for representing document formatting languages.

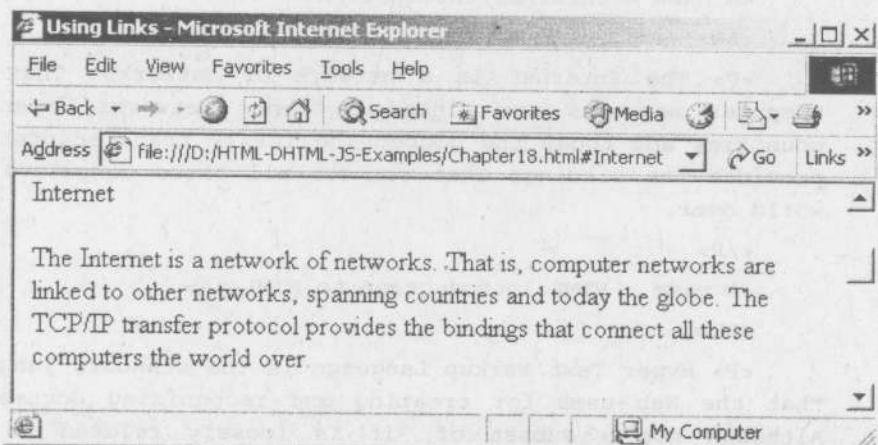
```

</P>
<A name = Consistency>Unity and Variety</A>
<BR>
<P> A basic rule is that of unity and variety. That is, everything should fit together as a unit, but at the same time there is enough variety to keep things interesting. Consistency creates and reinforces the unique identity of a site. Colors, fonts, column layout and other design elements should be consistent throughout every section of the site.
</BODY>
</HTML>

```



Hình 2.10. Kết quả ví dụ 8



Hình 2.11. Kết quả ví dụ 8 sau khi kích vào Internet

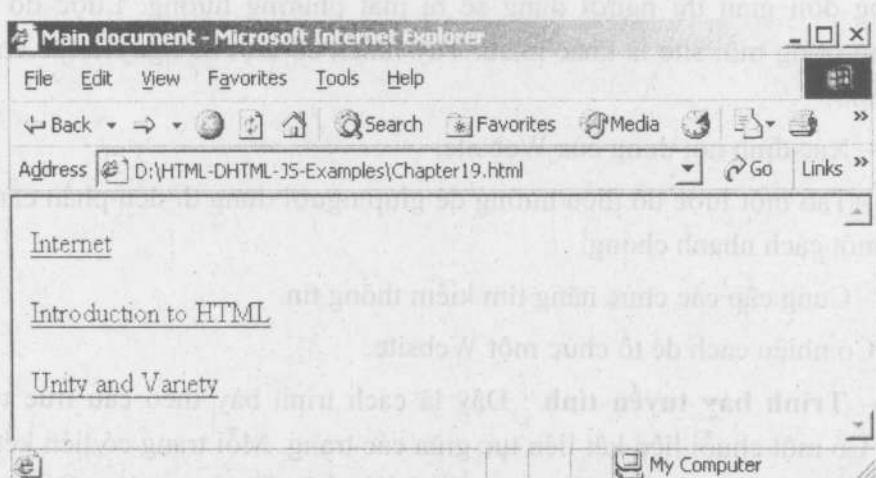
### 2.2.3. Liên kết đến một điểm xác định ở một tài liệu khác

Bây giờ chúng ta đã biết cách sử dụng các vạch dấu trong cùng một tài liệu, hãy thử “nhảy” đến một vị trí trên một tài liệu khác.

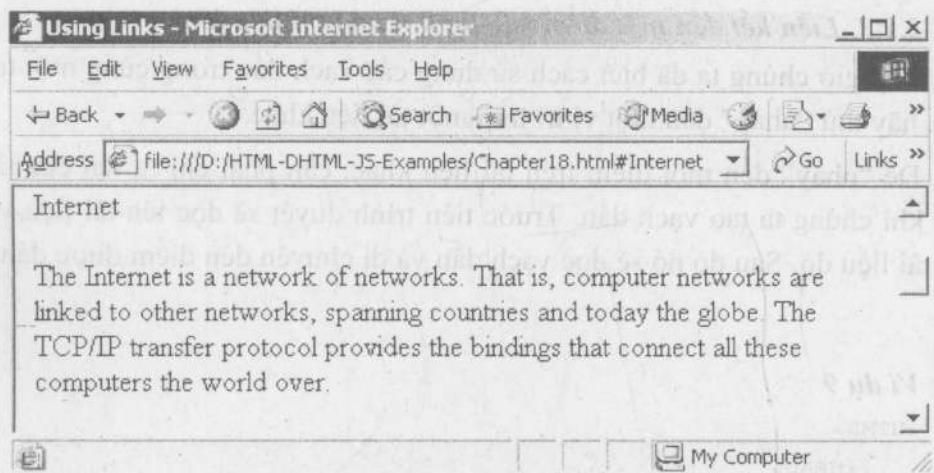
Để “nhảy” đến một điểm trên tài liệu khác, cần phải chỉ ra tên của tài liệu khi chúng ta tạo vạch dấu. Trước tiên trình duyệt sẽ đọc tên tài liệu và mở tài liệu đó. Sau đó nó sẽ đọc vạch dấu và di chuyển đến điểm được đánh dấu.

#### Ví dụ 9

```
<HTML>
  <HEAD>
    <TITLE>Main document</TITLE>
  </HEAD>
  <BODY>
    <A HREF=Chapter18.html#Internet>Internet</A>
    <BR>
    <BR>
    <A HREF= Chapter18.html#HTML>Introduction to HTML</A>
    <BR>
    <BR>
    <A HREF=Chapter18.html#Consistency>Unity and Variety</A>
  </BODY>
</HTML>
```



Hình 2.12. Kết quả ví dụ 9



Hình 2.13. Kết quả ví dụ 9 sau khi kích vào Internet

#### 2.2.4. Sử dụng e-mail

Nếu muốn người sử dụng gửi được e-mail, cần đưa một đặc tính vào trong trang Web cho phép gửi e-mail từ trình duyệt, đó là chèn giá trị *mailto* vào trong thẻ liên kết.

```
<A HREF="mailto:thisperson@mymail.com">
```

### 3. ĐIỀU HƯỚNG QUANH WEBSITE

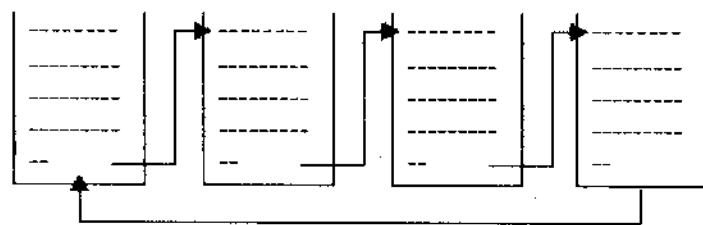
Dù Website có lôi cuốn đến đâu, nếu nó không có một lược đồ điều hướng đơn giản thì người dùng sẽ bị mất phương hướng. Lược đồ điều hướng trong mỗi site là khác nhau. Tuy nhiên có một số nguyên tắc cơ bản cần nhớ:

- Xác định nội dung của Website.
- Tạo một lược đồ điều hướng để giúp người dùng đi đến phần cần tìm đến một cách nhanh chóng.
- Cung cấp các chức năng tìm kiếm thông tin.

Có nhiều cách để tổ chức một Website.

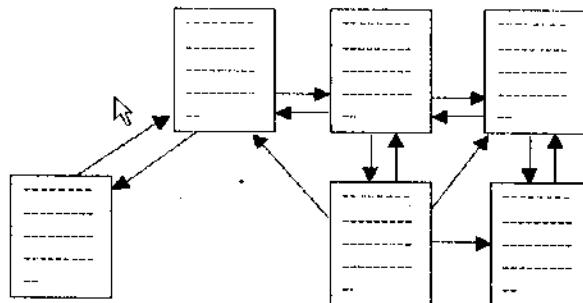
– **Trình bày tuyên tính**: Đây là cách trình bày theo cấu trúc tuyên tính. Có một chuỗi liên kết liên tục giữa các trang. Mỗi trang có liên kết đến trang trước và trang sau. Trang cuối có liên kết đến trang đầu. Định dạng

này được dùng khi cần trình bày thông tin liên tục. Ví dụ, các chương trong cuốn sách hoặc các slide của một bài trình bày mà yêu cầu người dùng đọc thông tin theo trình tự liên tiếp nhau.



Hình 2.14. Trình bày tuyến tính

– **Trình bày theo phân cấp** : Đây là cách trình bày thông thường nhất được sử dụng trong thiết kế Web. Trang chủ liên kết với nhiều trang khác. Người dùng có thể chọn một liên kết và “nhảy” đến trang cần đến. Mỗi trang đều có liên kết về trang chủ.



Hình 2.15. Trình bày theo phân cấp

– **Bản đồ ảnh** : Bản đồ ảnh hay bản đồ site được đưa vào các trang chủ vì đây là cách trình bày trực quan, dễ hiểu. Khi người dùng kích vào các điểm nóng thì tài liệu liên quan được hiển thị.

Khi đã quyết định một lược đồ điều hướng, bắt đầu tiến hành thiết kế tài liệu. Có một số nguyên tắc cần ghi nhớ:

– **Siêu liên kết** nên rõ ràng. Các từ nên được gạch chân và có màu xanh vì trên Web, màu xanh được quy ước là siêu liên kết. Siêu liên kết thường

xuất hiện ở cuối trang. Nếu chúng xuất hiện ở giữa đoạn văn bản, có thể làm người đọc không tập trung vào đề tài.

– **Các trang con** – nên có một thanh menu hoặc thanh điều hướng, nếu không người đọc có thể không thấy các trang chính. Khi người sử dụng kích vào một biểu tượng bất kỳ trên thanh điều hướng thì sẽ chuyển từ trang hiện thời đến trang mới.

– Luôn có một **bảng mục lục** để người dùng chuyển sang một đề tài nào đó. Bảng mục lục (TOC) được hiển thị ở cuối trang. Người dùng có thể chọn một đề tài nào đó để chuyển tới trang cần đến.

– Luôn có nút “Back” và “Forward” trong trang để người dùng có thể điều hướng đến trang cần đến. Trình duyệt có cung cấp các chọn lựa này, nhưng nó sử dụng các trang trong danh sách history mà có thể không phải là các trang trong Website.

## 4. CÁC THÈ CƠ BẢN

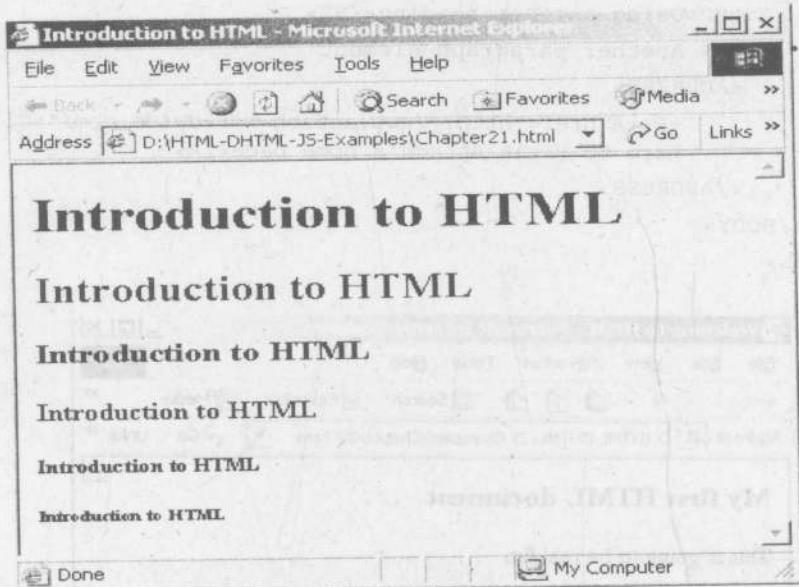
### 4.1. Headings (Tiêu đề)

Phần tử này được sử dụng để cung cấp tiêu đề cho phần nội dung hiển thị trên trang web. Tất cả những phần tử tiêu đề phải có thẻ kết thúc. Nó bắt đầu bằng thẻ `<H1>` và kết thúc bằng `</H1>`.

Những phần tiêu đề được hiển thị to và in đậm hơn để phân biệt chúng với các phần còn lại của văn bản. Phần tiêu đề có thể hiển thị theo một trong sáu kích thước từ H1 đến H6, tuy nhiên phải định rõ kích thước H1, H2... Trình duyệt chú trọng đến cách hiển thị.

#### Ví dụ 10

```
<HTML>
  <HEAD>
    <TITLE>Introduction to HTML.</TITLE>
  </HEAD>
  <H1>Introduction to HTML</H1>
  <H2>Introduction to HTML</H2>
  <H3>Introduction to HTML</H3>
  <H4>Introduction to HTML</H4>
  <H5>Introduction to HTML</H5>
  <H6>Introduction to HTML</H6>
</HTML>
```



Hình 2.16. Kết quả ví dụ 10

## 4.2. Các thẻ mức độ

Phần này sẽ trình bày ba loại thẻ mức độ văn bản: <ADDRESS>, <BLOCKQUOTE> và <PRE>.

### 4.2.1. Thẻ <ADDRESS>

Phần tử <ADDRESS> được dùng để hiển thị các thông tin như tác giả, địa chỉ, chữ ký trong tài liệu HTML. Phần tử này thường hiển thị ở cuối trang và có thể chứa một hoặc một số phần sau:

- Liên kết đến trang chủ.
- Đặc tính chuỗi tìm kiếm.
- Thông tin bản quyền.

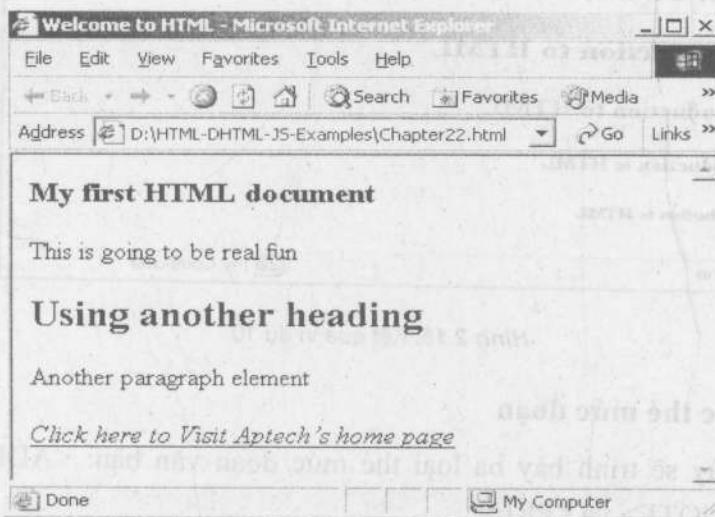
### Ví dụ 11

```
<HTML>
  <HEAD>
    <TITLE>Welcome to HTML</TITLE>
  </HEAD>
  <BODY BGCOLOR = lavender>
    <H3>My first HTML document</H3>
    <P>This is going to be real fun
```

```

<H2>Using another heading</H2>
<P> Another paragraph element
<ADDRESS>
    <P><A href="http://www.aptech-worldwide.com/">Click
        here to Visit Aptech's home page</A>
    </ADDRESS>
</BODY>
</HTML>

```



Hình 2.17. Kết quả ví dụ 11

#### 4.2.2. Thủ **<BLOCKQUOTE>**

Chúng ta có thể chỉ định một trích dẫn văn bản bên trong tài liệu bằng cách sử dụng phần tử BLOCKQUOTE và Q. BLOCKQUOTE được sử dụng cho những phần trích dẫn dài và được hiển thị như một đoạn văn bản thụt vào đầu dòng. Nếu phần trích dẫn ngắn và không cần ngắt đoạn thì sử dụng phần tử Q tốt hơn. Khi sử dụng phần tử Q, bạn phải xác định dấu ngoặc kép.

#### Ví dụ 12

```

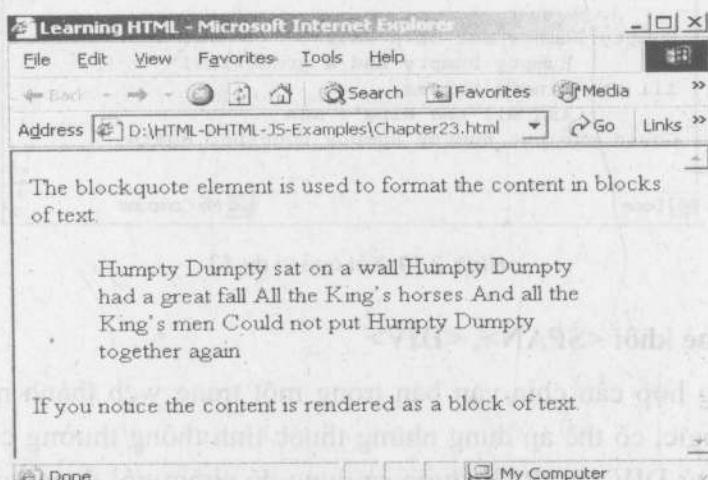
<HTML>
    <HEAD>
        <TITLE>Learning HTML</TITLE>
    </HEAD>
    <BODY>
        <P> The blockquote element is used to format the
            content in blocks of text.

```

```

<BLOCKQUOTE>
    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 Dumpty together again
</BLOCKQUOTE>
<P>If you notice the content is rendered as a
block of text.
</BODY>
</HTML>

```



Hình 2.18. Kết quả ví dụ 12

#### 4.2.3. Thủ <PRE>

Khi cần hiển thị văn bản với định dạng đã xác định trước, chúng ta sử dụng phần tử PRE. Phần tử này dùng để xác định định dạng cho văn bản. Khi văn bản được hiển thị trong trình duyệt, nó sẽ tuân theo tất cả các định dạng đã được xác định trước trong mã nguồn tài liệu HTML.

#### Ví dụ 13

```

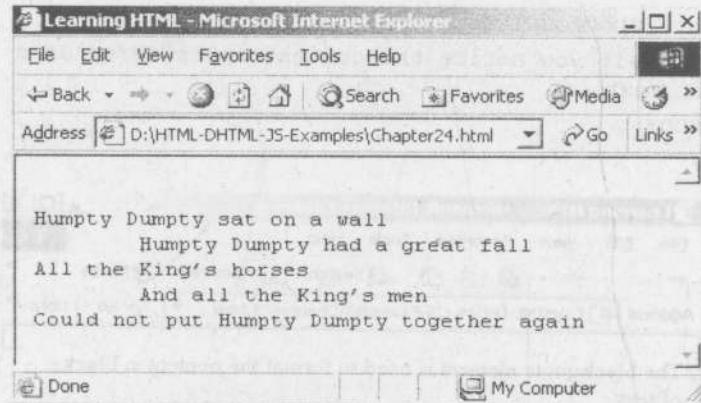
<HTML>
    <HEAD>
        <TITLE>Learning HTML</TITLE>
    </HEAD>
    <BODY>
        <PRE>
            Humpty Dumpty sat on a wall
            Humpty Dumpty had a great fall

```

```

All the King's horses
And all the King's men
Could not put Humpty Dumpty together again
</PRE>
</BODY>
</HTML>

```



*Hình 2.19. Kết quả ví dụ 13*

### 4.3. Thẻ khôi <SPAN>, <DIV>

Trường hợp cần chia văn bản trong một trang web thành những khôi thông tin logic, có thể áp dụng những thuộc tính thông thường cho toàn bộ khôi. Phần tử DIV và SPAN được sử dụng để nhóm nội dung lại với nhau. Phần tử DIV dùng để chia tài liệu thành các thành phần có liên quan với nhau. Phần tử SPAN để chỉ một khoảng các ký tự.

Phần tử SPAN dùng để định nghĩa nội dung trong dòng (in-line) còn phần tử DIV dùng để định nghĩa nội dung mức khôi (block-level).

#### Ví dụ 14

```

<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>

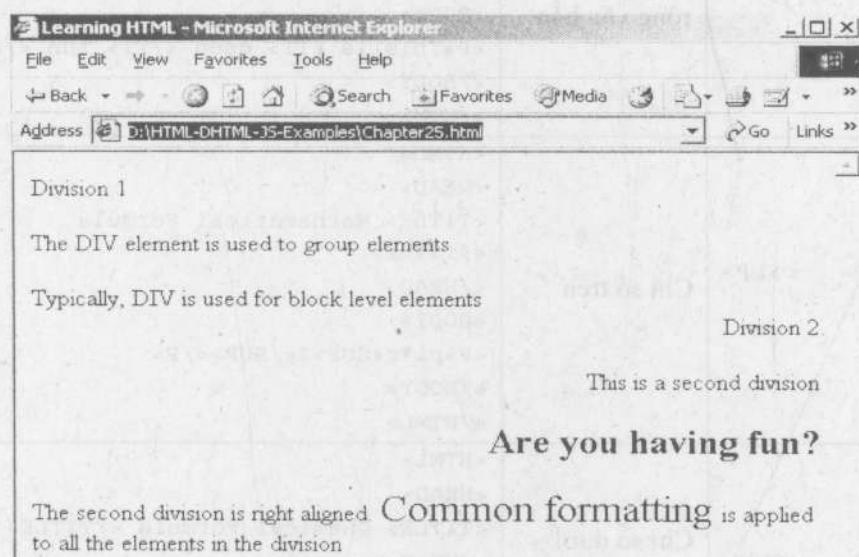
```

```

Division 2
<P>This is a second division
<BR>
<H2>Are you having fun?</H2>
</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>

```

**Chú ý :** Phần tử SPAN không hiển thị trong Netscape



**Hình 2.20.** Kết quả ví dụ 14

#### 4.4. Các thẻ định dạng ký tự thường dùng

HTML có nhiều thẻ có thể được sử dụng để định dạng nội dung của tài liệu. Các thẻ được phân thành các nhóm:

##### 4.4.1. Thẻ định dạng mức vật lý

Thẻ định dạng mức vật lý là những thẻ định rõ hoặc thay đổi đặc điểm của văn bản áp dụng thẻ.

Thẻ	Mô tả	Ví dụ
<B> ... </B>	In đậm	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt;&lt;B&gt;This is good fun&lt;/B&gt;&lt;/P&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<TT>...</TT>	Có định độ rộng văn bản	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt;This is &lt;TT&gt; good &lt;/TT&gt; fun &lt;/P&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<SUP>...</SUP>	Chi số trên	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Mathematical Formula &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt;<math>\pi \cdot r^2</math>&lt;/P&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<SUB>...</SUB>	Chi số dưới	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Chemical Formula &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt;<math>H_2O</math>&lt;/P&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

#### 4.4.2. Thẻ định dạng mức logic

Thẻ định dạng mức logic là những thẻ mô tả “hiệu ứng cần thiết” của văn bản. Sự hiển thị thật sự do trình duyệt điều khiển.

Những thẻ định dạng mức vật lý được các trình duyệt hiển thị như nhau. Những thẻ định dạng mức logic được hiển thị tùy theo trình duyệt mà tài liệu HTML được hiển thị.

Thẻ logic	Mô tả	Ví dụ
<EM>...</EM>	Nhấn mạnh văn bản	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt;&lt;EM&gt;This is good fun &lt;/EM&gt;&lt;/P&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<CODE>.....0</CODE>	Sử dụng trong phân tích của mã chương trình	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt; &lt;CODE&gt;X = X + 1 &lt;BR&gt; Y = Y + 1&lt;/CODE&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<VAR>...</VAR>	Sử dụng cho các biến trong chương trình	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Learning HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; &lt;P&gt;&lt;VAR&gt;X&lt;/VAR&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>
<CITE>...</CITE>	Sử dụng cho các trích dẫn và tham chiếu	<pre> &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt; Learning HTML &lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; It is the mark of an educated mind to be able to entertain a thought without accepting it&lt;CITE&gt; Aristotle &lt;/CITE&gt; &lt;/BODY&gt; &lt;/HTML&gt; </pre>

#### 4.5. Danh sách

Danh sách dùng để nhóm dữ liệu một cách logic. Chúng ta có thể thêm các danh sách vào tài liệu HTML để nhóm các thông tin có liên quan lại với nhau.

*Ví dụ :*

Roses  
Sunflowers  
Orchids  
Apples  
Oranges  
Daffodils  
Mangoes

Có thể được nhóm thành:

Fruits

Apples  
Mangoes  
Oranges

Flowers

Daffodils  
Orchids  
Roses  
Sunflowers

Các loại danh sách mà bạn có thể chèn vào tài liệu HTML là:

- [.] Danh sách không thứ tự
- [.] Danh sách có thứ tự
- [.] Danh sách định nghĩa

#### ***4.5.1. Danh sách không thứ tự***

Đây là loại danh sách đơn giản nhất có thể thêm vào tài liệu HTML. Danh sách không thứ tự là một “bulleted list”. Các mục được bắt đầu bằng dấu chấm tròn “bullet”. Danh sách không thứ tự được nằm trong cặp thẻ **<UL>...</UL>**. Mỗi mục trong danh sách được đánh dấu bằng thẻ **<LI>**. LI được viết tắt của từ List Item. Thẻ kết thúc **</LI>** là tùy chọn (không bắt buộc).

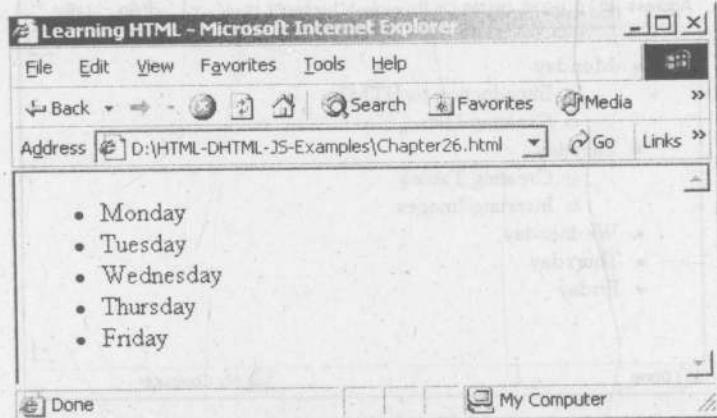
#### ***Ví dụ 15***

```
<HTML>
  <HEAD>
    <TITLE>Learning HTML</TITLE>
  </HEAD>
  <BODY>
```

```

<UL>
    <LI>Monday
    <LI>Tuesday
    <LI>Wednesday
    <LI>Thursday
    <LI>Friday
</UL>
</BODY>
</HTML>

```



Hình 2.21. Kết quả ví dụ 15

Chúng ta có thể tạo ra các danh sách lồng nhau để mô tả nhóm con của thông tin.

### Ví dụ 16

```

<HTML>
    <HEAD>
        <TITLE>Learning HTML</TITLE>
    </HEAD>
    <BODY>
        <UL>
            <LI>Monday
                <UL>
                    <LI>Introduction to HTML
                    <LI>Creating Lists
                </UL>
            <LI>Tuesday
                <UL>
                    <LI>Creating Tables
                    <LI>Inserting Images
                </UL>
            <LI>Wednesday

```

trợ ảnh màu thực và dài màu xám. Các file PNG cũng có thể được nén và chuyển qua mạng.

Khi quyết định định dạng đồ họa, cần nhớ một vài yếu tố:

- **Chất lượng của ảnh** – Chất lượng của ảnh có quan trọng trong quá trình hiển thị không?
- **Độ lớn dữ liệu** – Kích cỡ file càng lớn thì thời gian truyền càng cao.
- **Các yêu cầu hiển thị** - Ảnh hỗ trợ trong suốt, hiển thị tuần tự hay xen kẽ.

### 5.1. Chèn ảnh

Thẻ IMG dùng để chèn ảnh vào trong tài liệu HTML. Chúng ta có thể đặt thẻ IMG tại vị trí mà ảnh được hiển thị. Thẻ IMG không có nội dung, nó hiển thị nội dung bằng cách xác định thuộc tính SRC. Cú pháp là:

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

Trong đó SRC (source) là thuộc tính và giá trị là một URL, chỉ vị trí chính xác của file ảnh.

Khi cần cung cấp cho người sử dụng một số nội dung liên quan đến hình ảnh, ta có thể canh lề ảnh cùng với văn bản.

Thuộc tính ALIGN của thẻ IMG được sử dụng để điều chỉnh canh lề của ảnh với văn bản xung quanh.

```
<IMG ALIGN=position SRC="PICTURE.GIF">
```

Trong đó, vị trí của ảnh có thể là trên (TOP), dưới (BOTTOM), ở giữa (MIDDLE), trái (LEFT) hoặc phải (RIGHT).

#### Ví dụ 24

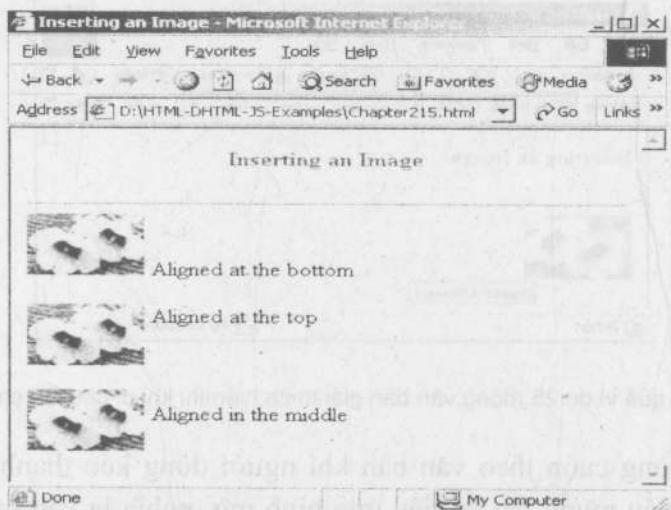
```
<HTML>
  <HEAD>
    <TITLE>Inserting an Image</TITLE>
  </HEAD>
  <H1><FONT SIZE=3 COLOR=FORESTGREEN>
    <B>Inserting an Image</B></FONT></H1>
    <HR>
    <BODY>
      <IMG ALIGN=BOTTOM SRC=Flowers.jpg> Aligned at the
      bottom</P>
      <P>
```

```

<IMG ALIGN=TOP SRC=Flowers.jpg> Aligned at the
top</P>
<P>
<IMG ALIGN=MIDDLE SRC=Flowers.jpg> Aligned in the
middle</P>
</BODY>
</HTML>

```

**Chú ý :** file Flowers.jpg nằm trong cùng thư mục với file HTML nguồn.



Hình 2.30. Kết quả ví dụ 24

**Chú ý :** Một vài trình duyệt không hiển thị những ảnh đồ họa. Trong trường hợp này, chúng ta cần xác định một dòng chú thích thay thế trong tài liệu HTML. Thuộc tính ALT được sử dụng để chỉ nội dung ảnh của bạn.

### Ví dụ 25

```

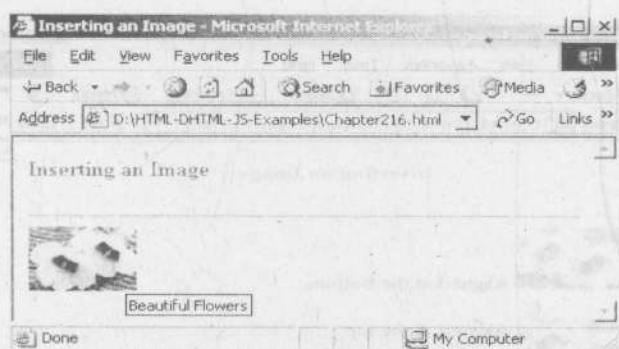
<HTML>
  <HEAD>
    <TITLE>Inserting an Image</TITLE>
  </HEAD>
  <H1><FONT SIZE=3 COLOR=FORESTGREEN>
    <B>Inserting an Image</B>
  </FONT></H1>
  <HR>
  <BODY>
    <IMG SRC=Flowers.jpg ALT="Beautiful Flowers">
  </BODY>
</HTML>

```

Nền của trang cũng quan trọng như văn bản. Người ta thường sử dụng màu cho trang web, mục đích là làm cho nội dung được nổi bật. Chúng ta cũng có thể sử dụng ảnh để làm nền. Để làm điều đó, cần phải dùng thuộc tính BACKGROUND trong thẻ BODY.

<BODY BACKGROUND=bgimage.gif> (Chỉ ra URL hoàn chỉnh của ảnh)

Nếu ảnh nhỏ hơn phạm vi hiển thị của tài liệu thì ảnh được xếp kề nhau để lấp đầy toàn bộ vùng hiển thị.



Hình 2.31. Kết quả ví dụ 25 (dòng văn bản giải thích hiển thị khi di chuột vào ảnh).

Ảnh thường cuộn theo văn bản khi người dùng kéo thanh cuộn trong trình duyệt. Nếu muốn tạo ra hiệu ứng hình mờ, nghĩa là văn bản cuộn còn ảnh đứng yên, ta thiết lập thuộc tính BGPROPERTIES trong thẻ BODY có giá trị là FIXED.

<BODY BACKGROUND=bgimage.gif BGPROPERTIES=FIXED>

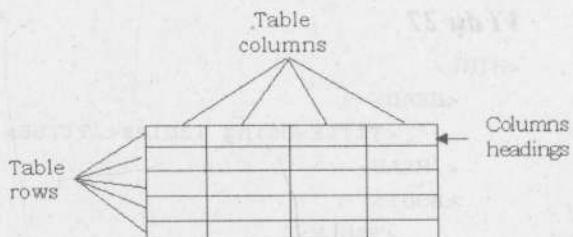
Các ảnh được chèn vào tài liệu HTML cũng có thể sử dụng như siêu liên kết. Những ảnh như vậy được gọi là siêu ảnh. Khi người dùng kích vào ảnh, sẽ hiển thị tài liệu hoặc file được chỉ ra trong URL. Để làm điều này, ta cần lồng ảnh vào trong thẻ neo (anchor).

## 6. SỬ DỤNG BẢNG VÀ LỚP

Phần trước đã hướng dẫn cách nhóm các dữ liệu có liên quan vào các danh sách, còn có một cách khác để điều khiển việc hiển thị văn bản trên trang web, đó là dùng các bảng. Trong phần này sẽ trình bày cách tạo bảng và thảo luận về các lớp.

## 6.1. Cách tạo bảng

Có thể sử dụng bảng để hiển thị dữ liệu dưới dạng các hàng và các cột. Bảng giúp điều khiển, xác định và sắp xếp vị trí của văn bản và hình ảnh trên trang web, thay vì giao cho trình duyệt thực hiện.



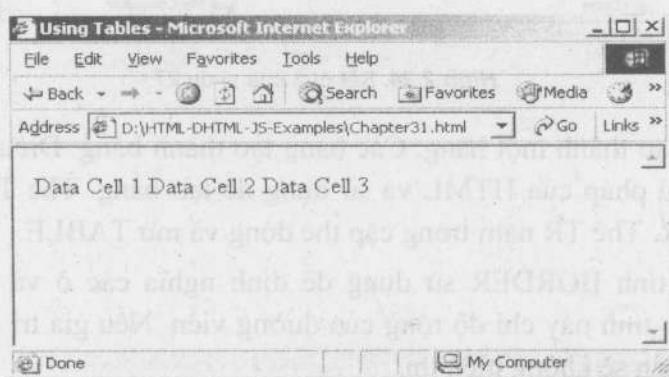
Hình 2.32. Mô hình của bảng

### 6.1.1. Thẻ tạo bảng

Thẻ <TABLE> được dùng để tạo bảng trong tài liệu HTML. Các thuộc tính của phần tử <TABLE> áp dụng cho bảng, nhưng không cho dữ liệu hiển thị trên bảng. Đơn vị cơ bản của bảng là một ô và được định nghĩa bằng thẻ <TD>.

#### Ví dụ 26

```
<HTML>
  <HEAD>
    <TITLE>Using Tables</TITLE>
  </HEAD>
  <BODY>
    <TABLE>
      <TD>Data Cell 1</TD>
      <TD>Data Cell 2</TD>
      <TD>Data Cell 3</TD>
    </TABLE>
  </BODY>
</HTML>
```

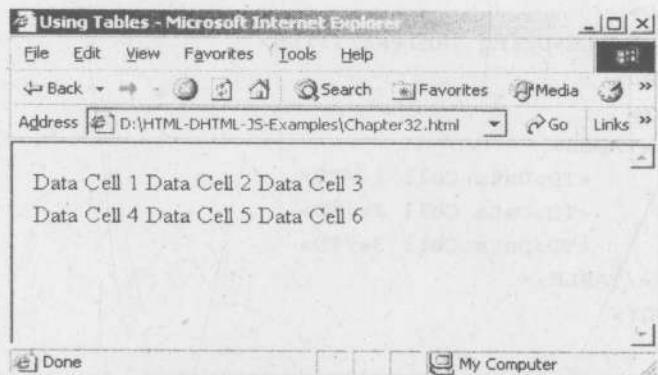


Hình 2.33. Kết quả của ví dụ 26

Một hàng của bảng được định nghĩa bằng thẻ <TR>

### Ví dụ 27

```
<HTML>
  <HEAD>
    <TITLE>Using Tables</TITLE>
  </HEAD>
  <BODY>
    <TABLE>
      <TR>
        <TD>Data Cell 1</TD>
        <TD>Data Cell 2</TD>
        <TD>Data Cell 3</TD>
      <TR>
        <TD>Data Cell 4</TD>
        <TD>Data Cell 5</TD>
        <TD>Data Cell 6</TD>
    </TABLE>
  </BODY>
</HTML>
```



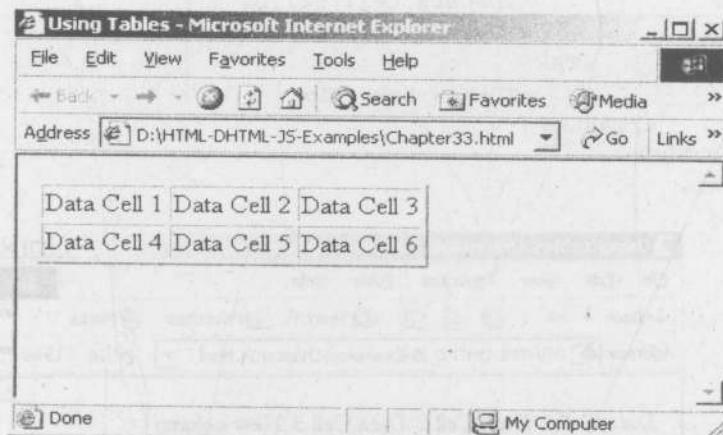
Hình 2.34. Kết quả của ví dụ 27

Các ô tạo thành một hàng. Các hàng tạo thành bảng. Điều này được đề cập trong cú pháp của HTML và sử dụng để tạo bảng. Thẻ TD được lồng trong thẻ TR. Thẻ TR nằm trong cặp thẻ đóng và mở TABLE.

Thuộc tính BORDER sử dụng để định nghĩa các ô và cấu trúc của bảng. Thuộc tính này chỉ độ rộng của đường viền. Nếu giá trị được đặt là 0 thì đường viền sẽ không hiển thị.

### Ví dụ 28

```
<HTML>
  <HEAD>
    <TITLE>Using Tables</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER = 2>
      <TR>
        <TD>Data Cell 1</TD>
        <TD>Data Cell 2</TD>
        <TD>Data Cell 3</TD>
      <TR>
        <TD>Data Cell 4</TD>
        <TD>Data Cell 5</TD>
        <TD>Data Cell 6</TD>
    </TABLE>
  </BODY>
</HTML>
```



Hình 2.35. Kết quả của ví dụ 28

Thẻ <TH> được sử dụng để định nghĩa tiêu đề cho mỗi cột trong bảng. Ví dụ:

```
<TH> Employee Name </TH>
```

Phần tử CAPTION dùng để thêm vào một chú thích cho bảng. Ở đây ta sử dụng nó để mô tả bảng. Ví dụ :

```
<CAPTION> Creating a Table </CAPTION>
```

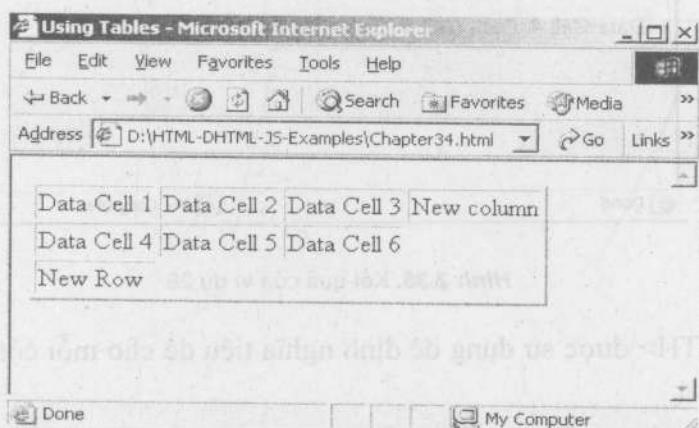
```
<CAPTION> nằm ngay sau thẻ mở <TABLE>
```

### 6.1.2. Chèn hàng và cột

Những thẻ <TD> và <TR> được dùng để chèn theo thứ tự các hàng và các cột vào trong bảng.

#### Ví dụ 29

```
<HTML>
  <HEAD>
    <TITLE>Using Tables</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER = 2>
      <TR>
        <TD>Data Cell 1</TD>
        <TD>Data Cell 2</TD>
        <TD>Data Cell 3</TD>
        <TD>New column</TD>
      <TR>
        <TD>Data Cell 4</TD>
        <TD>Data Cell 5</TD>
        <TD>Data Cell 6</TD>
      <TR>
        <TD>New Row</TD>
    </TABLE>
  </BODY>
</HTML>
```



Hình 2.36. Kết quả của ví dụ 29

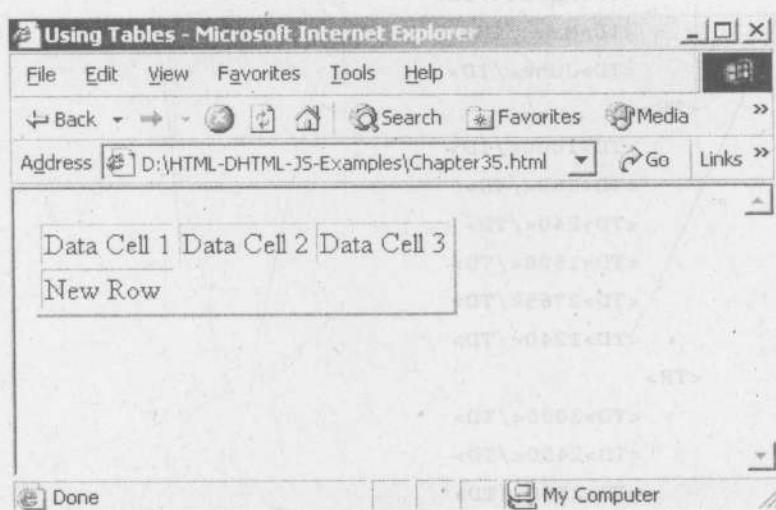
### 6.1.3. Xoá hàng và cột

Để xoá một hàng, xoá thẻ <TR> tương ứng cùng với tất cả các thẻ <TD> bên trong nó từ một file mã nguồn HTML. Trong hình 2.36, để xoá

dòng thứ hai, cần xoá thẻ <TD> cho ô dữ liệu 4, 5 và 6 cùng với thẻ <TR> mà nó kèm theo những thẻ <TD> này từ file HTML nguồn. Để xoá cột cuối cùng, đó là *New column*, chúng ta cần xoá thẻ <TD> lập nên ô này.

### Ví dụ 30

```
<HTML>
<HEAD>
    <TITLE>Using Tables</TITLE>
</HEAD>
<BODY>
    <TABLE BORDER = 2>
        <TR>
            <TD>Data Cell 1</TD>
            <TD>Data Cell 2</TD>
            <TD>Data Cell 3</TD>
        <TR>
            <TD>New Row</TD>
        </TABLE>
    </BODY>
</HTML>
```



Hình 2.37. Kết quả của ví dụ 30

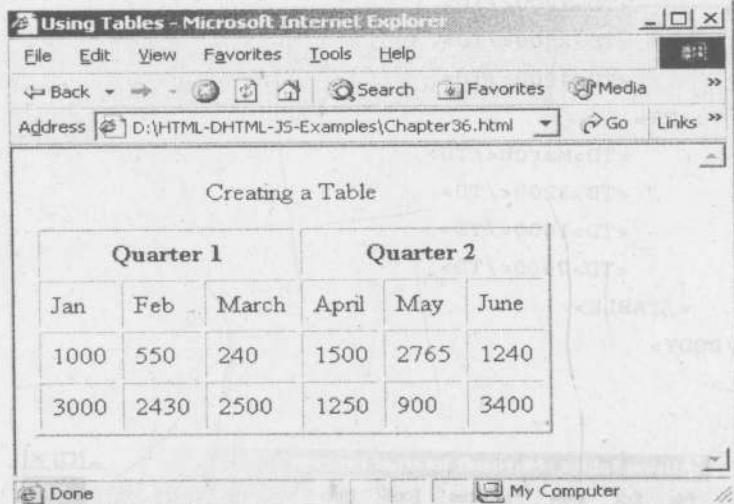
#### 6.1.4. Trộn ô : Kết hợp các cột hay dòng

Khi cần nối các dòng và các cột vào trong một ô, tức là tạo ra một cột rộng hơn, hay tạo ra một dòng rộng hơn, ta sử dụng thuộc tính COLSPAN

và ROWSPAN. Hai thuộc tính này được sử dụng để tạo ra những ô mà chúng có thể kéo rộng ra cho hơn một dòng hay cột. Thuộc tính COLSPAN sử dụng với thẻ <TH>, thuộc tính ROWSPAN sử dụng với thẻ <TD>.

### Ví dụ 31

```
<HTML>
  <HEAD>
    <TITLE>Using Tables</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER = 2 BGCOLOR = lavender CELLSPACING = 2
      CELLPADDING = 6>
      <CAPTION>Creating a Table</CAPTION>
      <TH ALIGN = CENTER COLSPAN = 3>Quarter 1</TH>
      <TH ALIGN = CENTER COLSPAN = 3>Quarter 2</TH>
      <TR>
        <TD>Jan</TD>
        <TD>Feb</TD>
        <TD>March</TD>
        <TD>April</TD>
        <TD>May</TD>
        <TD>June</TD>
      <TR>
        <TD>1000</TD>
        <TD>550</TD>
        <TD>240</TD>
        <TD>1500</TD>
        <TD>2765</TD>
        <TD>1240</TD>
      <TR>
        <TD>3000</TD>
        <TD>2430</TD>
        <TD>2500</TD>
        <TD>1250</TD>
        <TD>900</TD>
        <TD>3400</TD>
      </TABLE>
    </BODY>
  </HTML>
```



Hình 2.38. Kết quả của ví dụ 31

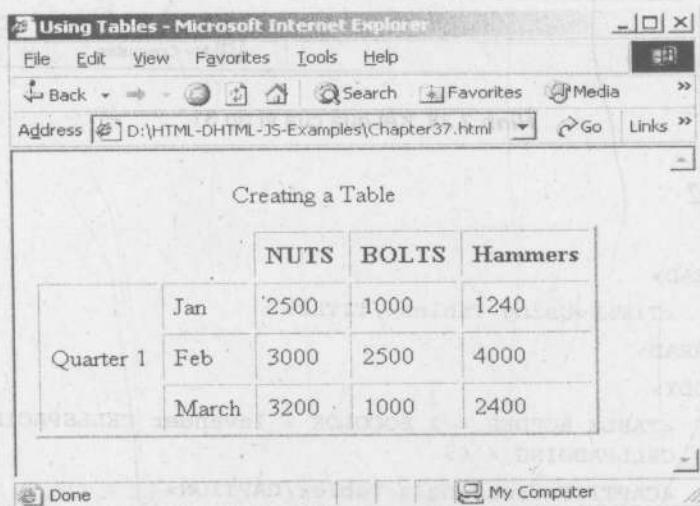
### Ví dụ 32

```
<HTML>
  <HEAD>
    <TITLE>Using Tables</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER = 2 BGCOLOR = lavender CELLSPACING = 2
      CELLPADDING = 6>
      <CAPTION>Creating a Table</CAPTION>
      <TR>
        <TH></TH>
        <TH></TH>
        <TH>NUTS</TH>
        <TH>BOLTS</TH>
        <TH>Hammers</TH>
      <TR>
        <TD ROWSPAN = 3>Quarter 1</TD>
        <TD>Jan</TD>
        <TD>2500</TD>
        <TD>1000</TD>
        <TD>1240</TD>
      <TR>
        <TD>Feb</TD>
```

```

<TD>3000</TD>
<TD>2500</TD>
<TD>4000</TD>
<TR>
<TD>March</TD>
<TD>3200</TD>
<TD>1000</TD>
<TD>2400</TD>
</TABLE>
</BODY>
</HTML>

```



Hình 2.39. Kết quả của ví dụ 32

### 6.1.5. Định dạng cho ô

Có thể chỉ định được vị trí của văn bản bên trong mỗi ô của bảng. Những thuộc tính canh lề ngang (ALIGN) và canh lề dọc (VALIGN) được sử dụng để điều khiển việc canh lề trong các ô của bảng. Thuộc tính ALIGN có thể là các giá trị: trái, phải, giữa, đều hai bên. Thuộc tính VALIGN có thể là các giá trị: trên, giữa và dưới.

Ngoài việc xác định vị trí nội dung của các ô trong bảng, chúng ta cũng có thể thay đổi kích thước, màu sắc của bảng. Để xác định chiều rộng của bảng – là vùng trên màn hình mà bảng sẽ mở rộng ra, ta sử dụng thuộc tính

WIDTH của thẻ TABLE. Giá trị của thuộc tính WIDTH có thể là một định dạng phần trăm hay thậm chí là các điểm. Ví dụ :

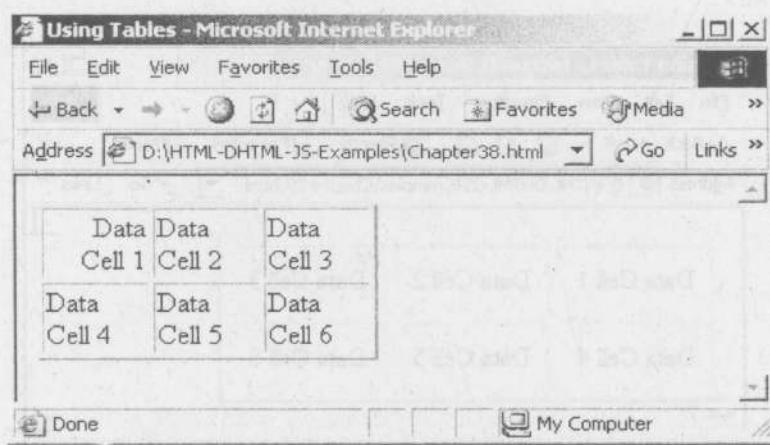
```
<TABLE WIDTH = 50%>
```

Lệnh này sẽ kéo rộng bảng đến 50% vùng màn hình.

Nếu muốn thêm màu sắc cho bảng, chúng ta sử dụng thuộc tính BGCOLOR. Giá trị này có thể là tên của một màu hay là giá trị ở hệ thập lục phân.

### Ví dụ 33

```
<HTML>
<HEAD>
    <TITLE>Using Tables</TITLE>
</HEAD>
<BODY>
    <TABLE BORDER = 2 WIDTH = 50% BGCOLOR = lavender>
        <TR>
            <TD ALIGN = right VALIGN = bottom>Data Cell 1</TD>
            <TD>Data Cell 2</TD>
            <TD>Data Cell 3</TD>
        <TR>
            <TD>Data Cell 4</TD>
            <TD>Data Cell 5</TD>
            <TD>Data Cell 6</TD>
    </TABLE>
</BODY>
</HTML>
```

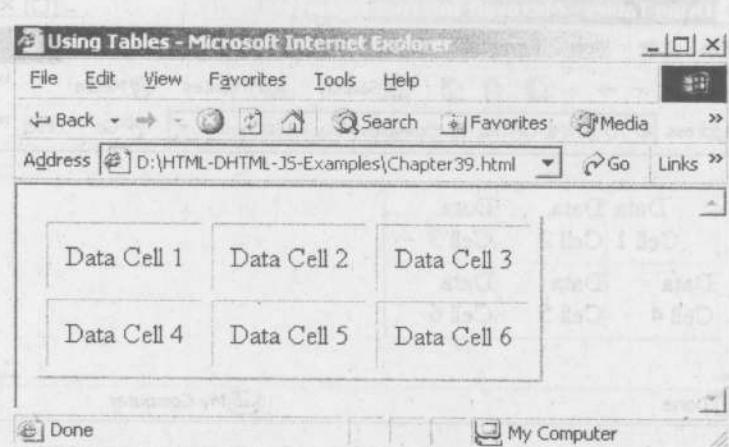


Hình 2.40. Kết quả của ví dụ 33

Nếu nội dung của văn bản trong ô nhỏ hơn không gian trong ô đó thì sẽ có không gian trống là khoảng cách giữa văn bản và đường viền của ô. CELLPACING là không gian giữa các ô và CELLPADDING là không gian giữa đường viền ô của bảng với nội dung văn bản được đặt trong ô. Như đã mô tả ở trên, thuộc tính CELLPACING định nghĩa không gian giữa các ô, tính theo điểm (pixel). Thuộc tính CELLPADDING định nghĩa những không gian bên trong ô của bảng.

#### Ví dụ 34

```
<HTML>
  <HEAD>
    <TITLE>Using Tables</TITLE>
  </HEAD>
  <BODY>
    <TABLE BORDER = 2 BGCOLOR = lavender CELLPACING = 5
           CELLPADDING = 10>
      <TR>
        <TD>Data Cell 1</TD>
        <TD>Data Cell 2</TD>
        <TD>Data Cell 3</TD>
      <TR>
        <TD>Data Cell 4</TD>
        <TD>Data Cell 5</TD>
        <TD>Data Cell 6</TD>
    </TABLE>
  </BODY>
</HTML>
```



Hình 2.41. Kết quả của ví dụ 34

## 6.2. Lớp (Layer)

Khi đặt vị trí cho một phần tử trên trang Web, cần xác định toạ độ x và y. Ngoài ra, cũng có thể xác định toạ độ z. Khi đó, trục toạ độ z là một đường ảo từ phía trước đến phía sau màn hình. Vì vậy, một phần tử có thể được đặt ở trên hay dưới phần tử khác trong lớp. Toạ độ z xác định thứ tự trong đó các phần tử được hiển thị.

Khái niệm về lớp có thể được đưa ra đối với một tài liệu HTML. Một lớp giống như một frame mà nó có thể chứa nội dung và được chỉ định vị trí ở nơi được yêu cầu. Trong phần tiếp theo sẽ trình bày kỹ hơn về lớp.

### 6.2.1. Lớp là gì?

Lớp là phần tử được thêm vào một trang Web. Chúng ta có thể xác định chính xác vị trí của một lớp trong trang Web. Netscape cung cấp một phần tử <LAYER> được sử dụng để xác định vị trí nội dung trên trang Web.

Lớp có tập thuộc tính riêng, có thể được truy xuất và lập trình. Các lớp được lập trình thông qua script (tức là đoạn mã của một chương trình nhỏ) để tạo ra những trang động hay những trang có khả năng tương tác với nhau. Vì vậy, chúng ta sẽ xác định được vị trí các phần tử trên trang mà ta cần, thay vì phải di chuyển nó đến *phản chức năng* (functionality) được yêu cầu bởi mỗi trình duyệt.Thêm vào đó, chúng ta có thể kết nối các văn bản và hình ảnh lại với nhau để tạo ra một trang web sinh động và dễ đọc. Các hiệu ứng đặc biệt có thể được thêm vào trên một trang web mà không cần phải tăng kích thước của file lên nhiều hoặc thay đổi các phần của trang mà không phải thay đổi lại, thiết kế lại trang chủ, như vậy, chúng ta sẽ thay những nội dung của một tầng đặc biệt mà không phải hiệu chỉnh trang chủ.

Internet Explorer và Netscape Navigator đều cung cấp công nghệ lớp nhưng mỗi nhà cung cấp lại theo một cách riêng. Internet Explorer hỗ trợ mô hình CSS (Cascading Style Sheet – Mô hình thác nước). Khi sử dụng những kiểu Bảng kiểu như vậy thì chúng ta có thể nhóm các thành phần của trang lại với nhau trong một Bảng kiểu hình thác nước. Bảng kiểu loại này được đưa ra để chọn một hay tất cả các phần tử trong một tài liệu HTML. Những phần tử được điều khiển thông qua VBScript hay Java Script. Để tạo ra một tầng tài liệu, cần phải thay đổi thuộc tính của kiểu Bảng kiểu này.

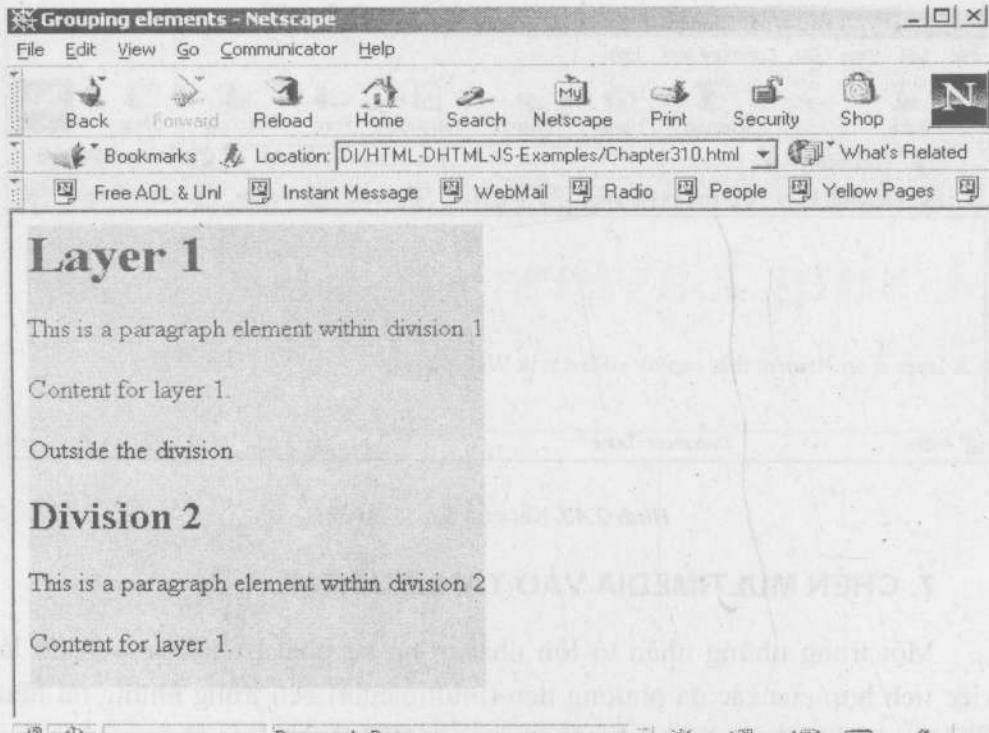
Netscape Navigator cũng hỗ trợ việc sử dụng CSS, nhưng theo một cách khác, đó là phải sử dụng thẻ <LAYER> do Netscape hỗ trợ.

### 6.2.2. Sử dụng các lớp

Lớp ở đây được hiểu như là một thực thể đơn của nội dung và được coi giống như bất kỳ một kiểu nào khác mà nó chỉ đến những phần tử bên trong một trang HTML. Vì vậy, nếu chúng ta muốn ứng dụng kiểu lớp cho tất cả những phần tử bên trong lớp, chúng ta có thể sử dụng những phần tử như DIV hay SPAN.

#### Ví dụ 35

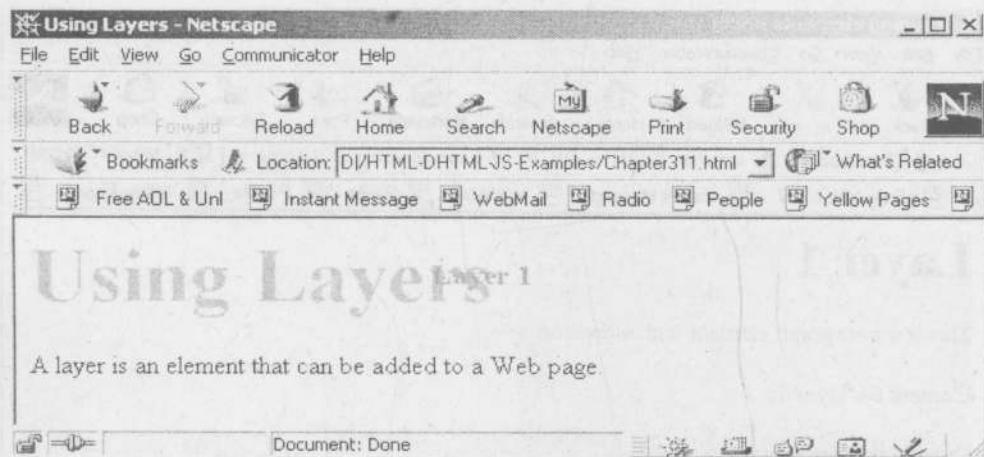
```
<HTML>
    <HEAD>
        <TITLE>Grouping elements</TITLE>
    </HEAD>
    <BODY BGCOLOR=lavender>
        <LAYER id = layer1 bgcolor=aqua>
            <DIV><font color = red>
                <H1>Layer 1</H1>
                <P>This is a paragraph element within
                    division 1</P>
                <P>Content for layer 1.</P></FONT>
            </DIV>
            <P>Outside the division</P>
            <DIV> <font color=blue>
                <H2>Division 2</H2>
                <P>This is a paragraph element within
                    division 2</P>
                <P>Content for layer 1.</P></FONT>
            </DIV>
        </LAYER>
    </BODY>
</HTML>
```



Hình 2.42. Kết quả của ví dụ 35

### Ví dụ 36

```
<HTML>
  <HEAD>
    <TITLE>Using Layers</TITLE>
  </HEAD>
  <H1><Font color = hotpink size = 7>Using
Layers</FONT></H1>
  <BODY text = "blue">
    <LAYER top = 25 ID=layer1 visibility=show>
      <P><Center><Font color = limegreen size =
4><B>Layer 1</B></FONT></Center></P>
    </LAYER>
    <P>
      A layer is an element that can be added to a Web page.
    </P>
  </BODY>
</HTML>
```



Hình 2.43. Kết quả của ví dụ 36

## 7. CHÈN MULTIMEDIA VÀO TÀI LIỆU HTML

Một trong những nhân tố lớn nhất trong sự phát triển của web đó là việc tích hợp của các đa phương tiện (multimedia) bên trong những tài liệu HTML. Khi những phiên bản đầu tiên của HTML ra đời, đã bao gồm các đối tượng liên quan đến hình ảnh ở trong một tài liệu để cho phép nhúng hình ảnh nội tuyến vào nội dung tài liệu. Sau đó, HTML được mở rộng để cho phép nhúng không chỉ là những ảnh tĩnh mà còn là âm thanh và video. Hiện nay, tất cả những tính năng đó đã được sử dụng để làm phong phú thêm cho các trang web.

### 7.1. Chèn ảnh động (.GIF) vào tài liệu HTML

Mục 5 (chương 2) đã trình bày về khái niệm file GIF và cách đưa file GIF vào một tài liệu HTML. Cần nhớ rằng, thẻ **<IMG>** được dùng để chèn một ảnh vào trong trang web.

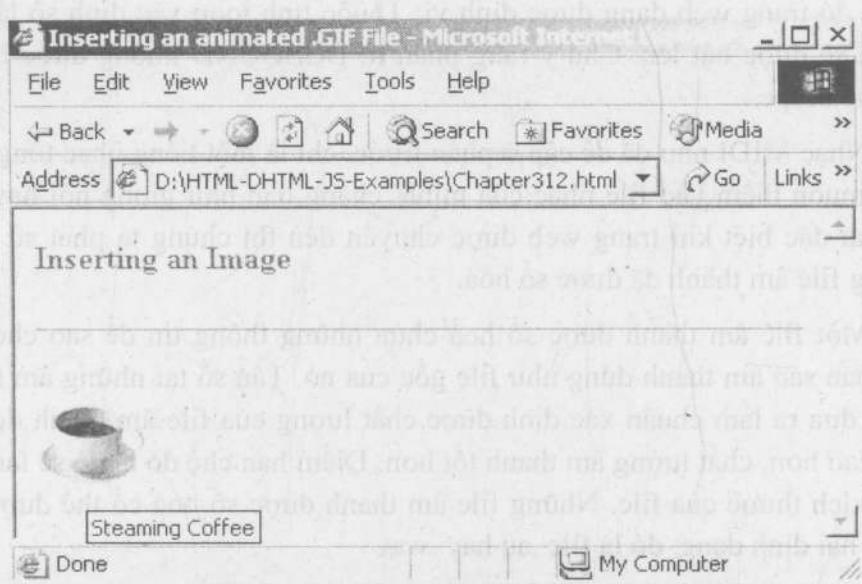
#### Ví dụ 37

```
<HTML>
  <HEAD>
    <TITLE>Inserting an animated .GIF File</TITLE>
  </HEAD>
  <H1><FONT SIZE=3 COLOR=FORESTGREEN><B>Inserting an
  Image</B></FONT></H1>
```

```

<HR>
<BODY>
    <IMG SRC=coffeecup.gif ALT="Steaming Coffee">
</BODY>
</HTML>

```



Hình 2.44. Kết quả của ví dụ 37

**Chú ý :** Trong ví dụ 37, coffeecup.gif là một ảnh động có định dạng file.gif, khi hiển thị trong trình duyệt thì nó hiển thị một luồng khói phía trên cốc cà phê. File này được đặt vào cùng thư mục với file HTML nguồn. Tuy nhiên, nó có thể được thay thế bởi các hình ảnh động khác.

## 7.1. Chèn âm thanh vào tài liệu HTML

Tính hấp dẫn của một trang Web tương tác đó là thường sử dụng tiếng “bip” khi người dùng kích hoạt đến một trang web khác. Một cách tùy chọn, một trang web sẽ yêu cầu kết hợp tiếng nhạc để làm cho trang web đó lôi cuốn hơn. MIDI là một định dạng chuẩn của các file nhạc được dùng trong tài liệu HTML. Các file MIDI chứa những nốt nhạc và các loại nhạc cụ cho các bản nhạc. Nhạc cụ điện tử trong card âm thanh mô phỏng tiếng nhạc. Nói cách khác, các file .wav và .au dùng để lưu âm thanh.

Để thêm âm thanh cho trang web, cần sử dụng các file âm thanh (.wav hay .midi) trên hệ thống. Ví dụ :

```
<BGSOUND SRC="path\sound filename" loop="positive number/infinite">
```

Nếu chúng ta không xác định được đường dẫn thì trình duyệt sẽ tìm file mà ở đó trang web đang được định vị. Thuộc tính loop xác định số lần âm thanh sẽ được bật lên. Chú ý rằng phần tử BGSOUND không được hỗ trợ bởi Netscape.

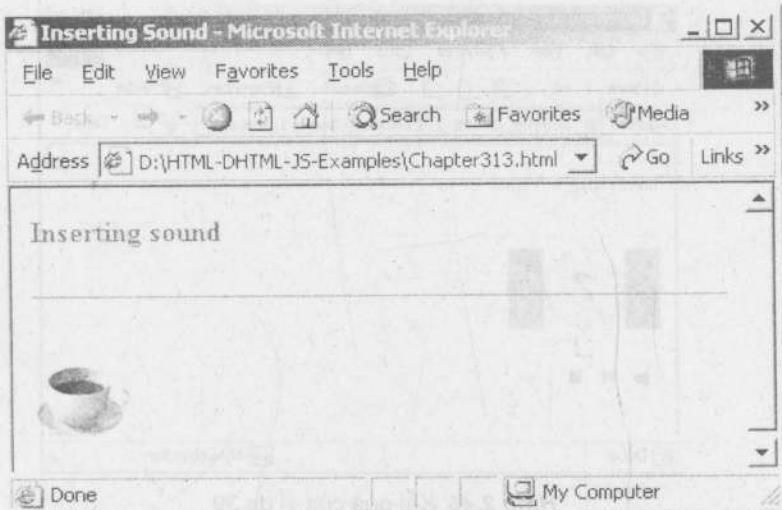
Nhạc MIDI như đã đề cập ở phần trước, chỉ là một tiếng nhạc tổng hợp. Nếu muốn thêm vào file nhạc của mình, chẳng hạn như giọng nói hay một bài hát đặc biệt khi trang web được chuyển đến thì chúng ta phải sử dụng những file âm thanh đã được số hoá.

Một file âm thanh được số hoá chứa những thông tin để sao chép lại một bản sao âm thanh đúng như file gốc của nó. Tần số tại những âm thanh được đưa ra làm chuẩn xác định được chất lượng của file âm thanh đó, tần suất cao hơn, chất lượng âm thanh tốt hơn. Điểm hạn chế đó là nó sẽ làm tăng kích thước của file. Những file âm thanh được số hoá có thể được lưu trong hai định dạng, đó là file .au hay .wav

Định dạng file .wav bắt đầu với hệ điều hành Windows. Nó có tỷ lệ nén thấp nhưng kích thước file lại rất lớn. Còn định dạng file .au được tổ chức với những máy cài hệ điều hành Unix, sẽ tốt hơn cho việc lưu trữ các file âm thanh và nó có một tỷ lệ nén cao hơn so với những file .wav.

### Ví dụ 38

```
<HTML>
  <HEAD>
    <TITLE>Inserting Sound</TITLE>
  </HEAD>
  <H1><FONT SIZE=3 COLOR=FORESTGREEN><B>Inserting
sound</B></FONT></H1>
  <HR>
  <BODY>
    <BGSOUND src="WindowsLogonSound.wav" loop="infinite">
    <IMG SRC=coffeecup.gif ALT="Steaming Coffee">
  </BODY>
</HTML>
```



Hình 2.45. Kết quả của ví dụ 38

**Chú ý :** Trong ví dụ 38, khi hiển thị trang web trong trình duyệt file nhạc đăng nhập Windows sẽ chạy. File windowLogonSound.wav được đặt trong cùng thư mục với file HTML nguồn. Tuy nhiên, nó có thể được thay thế bởi các file nhạc khác.

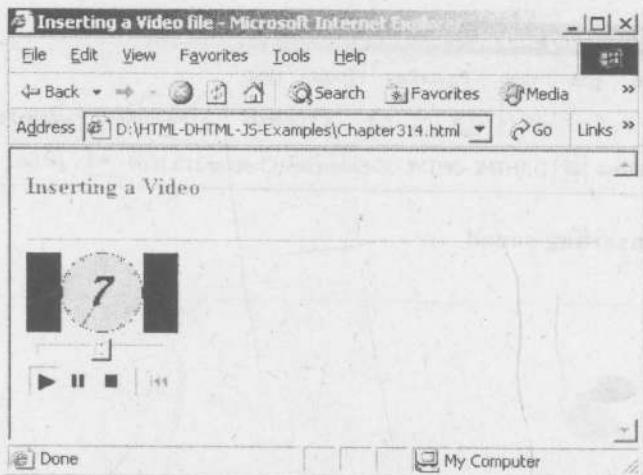
### 7.3. Chèn Video vào tài liệu HTML

Một file video thường có phần mở rộng là: .avi, .asf, .ram hay là .ra. Để chèn một file video vào tài liệu HTML, sử dụng thẻ <EMBED>. Ví dụ :

```
<EMBED SRC="path\video file name" WIDTH="width in pixels or percentage" HEIGHT="height in pixels or percentage">
```

#### Ví dụ 39

```
<HTML>
  <HEAD>
    <TITLE>Inserting a Video file</TITLE>
  </HEAD>
  <H1><FONT SIZE=3 COLOR=FORESTGREEN><B>Inserting
Video</B></FONT></H1>
  <HR>
  <BODY>
    <EMBED SRC=clock.avi HEIGHT=100 WIDTH=100>
  </BODY>
</HTML>
```



Hình 2.46. Kết quả của ví dụ 39

**Chú ý :** Để xem kết quả 39, trình Media Player phải được cài đặt. File clock. avi có thể được thay thế bởi các file video khác.

#### 7.4. Chèn các Java Applets

Bản thân trang HTML rất thụ động. Điều này có nghĩa là nội dung của chúng ít nhiều là tĩnh và không thay đổi trong một khoảng thời gian. Có một vài cách để tạo các chương trình tương tác trên Web, một trong những cách đáng chú ý nhất là sử dụng applet. Applet là một chương trình được viết trong một ngôn ngữ giống như Java, đặc biệt là được thiết kế để làm việc trên Internet thông qua một trình duyệt Web. Một applet có thể sử dụng cho các mục đích khác nhau, từ những phương tiện giao tiếp dựa trên Web cho đến những việc thiết kế các giao diện đồ họa người dùng cho các ứng dụng back-end. Với việc sử dụng applet, chúng ta có thể tạo ra các trò chơi đa phương tiện (multimedia game), các cuộc thi và các chương trình tương tác kiểu như vậy cho Web.

Một applet có thể được nhúng trong một trang HTML và được thực thi trên một trình duyệt Web có hỗ trợ Java. Ví dụ, các trình duyệt Web có hỗ trợ Java là Internet Explorer 3.0 hoặc Netscape Navigator 3.0.

Để hiển thị một applet, cần tạo một trang HTML mà nó có thể nạp các file applet vào trong một trình duyệt.

Ví dụ :

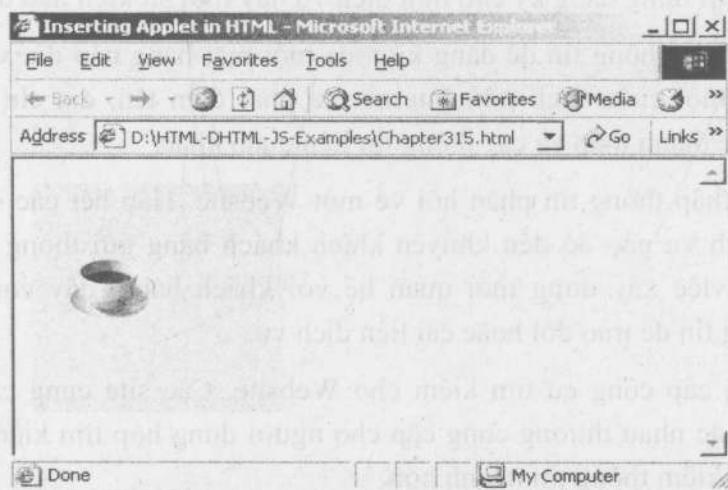
```
<applet code = FirstApplet width=200 height= 200></applet>
```

File HTML sử dụng thẻ Applet. Thẻ applet lần lượt trả đến đường dẫn của file applet như tham số đầu tiên của nó. Thẻ applet cũng có hai thuộc tính, đó là chiều rộng và chiều cao. Hai thuộc tính này sẽ xác định hướng của applet trên màn hình của trình duyệt.

Cũng có thể thông qua các tham số để đến được một applet bằng cách sử dụng thẻ <PARAM> trong file HTML. Tên của tham số được xác định bằng cách sử dụng thẻ <PARAM> và giá trị của nó được xác định bằng cách sử dụng thuộc tính giá trị trong bên trong thẻ. Ví dụ sau đây chỉ ra cách để file ảnh được chuyển qua như là một tham số thông qua một file HTML đến một Applet.

#### Ví dụ 40

```
<HTML>
  <HEAD>
    <TITLE> Inserting Applet in HTML </TITLE>
  </HEAD>
  <BODY>
    <APPLET code = AppletDemo width = 150 height = 150>
      <PARAM name = "image" value = "coffeecup.gif">
    </APPLET>
  </BODY>
</HTML>
```



Hình 2.47. Kết quả của ví dụ 40

**Chú ý :** Trong ví dụ 40, AppletDemo là một file Java Applet. File này có thể được thay thế bởi các file java applet khác.

## **8. SỬ DỤNG BIỂU MẪU VÀ KHUNG**

Người sử dụng Web thường xuyên sẽ hay gặp các loại biểu mẫu khác nhau và nhiều khi phải điền thông tin vào các loại biểu mẫu đó. Nội dung chương này sẽ trình bày cách thêm các phần tử nhập vào một tài liệu HTML và tạo ra một biểu mẫu. Ngoài ra, sẽ trình bày thêm về khái niệm khung để phân chia tài liệu.

### **8.1. Giới thiệu biểu mẫu**

Form HTML là một phần của tài liệu, nó chứa các phần tử đặc biệt gọi là các điều khiển. Các điều khiển được sử dụng để nhập thông tin từ người dùng và cung cấp một số tương tác. Dữ liệu do người dùng nhập vào có thể được xác nhận hợp lệ nhờ các kịch bản phía máy khách (client-side scripts) và được chuyển đến máy chủ để xử lý thêm.

#### ***8.1.1. Sử dụng biểu mẫu***

Việc sử dụng biểu mẫu trên World Wide Web khá nhiều và liên tục tăng lên. Sau đây là một số cách sử dụng thông thường :

- Thu thập tên, địa chỉ, số điện thoại, địa chỉ e-mail và các thông tin khác để người dùng đăng ký cho một dịch vụ hay một sự kiện nào đó.
- Thu thập thông tin để đăng ký mua một mặt hàng nào đó, ví dụ, khi muốn mua một cuốn sách trên Internet, ta phải điền tên, địa chỉ gửi thư, phương thức thanh toán và các thông tin liên quan khác.
- Thu thập thông tin phản hồi về một Website. Hầu hết các site cung cấp một dịch vụ nào đó đều khuyến khích khách hàng gửi thông tin phản hồi. Ngoài việc xây dựng mối quan hệ với khách hàng, đây còn là một nguồn thông tin để trao đổi hoặc cải tiến dịch vụ.
- Cung cấp công cụ tìm kiếm cho Website. Các site cung cấp nhiều thông tin khác nhau thường cung cấp cho người dùng hộp tìm kiếm để cho phép họ tìm kiếm thông tin nhanh hơn.

Một biểu mẫu điển hình trên trang web như sau :

**Sample Form - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address D:\HTML-DHTML-25-Examples\DemoForm.html

**Sample Stock Survey**

Describe your investment experience

beginner  intermediate  expert

Types of Investments you make

Individual Stocks  Options  Mutual Funds

How do you buy your stocks?

What is your hot stock pick for

Any Investment Advice for others?

Submit Reset

Done My Computer

Hình 2.48. Một dạng biểu mẫu

### 8.1.2. Phần tử FORM

Phần tử <FORM> được sử dụng để tạo một vùng trên trang như một biểu mẫu. Nó chỉ ra cách bố trí của biểu mẫu. Các thuộc tính bao gồm:

Thuộc tính	Mô tả
ACCEPT	Thuộc tính này xác định danh sách các kiểu MIME được máy chủ nhận ra, trong đó có chứa kịch bản (script) để xử lý biểu mẫu. Cú pháp là ACCEPT = "Internet media type".
ACTION	Thuộc tính này xác định vị trí của script sẽ xử lý biểu mẫu hoàn chỉnh và đã được gửi đi Cú pháp là: ACTION = "URL"
METHOD	Thuộc tính này xác định phương thức dữ liệu được gửi đến máy chủ. Nó cũng xác định giao thức được sử dụng khi máy khách gửi dữ liệu lên cho máy chủ. Nếu giá trị là GET thì trình duyệt sẽ tạo một câu hỏi có chứa địa chỉ URL của trang, một dấu chấm hỏi và các giá trị do biểu mẫu tạo ra. Trình duyệt sẽ trả lại câu hỏi cho kịch bản được xác định trong URL để xử lý. Nếu giá trị là POST, thì dữ liệu trên biểu mẫu được gửi đến kịch bản xử lý như một khối dữ liệu. Người ta không sử dụng chuỗi câu hỏi. Cú pháp là METHOD = (GET   POST).

Ví dụ, để đưa một biểu mẫu đến chương trình “xử lý biểu mẫu” sử dụng theo phương thức POST

```
<FORM action="http://mysite.com/processform" METHOD="post">  
...form contents...  
</FORM>
```

### 8.1.3. Các phần tử nhập của HTML

Khi tạo một biểu mẫu, ta có thể đặt các điều khiển lên biểu mẫu để nhận dữ liệu nhập vào từ người dùng. Các điều khiển này được sử dụng với phần tử <FORM>. Tuy nhiên, ta cũng có thể sử dụng chúng ở bên ngoài biểu mẫu để tạo các giao diện người dùng.

#### a) Phần tử INPUT

Phần tử <INPUT> xác định loại và sự xuất hiện của điều khiển trên biểu mẫu. Các thuộc tính của phần tử này là :

Thuộc tính	Mô tả
TYPE	Thuộc tính này xác định loại phần tử. Ta có thể chọn một trong các lựa chọn: TEXT, PASSWORD, CHECKBOX, RADIO, SUBMIT, RESET, FILE, HIDDEN và BUTTON. Mặc định là TEXT.
NAME	Thuộc tính này chỉ tên của điều khiển. Ví dụ nếu có nhiều hộp văn bản (text box) trên một biểu mẫu thì bạn nên sử dụng tên để xác định chúng – TEXT1, TEXT2 hoặc bất kỳ tên nào mình chọn. Phạm vi hoạt động của thuộc tính NAME nằm trong phần tử FORM.
VALUE	Đây là một thuộc tính tùy chọn, nó xác định giá trị khởi tạo của điều khiển. Tuy nhiên, đối với kiểu (TYPE) là RADIO thì ta phải xác định cho nó một giá trị.
SIZE	Thuộc tính này xác định độ rộng ban đầu của điều khiển. Đối với kiểu là TEXT hay PASSWORD thì kích thước được xác định theo ký tự. Đối với các loại phần tử nhập khác, độ rộng được xác định bằng điểm (pixels).
maxlength	Thuộc tính này được sử dụng để xác định số ký tự lớn nhất có thể nhập vào phần tử TEXT hoặc PASSWORD. Mặc định là không giới hạn.
checked	Đây là thuộc tính logic để xác định nút có được chọn hay không. Thuộc tính này được sử dụng khi kiểu nhập là RADIO hay CHECKBOX.
src	src = "URL". Thuộc tính này được dùng khi ta muốn sử dụng một ảnh trong kiểu INPUT. Nó xác định vị trí của ảnh.

Phần này sẽ trình bày về các loại phần tử nhập cùng với một số đặc tính và sự kiện thường dùng.

## b) Nút bấm

Loại này tạo ra một điều khiển nút (button)

Tên	Mô tả
NAME	Thiết lập hoặc truy xuất tên của điều khiển
SIZE	Thiết lập hoặc truy xuất kích thước của điều khiển
TYPE	Truy xuất loại điều khiển bên trong được biểu diễn bởi <INPUT type = button>
VALUE	Thiết lập hoặc truy xuất giá trị của nút <INPUT type=button>.

Ví dụ :

```
<INPUT TYPE=button VALUE="click" NAME="b1">
```

## c) Văn bản

Loại này tạo một điều khiển nhập văn bản có một dòng. Thuộc tính SIZE xác định số ký tự có thể hiển thị trong phần tử. Thuộc tính MAXLENGTH xác định số ký tự tối đa có thể nhập vào phần tử này.

Ví dụ :

```
<INPUT TYPE=text VALUE="" NAME="textbox" SIZE=20>
```

Giá trị Value ở đây để hiển thị nội dung ban đầu của văn bản và để truy xuất văn bản khi biểu mẫu được gửi đi.

## d) Checkbox (hộp kiểm)

Loại này tạo ra một điều khiển checkbox. Người dùng có thể chọn một hoặc nhiều checkbox. Khi một phần tử checkbox được chọn, thì cặp tên/giá trị được nhận cùng với biểu mẫu. Giá trị mặc định của checkbox là bật (on). Phần tử checkbox là một phần tử trên dòng và không cần thẻ đóng.

Tên	Mô tả
CHECKED	Thiết lập hoặc truy xuất trạng thái của checkbox
NAME	Thiết lập hoặc truy xuất tên của điều khiển
SIZE	Thiết lập hoặc truy xuất kích thước của điều khiển
STATUS	Thiết lập hoặc truy xuất trạng thái xem checkbox có được chọn hay không
TYPE	Truy xuất loại điều khiển, sử dụng <INPUT type = checkbox>
VALUE	Thiết lập hoặc truy xuất giá trị của checkbox <INPUT type=checkbox>

## e) Radio

Loại này tạo ra điều khiển kiểu nút radio. Một điều khiển kiểu nút radio (radio button control) được sử dụng đối với các tập giá trị loại trừ lẫn nhau.

Các điều khiển radio trong một nhóm phải có *cùng tên*. Vào một thời điểm, người dùng chỉ có thể chọn một lựa chọn. Chỉ có nút radio được chọn trong nhóm mới tạo tên cặp giá trị tên/value trong dữ liệu được nhận. Các nút radio nên đặt thuộc tính **giá trị**.

Tên	Mô tả
CHECKED	Thiết lập hoặc truy xuất trạng thái của nút radio
NAME	Thiết lập hoặc truy xuất tên của điều khiển
SIZE	Thiết lập hoặc truy xuất kích thước của điều khiển
STATUS	Thiết lập hoặc truy xuất trạng thái xem nút radio có được chọn hay không
TYPE	Truy xuất loại điều khiển, sử dụng <INPUT type = radio>
VALUE	Thiết lập hoặc truy xuất giá trị của radio <INPUT type=radio>

Ví dụ :

```
<INPUT TYPE=radio NAME="sex" VALUE="male">Male
```

### g) Submit

Loại này tạo ra một nút submit. Khi người dùng nhấp vào nút Submit, biểu mẫu được chuyển đến vị trí được xác định trong thuộc tính ACTION. Cặp tên/giá trị của nút submit được nhận cùng với biểu mẫu.

Ví dụ,

```
<INPUT TYPE=submit VALUE="click" NAME="b1">
```

### h) Ảnh

Điều khiển này tạo ra một nút submit dạng ảnh. Giá trị của thuộc tính SRC xác định URL của ảnh được đặt trong nút đó. Khi người dùng nhấp vào điều khiển ảnh này, biểu mẫu được chuyển đi để xử lý. Toạ độ x và y (được đo bằng điểm) tại vị trí nhấp chuột sẽ chuyển đến máy chủ với định dạng sau :

```
Name.x = valueofx
```

```
Name.y = valueofy
```

Trong đó, 'name' là tên của điều khiển.

Ta có thể sử dụng nhiều nút submit với các hình ảnh khác nhau thay vì một nút submit chỉ có một hình. Nếu cần trình bày nhiều ảnh ta có thể sử dụng bản đồ ảnh.

Ví dụ :

```
<INPUT TYPE =image SRC="usamap.gif" NAME="name">
```

## k) Reset

Điều khiển này tạo ra nút reset. Khi người dùng nhập vào nút này, các giá trị của tất cả các điều khiển được tái thiết lập trở về giá trị ban đầu, được xác định trong các giá trị thuộc tính của chúng.

Ví dụ :

```
<INPUT TYPE=reset VALUE="Reset" NAME="B2">
```

Ví dụ sau thể hiện việc sử dụng nhiều kiểu nhập khác nhau.

### Ví dụ 41

<HTML>

<HEAD>

<TITLE> Sample Form </TITLE>

</HEAD>

<BODY bgColor="#ffffcc" Text="#000099">

```
<FORM ACTION="http://www.mysite.com/FormSite"
METHOD="POST">
```

<B><H2 align="left">Sample Stock Survey</H2></B>

<p><B>Describe your investment experience</B></p>

<p>

```
<INPUT TYPE="RADIO" NAME="RESULT_RadioButton-3"
VALUE="Radio-0">beginner
```

```
<INPUT TYPE="RADIO" NAME="RESULT_RadioButton-3"
VALUE="Radio-1">intermediate
```

```
<INPUT TYPE="RADIO" NAME="RESULT_RadioButton-3"
VALUE="Radio-2">expert
```

</p>

<p><B>Types of Investments you make</B></P>

<P>

```
<INPUT TYPE="CHECKBOX" NAME="RESULT_CheckBox-4"
VALUE="CheckBox-0">Individual Stocks
```

```
<INPUT TYPE="CHECKBOX" NAME="RESULT_CheckBox-4"
VALUE="CheckBox-1">Options
```

```
<INPUT TYPE="CHECKBOX" NAME="RESULT_CheckBox-4"
VALUE="CheckBox-2">Mutual Funds<BR></p>
```

<p><B>What is your stock pick for this
year?</B></P>

<P>

```

<INPUT TYPE="TEXT" NAME="RESULT_TextField-6"
SIZE="30" MAXLENGTH="30">
</p>
<p>
    <INPUT TYPE="SUBMIT" NAME="Submit"
VALUE="Submit">
    <INPUT TYPE="RESET" NAME="Reset"
VALUE="Reset"></p>
</FORM>
</BODY>
</HTML>

```

The screenshot shows a Microsoft Internet Explorer window titled "Sample Form - Microsoft Internet Explorer". The address bar displays "D:\HTML-DHTML-JS-Examples\Chapter41.html". The page content is a "Sample Stock Survey" form:

- Describe your investment experience**: Radio buttons for "beginner", "intermediate", and "expert" are shown, with "intermediate" selected.
- Types of Investments you make**: Checkboxes for "Individual Stocks", "Options", and "Mutual Funds" are shown, with "Individual Stocks" checked.
- What is your stock pick for this year?**: A large text input field is present.
- Action Buttons**: "Submit" and "Reset" buttons are at the bottom left.
- Status Bar**: "Done" and "My Computer" are visible at the bottom right.

Hình 2.49. Kết quả ví dụ 41

### I) Phần tử TextArea (vùng văn bản)

Loại này tạo ra một điều khiển nhập văn bản trên nhiều dòng so với hộp văn bản nhập một dòng. Ta phải xác định kích thước của textarea, xác định số dòng, số cột trong textarea và phải kết thúc phần tử với thẻ đóng </TEXTAREA>.

Tên	Mô tả
COLS	Truy xuất độ rộng của textarea
ROWS	Thiết lập hoặc truy xuất số hàng ngang trong <TEXTAREA>
SIZE	Thiết lập hoặc truy xuất kích thước của điều khiển
TYPE	Truy xuất loại điều khiển, sử dụng <TEXTAREA>
VALUE	Thiết lập hoặc truy xuất giá trị của <TEXTAREA>

Ví dụ :

```
<TEXTAREA NAME="text1" COLS=20 ROWS=5> </TEXTAREA>
```

### m) Phần tử BUTTON (nút bấm)

Điều khiển này tạo ra điều khiển button. Khi người dùng nhập vào nút Submit, biểu mẫu được nhận để xử lý. Cặp tên/giá trị của nút submit được nhận cùng với biểu mẫu.

Thuộc tính	Mô tả
NAME	Gán tên cho nút
VALUE	Gán giá trị cho nút
TYPE	Xác định loại nút. Các giá trị có thể là: <b>Submit</b> – Tạo nút nhận biểu mẫu khi được nhập vào <b>Button</b> – Tạo nút kích hoạt một script khi được kích vào <b>Reset</b> – Tạo nút thiết lập lại (reset) biểu mẫu và các giá trị của các điều khiển trong biểu mẫu

Một nút (BUTTON) có loại là submit (type=submit) giống như một phần tử INPUT của loại nút. Sự khác nhau là ở chỗ khi phần tử BUTTON được nhập vào thì cặp tên/giá trị được nhận cùng biểu mẫu. Một nút (BUTTON) có loại là submit cũng chứa một ảnh và giống một phần tử INPUT có loại là ảnh. Sự khác nhau là ở chỗ phần tử INPUT có dạng một ảnh “phẳng” trong khi phần tử BUTTON thì hiển thị như một nút có hiệu ứng lén / xuông khi nhập vào.

Ví dụ sau minh họa cho việc sử dụng phần tử TEXTAREA và BUTTON :

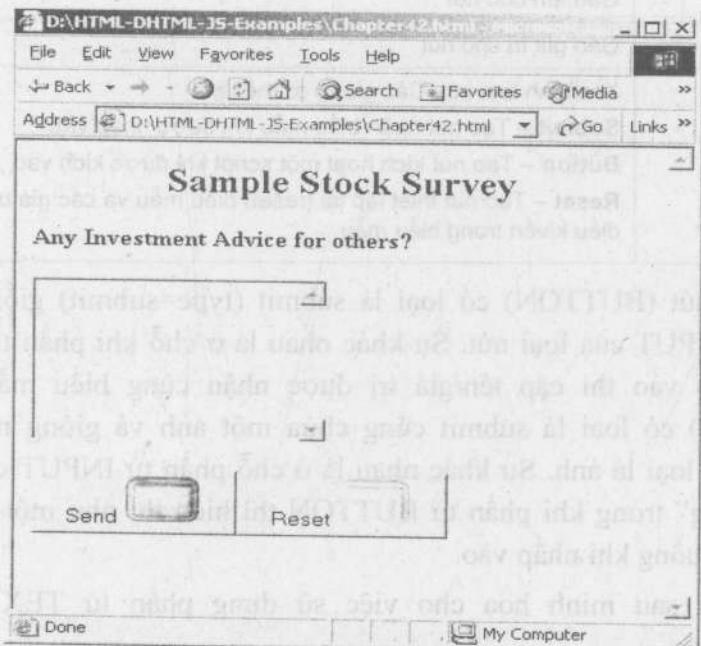
#### Ví dụ 42

```
<HTML>
<HEAD>
<TITLE> Sample Form </TITLE>
```

```

</HEAD>
<BODY bgColor="#ffffcc" Text="#000099">
<FORM ACTION="http://www.mysite.com/FormSite"
METHOD="POST">
<B><CENTER><H2>Sample Stock Survey</H2></CENTER></B>
<P><B>Any Investment Advice for others?</B></P>
<TEXTAREA NAME="RESULT_TextArea-8" ROWS="7" COLS="20">
</TEXTAREA>
<P>
<BUTTON type="submit" name="submit" value="submit">
Send <IMG src="submit.gif" alt="Submit"> </BUTTON>
<BUTTON type="reset" name="reset" >
Reset <IMG src="reset.gif" alt="Reset"> </BUTTON>
</P>
</FORM>
</BODY>
</HTML>

```



Hình 2.50. Kết quả ví dụ 42

### n) Phần tử lựa chọn (Select)

Phần tử SELECT được sử dụng để hiển thị một danh sách các lựa chọn cho người dùng. Mỗi lựa chọn được biểu diễn bởi phần tử OPTION.

Một phần tử SELECT phải chứa ít nhất một phần tử OPTION. Thành phần được chọn lựa sẽ hiển thị với màu khác so với các thành phần còn lại.

Thuộc tính	Mô tả
NAME	Gán tên cho phần tử. Khi biểu mẫu được gửi đi, thuộc tính tên được gán với giá trị lựa chọn.
SIZE	Nếu có nhiều sự lựa chọn, người dùng sử dụng cuộn, thuộc tính này xác định số dòng trong danh sách được hiển thị.
MULTIPLE	Là thuộc tính logic cho phép người dùng chọn một hoặc nhiều lựa chọn.

Mỗi lựa chọn trong hộp chọn được lấy giá trị thông qua mô tả văn bản của nó xem có được chọn hay không.

Mảng Option được tạo ra theo danh sách lựa chọn trong phần tử SELECT. Mỗi lựa chọn có thuộc tính Text và Selected cho phép chúng ta kiểm tra tùy chọn đó có được chọn hay không và truy xuất văn bản của lựa chọn theo thứ tự. Bây giờ ta có thể kiểm tra mỗi phần tử trong mảng và xác nhận nó.

### Ví dụ 43

```
<HTML>
<HEAD>
    <TITLE> Sample Form </TITLE>
</HEAD>
<BODY bgColor="#ffffcc" Text="#000099">
    <FORM ACTION="http://www.mysite.com/FormSite"
METHOD="POST">
        <B><CENTER><H2>Sample Stock Survey</H2></CENTER></B>
        <P><B>How do you buy your stocks?</B>
        <SELECT NAME="RESULT_RadioButton-5">
            <OPTION></OPTION>
            <OPTION>1) On-Line</OPTION>
            <OPTION>2) Touch Tone Trading</OPTION>
            <OPTION>3) Broker Assisted</OPTION>
            <OPTION>4) Other</OPTION>
        </SELECT></P>
        <BR><BR><BR>
        <P><INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Submit">
        <INPUT TYPE="RESET" NAME="Reset" VALUE="Reset"></P>
    </FORM>
</BODY>
</HTML>
```

**Sample Form - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address D:\HTML-DHTML-JS-Examples\Chapter43.html Go Links

**Sample Stock Survey**

How do you buy your stocks?

Submit Reset

1) On-Line  
2) Touch Tone Trading  
3) Broker Assisted  
4) Other

Done My Computer

Hình 2.51. Kết quả ví dụ 43

Phân tử OPTGROUP được sử dụng để nhóm các lựa chọn vào một cây phân cấp. Ví dụ, bảng nội dung có thể có tên các chương. Các chủ đề và chủ đề con trong một chương có thể được nhóm vào chương đó.

Thuộc tính	Mô tả
SELECTED	Đây là thuộc tính logic sử dụng để chọn trước một tùy chọn.
VALUE	Xác định giá trị được nhận vào cho tùy chọn được chọn. Giá trị này được gán với tên của phân tử SELECT. Nội dung của phân tử OPTION là giá trị mặc định.
LABEL	Xác định văn bản hiển thị cho một tùy chọn.

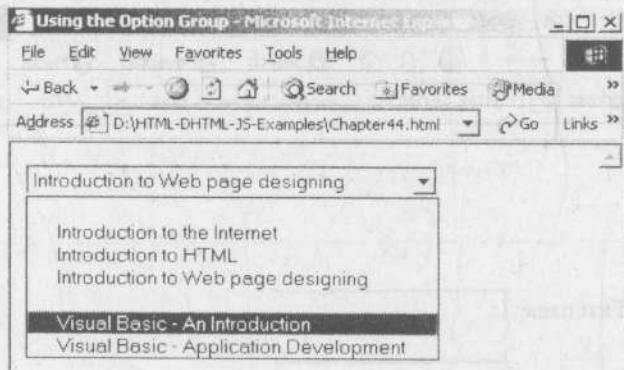
#### Ví dụ 44

```
<HTML>
<Head>
<Title>Using the Option Group</Title>
</head>
<BODY>
<FORM action = "http://www.mysite.com/FormSite"
      method = "post">
<P><SELECT name="course">
<OPTGROUP>
<OPTION value = "Internetintro">Introduction to the
      Internet
      <OPTION value="Introhtml">Introduction to HTML
```

```

        <OPTION value="Introweb">Introduction to Web
        page designing
    </OPTGROUP>
    <OPTGROUP>
        <OPTION value="vbintro">Visual Basic-An
        Introduction
        <OPTION value="vbdev">Visual Basic - Application
        Development
    </OPTGROUP>
</SELECT>
</FORM>
</BODY>
</HTML>

```



Hình 2.52. Kết quả ví dụ 44

### p) Phần tử LABEL (Nhãn)

Phần tử LABEL được sử dụng để gắn thông tin vào các phần tử điều khiển. Ví dụ, phần tử TEXT không có nhãn để xác định rõ nó. Ta có thể gán nhãn vào phần tử TEXT khi trang hiển thị và phải xác định thuộc tính ID của điều khiển mà nó được gán vào.

#### Ví dụ 45

```

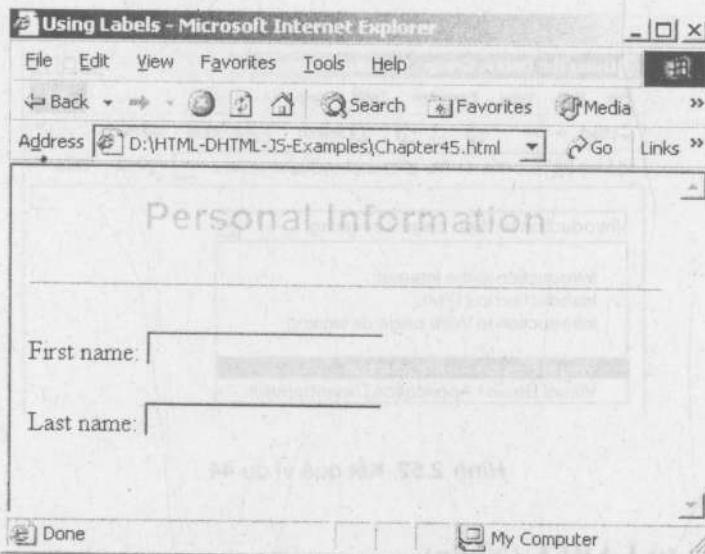
<HTML>
    <HEAD>
        <TITLE>Using Labels</TITLE>
    </HEAD>
    <BODY>
        <H2><CENTER><FONT size = 5 color = hotpink face =
        arial>Personal Information</FONT></CENTER></H2>

```

```

<HR align = center>
<FORM action="http://somesite.com" method="post">
<P>
    <LABEL for="firstname">First name: </LABEL>
    <INPUT type="text" id="firstname"><BR><BR>
    <LABEL for="lastname">Last name: </LABEL>
    <INPUT type="text" id="lastname"><BR><BR>
</P>
</FORM>
</BODY>
</HTML>

```



Hình 2.53. Kết quả ví dụ 45

#### 8.1.4. Tạo biểu mẫu

Chúng ta đã thảo luận về phần tử `<FORM>` và các điều khiển có thể thêm vào biểu mẫu để nhận thông tin người dùng. Trong phần này, chúng ta sẽ tạo một biểu mẫu để nhận thông tin về một người xin việc. Các nút `RESET` và `SUBMIT` thực hiện các công việc cần thiết.

#### Ví dụ 46

```

<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>
            <LABEL for="firstname">Name: </LABEL>
            <INPUT type="text" id="firstname"><BR>
        <P>Area of Interest
        <BR><BR>
        <INPUT TYPE=RADIO NAME="CONTROL1" VALUE="0"
CHECKED>Web Designer
        <INPUT TYPE=RADIO NAME="CONTROL1" VALUE="1" >Web
Administrator
        <INPUT TYPE=RADIO NAME="CONTROL1" VALUE="2" >Web
Developer
        <P>Experience
        <SELECT NAME="CONTROL2">
            <OPTION>None</OPTION>
            <OPTION>1 Year</OPTION>
            <OPTION>3 Years</OPTION>
            <OPTION>5 Years</OPTION>
        </SELECT>
        <BR>
        <P>Comments
        <BR>
        <TEXTAREA NAME="CONTROL3" COLS="30" ROWS="5">Type
your comments here.</TEXTAREA>
        <BR>
        <P><INPUT NAME="CONTROL4" TYPE=CHECKBOX CHECKED>Send
acknowledgement
        <BR>
        <P><INPUT TYPE=SUBMIT VALUE=OK>
        <INPUT TYPE=RESET VALUE=RESET>
    </FORM>
</BODY>
</HTML>

```

The screenshot shows a Microsoft Internet Explorer window with the title bar "Job application - Microsoft Internet Explorer". The menu bar includes File, Edit, View, Favorites, Tools, and Help. The toolbar includes Back, Forward, Stop, Home, Search, Favorites, Media, and Links. The address bar shows the path "D:\HTML-DHTML-JS-Examples\Chapter46.html". The main content area displays an "Application Form" with the following fields:

- Name: [Text input field]
- Area of Interest:
  - Web Designer
  - Web Administrator
  - Web Developer
- Experience: [Dropdown menu showing "None"]
- Comments:
 

Type your comments here.
- Send acknowledgement

At the bottom are two buttons: "OK" and "RESET".

Hình 2.54. Kết quả ví dụ 46

Khi có nhiều phần tử trong một form, ta cần phải điều khiển chúng. Sau đây là các thuộc tính để điều khiển các phần tử.

#### a) Thiết lập tiêu điểm (Focus)

Một phần tử trở thành hoạt động khi nó nhận tiêu điểm. Ví dụ, để nhập văn bản vào phần tử TEXT, tiêu điểm phải nằm trên phần tử đó. Khi phần tử mất tiêu điểm, nó sẽ không hoạt động nữa. Cách đơn giản nhất để đặt tiêu điểm cho phần tử là kích vào nó bằng cách sử dụng chuột, joystick... hoặc dùng bàn phím để đặt.

#### b) Thứ tự tab

Thuộc tính tabindex của một phần tử xác định trình tự phần tử nhận tiêu điểm thông qua bàn phím. Ở đây bao gồm các phần tử được lồng vào các phần tử khác. Giá trị có thể là bất cứ số nào giữa 0 và 32767. Tiêu điểm bắt đầu từ phần tử có giá trị tabindex thấp nhất. Nếu ta gán cùng một giá trị

tabindex cho hơn một phần tử, thì các phần tử nhận tiêu điểm theo thứ tự nó xuất hiện trong tài liệu.

Nếu phần tử nào không hỗ trợ thuộc tính tabindex, nó sẽ là phần tử nhận tiêu điểm cuối cùng. Nếu vô hiệu hóa một phần tử, nó sẽ không được liệt kê vào thứ tự tab – và sẽ không nhận được tiêu điểm.

Ví dụ :

```
<INPUT tabindex=2 TYPE=RADIO NAME="CONTROL1" VALUE="0"
CHECKED>Web Designer
<INPUT tabindex = 6 TYPE=SUBMIT VALUE=OK>
```

#### c) Phím truy cập (Access Keys)

Thuộc tính này được sử dụng để gán phím truy cập cho phần tử. Phím truy cập là một ký tự và thường được sử dụng cùng với phím ALT. Khi người dùng nhấn phím truy cập, phần tử được xác định sẽ nhận tiêu điểm và bắt đầu hoạt động.

Ví dụ :

```
<LABEL for="firstname">Name: </LABEL>
<INPUT accesskey="N" tabindex=1 type="text" id="firstname">
<TEXTAREA accesskey="C" tabindex=4 NAME="CONTROL3" COLS="30"
ROWS="5">Type your comments here</TEXTAREA>
```

#### d) Phần tử vô hiệu hoá (Disabling Elements)

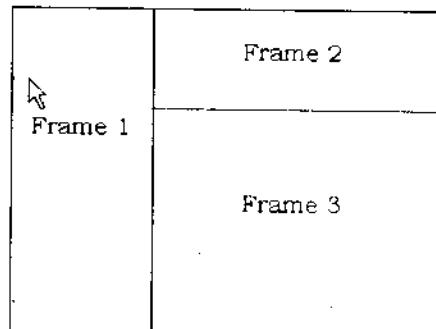
Nếu làm việc với trình soạn thảo văn bản, ta sẽ thấy rằng nếu không mở tài liệu nào thì các tùy chọn lưu và định dạng sẽ bị vô hiệu hoá. Đối với trang Web, có thể vô hiệu hoá các phần tử hoặc để ở trạng thái chỉ đọc (read-only) nếu không muốn người dùng truy cập chúng. Ví dụ, khi hiển thị một biểu mẫu, ta có thể vô hiệu hoá nút “Submit” cho đến khi người dùng nhập dữ liệu vào. Thuộc tính vô hiệu hoá được sử dụng để điều khiển truy cập một phần tử. Khi một phần tử bị vô hiệu hoá, nó không được liệt kê trong thứ tự tab. Do vậy, điều khiển không nhận được tiêu điểm và cuối cùng là các giá trị của điều khiển bị vô hiệu hoá không được chuyển đi cùng với biểu mẫu. Một điều khiển bị vô hiệu hoá có thể được kích hoạt nhờ các script lúc thực hiện.

Ví dụ :

```
<INPUT TYPE=SUBMIT VALUE=OK DISABLED=True>
```

## 8.2. Khung (Frame)

Khung chia một cửa sổ trình duyệt thành nhiều vùng riêng biệt, trong đó mỗi vùng có thể hiển thị một trang riêng biệt có thể cuộn được. Mỗi khung là một cửa sổ trong cửa sổ chính. Ví dụ, ta có thể sử dụng ba khung trong trang Web, một làm biểu ngữ (banner), một làm menu điều hướng và một để hiển thị dữ liệu. Mỗi khung có thể được tạo, thay đổi và cuộn độc lập nhau.



Hình 2.55. Khung trong trang Web

### 8.2.1. Tại sao cần sử dụng khung ?

Một trang có thể có một hoặc nhiều khung và lý do cần sử dụng khung là :

- Hiển thị một biểu tượng (logo) hoặc thông tin tĩnh tại một vị trí cố định trên trang Web.
- Đối với bảng nội dung trong trang, ở đó người dùng có thể kích vào và di chuyển quanh Website mà không cần phải liên tục quay lại trang nội dung.
- Nhiều cách hiển thị cho phép người thiết kế giữ một số thông tin tĩnh nào đó trong khi cuộn hay thao tác đối với những nội dung khác trên trang Web.

Tuy nhiên, mặt hạn chế của việc sử dụng khung trong trang Web là không phải tất cả các trình duyệt đều hỗ trợ khung, ví dụ, IE phiên bản 2.0 hoặc trước đó và Netscape 1.2 hoặc trước đó. Để nội dung vẫn hợp lệ mà người sử dụng không cần quan tâm đến trình duyệt có hỗ trợ khung hay không, người thiết kế cần phải cung cấp một cách khác để truy cập vào nội dung.

### 8.2.2. Sử dụng khung

Một tài liệu HTML chuẩn có phần HEAD và BODY. Một tài liệu HTML sử dụng khung thì có phần HEAD và phần FRAMESET. Phần FRAMESET xác định cách trình bày trong cửa sổ người dùng. Ta không thể sử dụng phần tử BODY và FRAMESET cùng với nhau. Trình duyệt chỉ nhận phần tử đầu tiên xuất hiện trong tài liệu và bỏ qua phần tử sau.

Nghĩa là, nếu bạn sử dụng phần tử BODY, thì phần tử FRAMESET sau đó sẽ bị bỏ qua và ngược lại.

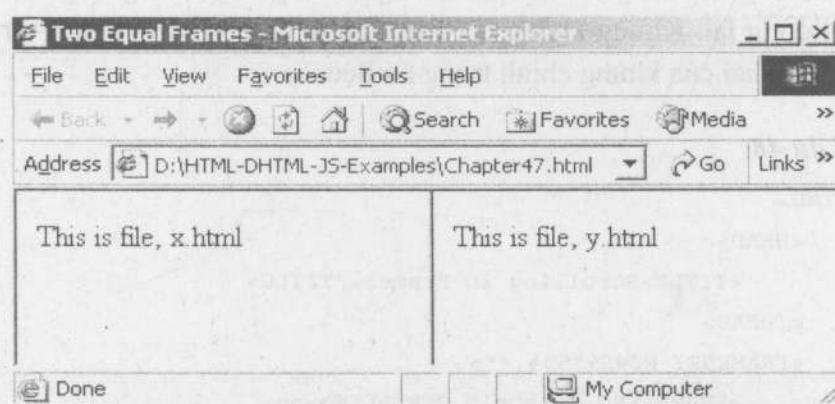
Khung được tạo ra bằng cách sử dụng phần tử FRAMESET. Các thuộc tính như sau :

Thuộc tính	Mô tả
ROWS	Xác định độ rộng của khung, được tính theo điểm(pixels), phần trăm hoặc độ rộng tương đối. Giá trị mặc định là 100%, nghĩa là một dòng
COLS	Xác định độ cao của khung, được tính theo điểm(pixels), phần trăm hoặc độ cao tương đối. Giá trị mặc định là 100%. Nó xác định chỉ có một cột

Phần tử FRAME xác định hình thức và nội dung của một khung trong FRAMESET. Ví dụ sau đây tạo hai khung bằng nhau, chia đôi cửa sổ.

#### Ví dụ 47

```
<HTML>
  <HEAD>
    <TITLE>Two Equal Frames</TITLE>
  </HEAD>
  <FRAMESET COLS="50%, *" >
    <FRAME SRC=x.htm>
    <FRAME SRC=y.htm>
  </FRAMESET>
</HTML>
```



Hình 2.56. Kết quả ví dụ 47

**Chú ý :** File x.html và y.html được lưu cùng thư mục với file. html chính.

Các thuộc tính của FRAME bao gồm:

Thuộc tính	Mô tả
NAME	Thuộc tính này gán tên cho khung hiện thời
SRC	Thuộc tính này xác định vị trí tài liệu ban đầu được chứa trong khung
NORESIZE	Đây là thuộc tính logic. Nó quy định cửa sổ khung không được thay đổi kích thước.
SCROLLING	Thuộc tính này xác định kiểu cuộn cho cửa sổ khung. Các giá trị có thể là: Auto - Xuất hiện thanh cuộn khi cần thiết. Đây là giá trị mặc định Yes - Luôn luôn xuất hiện thanh cuộn trong cửa sổ của khung No - Không xuất hiện thanh cuộn trong cửa sổ của khung
FRAMEBORDER	Xác định viền của khung. Các giá trị có thể là: 1 - Là giá trị mặc định. Có sự phân cách giữa khung hiện thời với các khung xung quanh. 0 - Không có sự phân cách giữa khung hiện thời với các khung lân cận. Tuy nhiên, nếu các khung lân cận có thiết lập thì vẫn xuất hiện sự phân cách này.
MARGINWIDTH	Xác định khoảng cách giữa nội dung trong khung với lề trái và lề phải của khung. Giá trị phải lớn hơn một.
MARGINHEIGHT	Xác định khoảng cách giữa nội dung trong khung với lề trên và lề dưới của khung. Giá trị phải lớn hơn một.

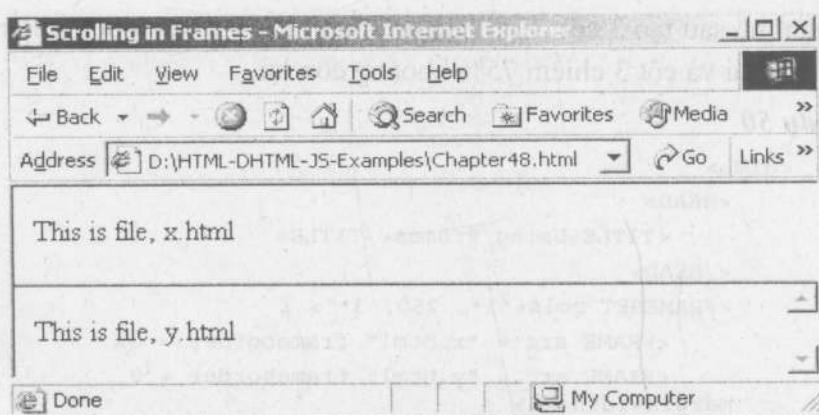
Ta không thể đóng cửa sổ của khung. Khung được đóng khi cửa sổ tạo ra nó bị đóng lại. Khung không có thanh trạng thái, vì vậy ta phải sử dụng thanh trạng thái của khung chính trong tài liệu.

#### Ví dụ 48

```

<HTML>
  <HEAD>
    <TITLE>Scrolling in Frames</TITLE>
  </HEAD>
  <FRAMESET ROWS="50%, *" >
    <FRAME SRC="x.html" SCROLLING=no>
    <FRAME SRC="y.html" SCROLLING=yes>
  </FRAMESET>
</HTML>

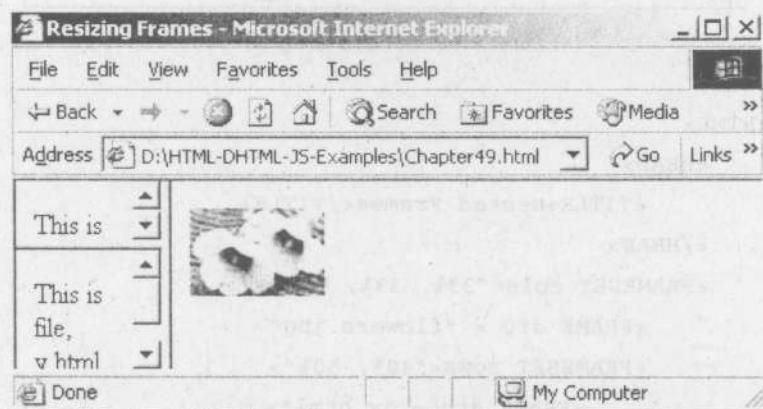
```



Hình 2.57. Kết quả ví dụ 48

### Ví dụ 49

```
<HTML>
  <HEAD>
    <TITLE>Resizing Frames</TITLE>
  </HEAD>
  <FRAMESET cols="20%, 80%">
    <FRAMESET rows="100, 200">
      <FRAME src="x.html" noresize>
      <FRAME src="y.html">
    </FRAMESET>
    <FRAME src="flowers.jpg">
  </FRAMESET>
</HTML>
```



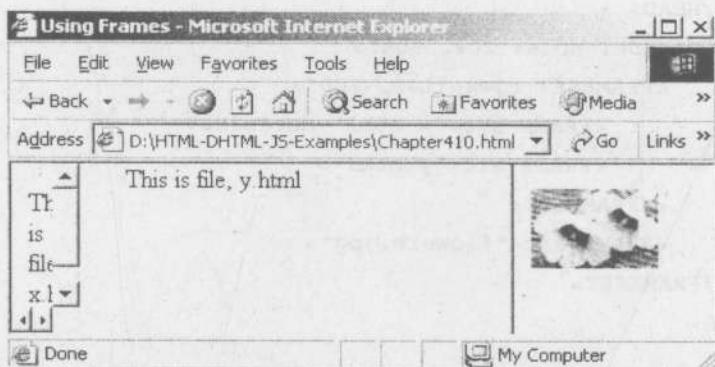
Hình 2.58. Kết quả ví dụ 49

Đoạn mã sau tạo 3 cột: Cột 2 có độ rộng là 250 pixel, cột 1 chiếm 25% khoảng còn lại và cột 3 chiếm 75% khoảng còn lại.

### Ví dụ 50

```
<HTML>
  <HEAD>
    <TITLE>Using Frames</TITLE>
  </HEAD>
  <FRAMESET cols="1*, 250, 3*">
    <FRAME src = "x.html" frameborder = 0>
    <FRAME src = "y.html" frameborder = 0
marginwidth=25>
    <FRAME src = "flowers.jpg">
  </FRAMESET>
</HTML>
```

Trong ví dụ 49 đã sử dụng phần tử FRAMESET lồng nhau (Nested framesets). Ta có thể tạo ra các frameset (tập khung) lồng nhau ở bất kỳ mức nào như ví dụ minh họa dưới đây:



Hình 2.59. Kết quả ví dụ 50

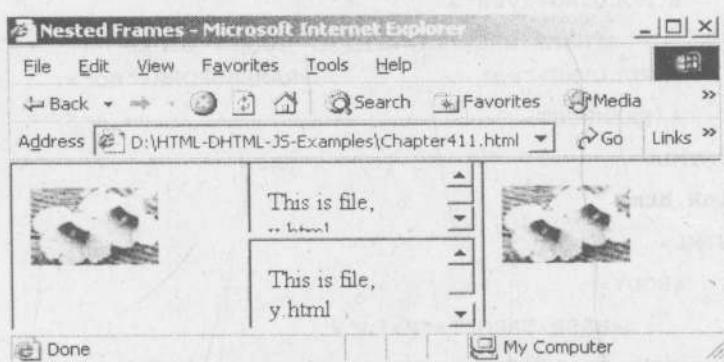
### Ví dụ 51

```
<HTML>
  <HEAD>
    <TITLE>Nested Frames</TITLE>
  </HEAD>
  <FRAMESET cols="33%, 33%, 34%">
    <FRAME src = "flowers.jpg">
    <FRAMESET rows="40%, 50%">
      <FRAME src = "x.html">
      <FRAME src = "y.html">
```

```

</FRAMESET>
<FRAME src = "flowers.jpg">
</FRAMESET>
</HTML>

```



Hình 2.60. Kết quả ví dụ 51

### Liên kết các khung

Khi tạo liên kết trong trang Web, ta có thể thiết lập một khung như một mục tiêu của liên kết (link). Trình duyệt tuân theo các bước sau:

- Nếu xác định một khung trong thuộc tính đích (TARGET) của phần tử, thì tài liệu mà phần tử chỉ ra sẽ được tải vào khung đó khi phần tử được kích hoạt.
- Nếu thuộc tính TARGET không được thiết lập thì thuộc tính TARGET của phần tử BASE sẽ được sử dụng để xác định khung.
- Nếu cả phần tử và phần tử BASE không đề cập đến TARGET, thì tài liệu được tải vào khung chứa phần tử đó.
- Nếu không tìm thấy khung thì trình duyệt tạo một cửa sổ và khung mới, sau đó tải tài liệu vào khung mới này.

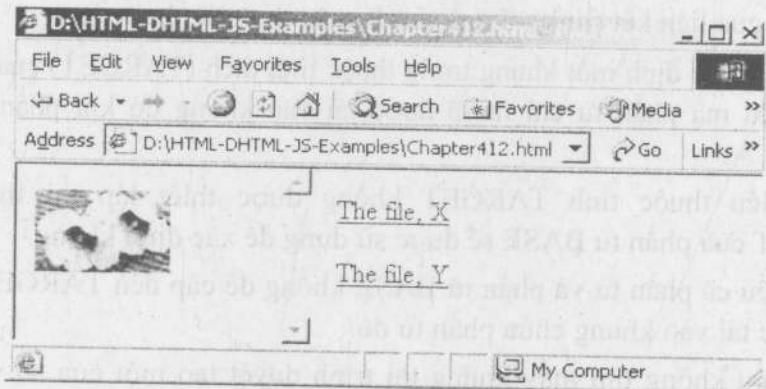
Thuộc tính TARGET được sử dụng để xác định tên khung mà tài liệu được mở trong đó. Khi tạo khung, ta cần phải đặt thuộc tính tên. Tên này được dùng khi tạo liên kết. Sau khi thay đổi nội dung của một khung thì định nghĩa FRAMESET ban đầu bị mất đi. Nếu có nhiều liên kết đến cùng một đích, ta có thể thiết lập một TARGET mặc định trong phần tử BASE. Sau đó, việc xác định thuộc tính TARGET trong mỗi phần tử sẽ không cần thiết nữa.

## Ví dụ 52

```
<HTML>
  <FRAMESET COLS="40%, 60%">
    <FRAME SRC="FLOWERS.JPG" NAME="Flowers"
           SCROLLING="yes">
    <FRAME SRC="LINK.HTML" NAME="Links"
           SCROLLING="no" FRAMEBORDER="no" >
  </FRAMESET>
</HTML>

Link.html

<HTML>
  <BODY>
    <BASE TARGET="Main">
    <P><A HREF="X.HTML">The file, X</A><P>
    <P><A HREF="Y.HTML">The file, Y</A><P>
  </BODY>
</HTML>
```



Hình 2.61. Kết quả ví dụ 52

## Phần tử NOFRAMES

Nếu trình duyệt không hỗ trợ khung, với tư cách là người phát triển ứng dụng ta nên cung cấp một cách khác để hiển thị nội dung. Phần tử NOFRAMES được sử dụng để làm việc đó. Nó chỉ hoạt động trong trình duyệt không hỗ trợ khung.

### *Ví dụ 53*

```

<HTML>
    <FRAMESET COLS="40%, 60%">
        <FRAME SRC="Flowers.jpg" NAME="Flowers"
SCROLLING=yes>
        <FRAMESET ROWS="60, *">
            <FRAME SRC="x.html" NAME="x" SCROLLING=no
FRAMEBORDER=no>
            <FRAME SRC="y.html" NAME="y">
            <NOFRAMES>
                Frames are not being displayed. Click
                here <A href="main.htm">for a nonframes
                version</A>
            </NOFRAMES>
        </FRAMESET>
    </FRAMESET>
</HTML>

```

#### *8.2.3. Phần tử IFRAME – khung trên dòng (inline frame)*

Phần tử IFRAME được sử dụng để tạo khung trên dòng (inline frame) hay khung nổi (floating frame). Ta có thể tạo và chèn một khung vào một khối văn bản. Khi trình duyệt không hỗ trợ khung thì nội dung nằm trong thẻ IFRAME bị trả lại. Khung trên dòng (inline frame) không thể thay đổi kích thước.

Thuộc tính	Mô tả
NAME	Gán tên cho khung hiện thời
WIDTH	Xác định độ rộng của khung trên dòng
HEIGHT	Xác định chiều cao của khung trên dòng

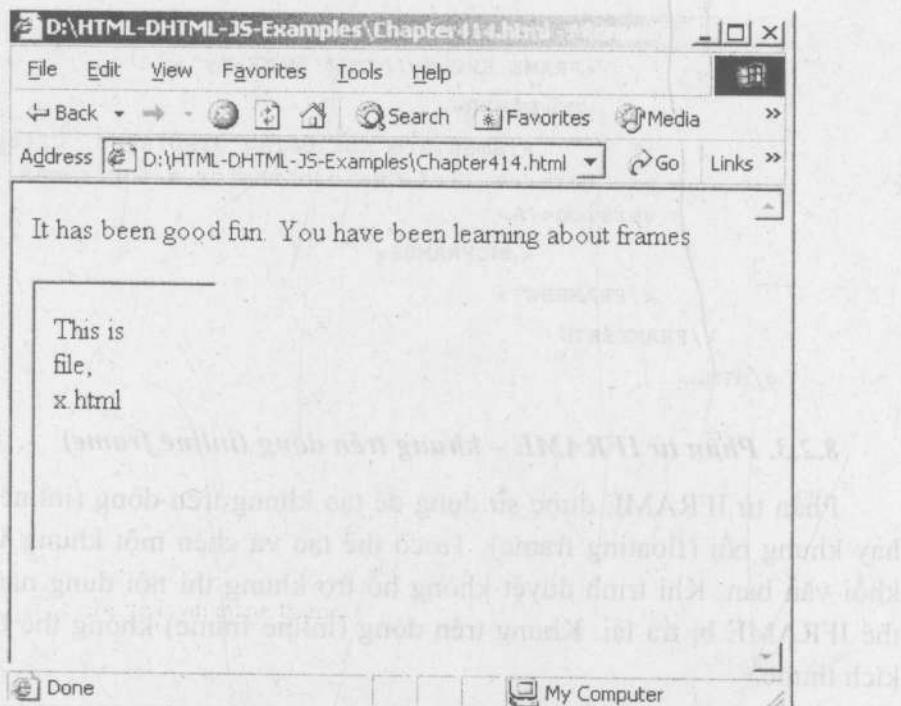
### *Ví dụ 54*

```

<HTML>
    <P>It has been good fun. You have been learning about
    frames
    <BR><BR>
    <IFRAME src="x.html" width="100" height="150"
scrolling=auto frameborder=1>

```

```
[The browser does not support frames or is currently configured  
not to display frames. These are the <A href="x.html">contents</A>]  
</IFRAME>  
<BR>  
<P><FONT color = hotpink>The above is an inline frame  
</HTML>
```



Hình 2.62. Kết quả ví dụ 54 (trình duyệt có hỗ trợ khung)

# *Chương 3*

## **CÁC STYLE SHEET**

Trước đây, khi một trang web được hiển thị trong một trình duyệt, người ta không thể thay đổi bất cứ nội dung gì trên đó, cả người sử dụng và tác giả của trang web đều không thể điều khiển các phần tử của HTML trên trang web. Với những phiên bản mới hơn của các trình duyệt đã hỗ trợ thêm đặc tính gọi là HTML động. Phần này sẽ trình bày về đặc tính HTML động và một số những điểm mới mà đặc tính này đem lại cho các nhà thiết kế web.

Chương 3 cũng sẽ trình bày thêm về đặc tính style sheet. Style sheet là một đặc tính quan trọng được sử dụng trong HTML động. Mặc dù trang Web không cần có một style sheet, nhưng việc sử dụng một style sheet sẽ mang lại những lợi ích nhất định. Chúng ta sẽ tìm hiểu về style sheet như là một kỹ thuật và bằng cách nào để có thể sử dụng nó trong việc thiết kế và phát triển Web.

### **1. DHTML**

“HTML động” được định nghĩa như là một phần mềm sử dụng cho việc mô tả sự kết hợp giữa HTML, các style sheet và ngôn ngữ script làm cho tài liệu trở nên sinh động.

Vào thời gian đầu, HTML được phát triển như một định dạng tài liệu được dùng để trao đổi thông tin trên Internet. Việc truyền tải dữ liệu độc lập với nền tảng (platform). Tuy nhiên, trọng tâm đã chuyển từ các vấn đề khoa học và hàn lâm thành các vấn đề của người dùng hàng ngày, hiện nay, mọi người hiện nay đang xem Internet như là nguồn thông tin và giải trí. Các trang web trở nên hấp dẫn và nhiều màu sắc hơn, làm thu hút người dùng, nhưng nhìn chung các dáng vẻ cơ bản và nội dung của những trang web vẫn cố định. Người phát triển can thiệp rất ít hoặc không có sự điều khiển nào đối với một trang web khi nó được hiển thị. HTML vẫn còn “tĩnh”.

## 1.1. Giới thiệu DHTML

Sự ra đời của lập trình script đã thêm vào phần động cho các trang web. Người dùng có thể tương tác với trang web. Tuy nhiên, một số hạn chế vẫn còn tồn tại. Bất kỳ sự xác nhận dữ liệu nào hoặc việc lập trình script đều phải được thực hiện trên máy chủ. Để thay đổi nội dung hoặc một kiểu của trang thì trang đó phải được viết đè lên hoàn toàn. Bất cứ tương tác nào của người dùng cũng đều thông qua máy Web server.

Với mỗi phiên bản trình duyệt mới ra đời, lại bổ sung thêm các đặc tính tốt hơn cho HTML. Ngày nay Internet và Netscape Navigator hỗ trợ một mô hình đối tượng tài liệu “*Document Object Model*” mà ở đó các phần tử HTML và các thẻ được coi như một đối tượng. Những đối tượng có những phương thức, thuộc tính và sự kiện có thể lập trình để điều khiển các hoạt động của chúng. Ví dụ, cú pháp thêm vào để thay đổi màu của văn bản trong phần tử phân đoạn <p> khi người dùng kích chuột lên nó.

Các script (là một chương trình nhỏ) có thể thực thi trong trình duyệt. Điều này có nghĩa là dữ liệu có thể được thao tác, định dạng và thay đổi một cách năng động ở máy khách (client) mà không cần quá nhiều sự hỗ trợ từ máy chủ. Tương tác của người dùng giờ đây có thể được xử lý bởi chính máy khách.

**Chú ý :** Một ứng dụng Client/Server được tách ra hai phần, một là giao diện người dùng “front-end client” và phần “back-end server”. Client là một phần của ứng dụng, nó trình bày dữ liệu đối với người dùng. Thông thường, Client “giao diện người dùng” không thực thi các chức năng của cơ sở dữ liệu, thay vào đó, client gửi các yêu cầu dữ liệu đến một máy chủ “server”, định dạng và hiển thị kết quả. Vai trò của máy chủ “server” cung cấp xử lý hoặc thông tin đến cho client. Máy chủ cung cấp dữ liệu đến client, nhưng đôi khi máy chủ “server” có thể cần thực hiện một số công việc xử lý đem lại một kết quả dữ liệu yêu cầu. Ví dụ, nếu một client yêu cầu về dữ liệu bán hàng cho một vùng cụ thể, server cần thực hiện chính xác một số xử lý dữ liệu từ tập tất cả các dữ liệu và định dạng nó theo yêu cầu từ phía client.

Mỗi công ty Microsoft hay Netscape đều có cách triển khai HTML động riêng của họ. Microsoft tập trung vào dùng các Cascading Style Sheet (CSS). Chúng ta có thể dùng script để tương tác những phần tử CSS.

Netscape, ngược lại, dựa vào các phương thức dùng các tầng. The LAYER được dùng để cung cấp hầu hết các thuộc tính của HTML động.

## 1.2. Các đặc điểm của DHTML

DHTML không dừng lại ở một số phần mở rộng của HTML. Trong phần này sẽ trình bày về các đặc điểm của DHTML và cách thức dùng nó để thêm vào những tính năng động cho trang web.

– **Kiểu động (Dynamic Style)** : Trong các phiên bản của HTML trước đây, khi muốn thay đổi kiểu hay nội dung của một trang web khi đã hiển thị trong trình duyệt thì toàn bộ trang đó phải được nạp lại "refresh". Điều đó có nghĩa là yêu cầu phải được gửi đến máy chủ thành một trang mới để hiển thị. Đối với người dùng thì đây là một quy trình rõ ràng. Tuy nhiên, nếu trang đó được nạp lại 'refresh' thường xuyên sẽ tốn nhiều thời gian và làm cho máy chủ trở nên quá tải.

Trong HTML động, cách làm này khác một chút. Bằng cách dùng các bảng kiểu "style sheets", có thể xác định màu, kiểu hoặc kích cỡ của nội dung trang. Chúng ta có thể thay đổi kiểu của trang mà không cần gửi đến máy chủ 'Web server' cho mỗi lần thêm vào các thẻ và thuộc tính. Điều đó có nghĩa là chúng ta có thể thay đổi màu, font, kích cỡ hoặc nội dung văn bản khi đáp lại những tương tác của người dùng. Trong HTML động, kiểu liên quan đến cách thức, định dạng xuất hiện trên trang web nhiều hơn là nội dung. Kiểu "style" bao gồm màu sắc, kiểu chữ, khoảng cách, thụt đầu dòng, định vị và xuất hiện của văn bản.

– **Nội dung động (Dynamic Content)** : Được hỗ trợ bởi Internet Explorer. Ở đây chúng ta có thể thay đổi chữ và hình ảnh trên trang web sau khi nó hiển thị. Mặt khác, cũng có thể thay đổi nội dung của trang đó khi đáp lại dữ kiện nhập vào hay sự kiện người dùng kích chuột vào.

– **Định vị (Positioning)** : Các phiên bản của HTML trước đây không kiểm soát được vị trí của một phần tử trên trang web. Theo đó, vị trí chính xác của trang web về mặt toạ độ thì không thể chỉ ra được. Việc định vị dành cho trình duyệt. HTML chỉ có thể mô tả nội dung và vị trí tương đối của các phần tử.

Trong HTML động, chúng ta có thể chỉ ra vị trí chính xác (tuyệt đối hay tương đối), mối quan hệ giữa toạ độ x và y. Một khi trang web được

hiển thị, chúng ta có thể di chuyển các phần tử trên trang đó làm cho nó trở nên động.

+ Định vị tuyệt đối – chỉ rõ vị trí chính xác theo các điểm ‘pixels’.

+ Định vị tương đối – chỉ ra vị trí tương đối của các phần tử. Trình duyệt xử lý việc định vị hiện thời.

Đặc điểm việc định vị cũng cho phép chúng ta xác định thứ tự sắp xếp ‘z-order’ của các phần tử, có nghĩa là các đối tượng này nằm chồng lên đối tượng khác.

– **Liên kết dữ liệu (Data Binding)** : Trong HTML động, chúng ta có thể kết nối một cơ sở dữ liệu vào bảng của trang web. Nó được hỗ trợ bởi Internet Explorer. Khi trang được nạp lên, dữ liệu từ cơ sở dữ liệu trên máy chủ được hiển thị trong bảng. Dữ liệu có thể được sắp xếp, lọc và hiển thị cho phù hợp với yêu cầu.

– **Downloadable Fonts (khả năng tải Font chữ)** : Được Netscape hỗ trợ. Downloadable fonts có một đặc điểm là cho phép chèn các font tùy chọn trên trang web. Chúng ta có thể gói font trong trang. Điều này đảm bảo rằng, văn bản trong trang web đó luôn luôn hiển thị theo font đã chọn. Đây là đặc điểm quan trọng bởi vì thông thường nếu các font được chỉ ra không có trong máy của người dùng thì trình duyệt sẽ dùng font mặc định có sẵn. Điều này sẽ làm hủy bỏ mục đích chỉ ra kiểu font của người dùng.

– **Scripting** : Chúng ta có thể viết các script để thay đổi kiểu và nội dung của trang web. Script này được lồng vào trong trang web.

– **Cấu trúc đối tượng (Object Structure)** : HTML động có một cấu trúc đối tượng theo đó mỗi phần tử được đối xử như một đối tượng trong cấu trúc. Mỗi đối tượng có thể được truy cập và lập trình độc lập.

## 2. STYLE SHEETS

Style sheet được tạo nên từ các quy tắc kiểu cách mà nó báo cho trình duyệt biết được cách trình bày một tài liệu. Style sheets là một kỹ thuật thêm vào các kiểu (font, màu, khoảng cách) cho các trang web.

### 2.1. Khái niệm, chức năng và các thuận lợi của style sheet

Một tính năng quan trọng của HTML động là những kiểu (style) động. Nghĩa là bạn có thể thay đổi kiểu của những phần tử HTML trên trang sau

khi chúng được hiển thị trên trang đó. Sự thay đổi này có thể đáp ứng đối với sự tương tác người dùng hoặc thậm chí đối với sự thay đổi tình trạng như sự kiện thay đổi kích thước.

Có hai cách để thay đổi kiểu trên trang web.

- Thay đổi kiểu nội tuyến (inline style).
- Viết kịch bản để thay đổi kiểu.

Khi sử dụng kiểu nội tuyến, chúng ta có thể tạo ra các kiểu động mà không thêm bất cứ một kịch bản nào vào trang web.

Một kiểu nội tuyến là kiểu được gán trực tiếp cho một phần tử nào đó. Kiểu này không áp dụng vào tất cả các phần tử thuộc một loại nào hay một lớp nào đó. Kiểu nội tuyến được định nghĩa bằng thuộc tính **STYLE** đối với phần tử của thẻ đó. Ví dụ, nếu muốn đặt màu cho phần tử **<H1>** màu đỏ, chúng ta phải đặt thuộc tính **STYLE** bên trong thẻ H1 như sau:

```
<H1 style="color:red">
```

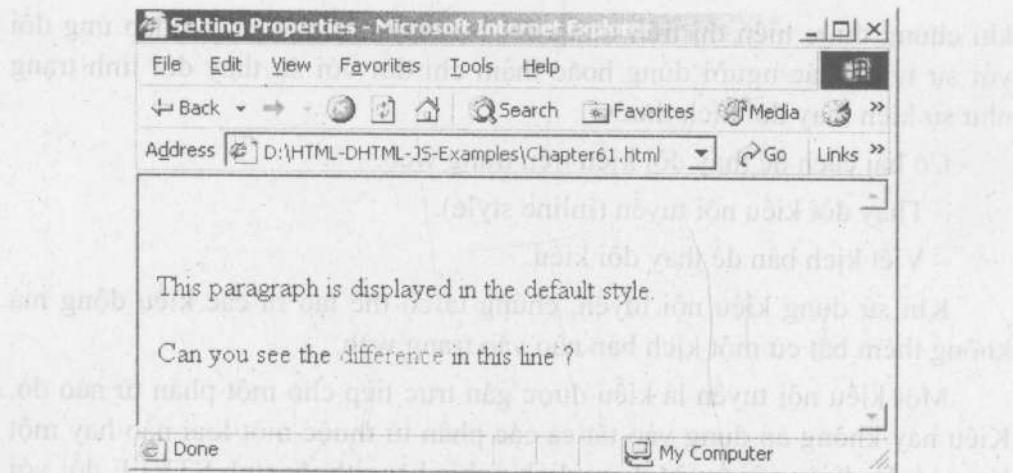
Khi muốn sử dụng kịch bản để thay đổi kiểu nội tuyến vào bất cứ lúc nào, thì khi đó phải sử dụng đến đối tượng kiểu (**Style Object**). Đối tượng kiểu hỗ trợ mọi tính chất mà CSS hỗ trợ đối với các kiểu. Để dùng thuộc tính trong kịch bản cần phải :

- Bỏ dấu gạch nối ra khỏi tên của kiểu CSS.
- Thay đổi chữ cái đầu tiên của từ sau dấu gạch nối thành từ viết hoa.

Ví dụ, **font-weight** của CSS trở thành **fontWeight** trong DHTML, hoặc **text-align** thành **textAlign**.

### *Ví dụ 1*

```
<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
        <BR>
        <P> This paragraph is displayed in the default style.
        <BR>
        <P>Can you see the <SPAN style = "color:red">difference
    </SPAN> in this line ?
    </BODY>
</HTML>
```



Hình 3.1. Kết quả ví dụ 1

\* **Lợi ích của các style sheet :** Các style sheet có thể được dùng để:

– **Nạp chòng trình duyệt :** Mỗi trình duyệt đều có thể hiển thị các trang web theo cách riêng của nó. Trước đây các nhà phát triển không kiểm soát được các trang web hiển thị trên trình duyệt, không biết trình duyệt nào mà người dùng cách nửa vòng trái đất sử dụng. Nhờ có các style sheet, ta có thể nạp chòng các quy ước của trình duyệt và đặt theo cách riêng. Ví dụ, bạn có thể xác định kiểu mà trong đó một phần tử <H1> cần hiển thị:

```
<H1><FONT SIZE=3 COLOR=AQUA>
<B>Overriding the browser</B></FONT></H1>
```

– **Bố cục trang (Page layout) :** Những style sheet có thể dùng để hiển thi font thay đổi màu mà không làm thay đổi cấu trúc của trang web. Điều này có nghĩa, với tư cách là một nhà thiết kế ta có thể tách biệt những yêu cầu về thiết kế hình ảnh trực quan từ cấu trúc logic của trang web và địa chỉ.

Khi sử dụng các biện pháp liên quan trong style sheet, ta có thể thể hiện các tài liệu sao cho đẹp mắt trên bất kỳ màn hình nào và theo bất kỳ độ phân giải nào.

– **Sử dụng lại các style sheet :** Khi đã định nghĩa kiểu thông tin, chúng ta có thể nhúng style sheet bên trong tài liệu HTML. Lần lượt thay thế, có thể kết nối tất cả các trang trên website đến style sheet. Điều này sẽ làm cho các trang web của chúng ta đều có cùng diện mạo thông tin khi được hiển thị. Vì vậy, ta có thể có được nền chung của trang ví dụ như logo của trang

và một số thông tin chuẩn (cho các trang) trong một style sheet. Điều này sẽ đảm bảo được cách nhìn và cảm nhận thông dụng về trang website.

– **Cần tạo stylesheet một lần thật tốt** : Chúng ta có thể tạo một style sheet và được liên kết đến nhiều tài liệu. Tất cả những tài liệu sẽ có diện mạo giống nhau. Tuy nhiên, khi thực hiện thay đổi style sheet thì tất cả các tài liệu được kết nối vào style sheet sẽ bị thay đổi theo.

## 2.2. Quy tắc style sheet

Style sheet phân cấp (cascading style sheet) định nghĩa các kiểu có thể được áp dụng vào các trang hoặc các phần tử của trang.

– **Quy tắc kiểu (Style Rule)** : Style sheet phân cấp là một tập hợp các quy tắc định nghĩa kiểu của tài liệu. Ví dụ, chúng ta có thể tạo ra một quy tắc kiểu được xác định cho tất cả phần tiêu đề <H1> hiển thị màu vàng xanh. Quy tắc kiểu có thể ứng dụng vào các thành phần được chọn của trang web. Ví dụ, chúng ta có thể xác định một đoạn văn bản bất kỳ in đậm và in nghiêng trên trang. Điều này được gọi là **khai báo kiểu có sẵn** mà nhờ đó các kiểu được áp dụng vào các phần tử HTML đơn lẻ trên một trang web.

– **Style Sheet** – Là một danh mục các quy tắc kiểu và có thể nhúng vào bên trong tài liệu HTML. Trong trường hợp đó, nó được gọi là Style sheet nhúng. Style sheet cũng có thể được tạo ra bằng một file bên ngoài và được liên kết với tài liệu HTML.

– **Các quy tắc (Rules)** – Bảng kiểu có thể có một hay nhiều quy tắc. Phần đầu của quy tắc gọi là bộ chọn (Selector). Mỗi bộ chọn có các thuộc tính và các giá trị liên quan đến nó.

```
RuleSelector {Declarations property: value; property: value; ... }
```

Phần của quy tắc được kèm theo trong phạm vi các dấu ngoặc móc được gọi là **khai báo**. Khai báo có hai phần, phần trước dấu hai chấm (:) là thuộc tính và phần nằm sau dấu hai chấm là giá trị của thuộc tính đó.

Các khai báo được phân cách bởi dấu chấm phẩy (:). Ta nên đặt dấu chấm phẩy sau lần khai báo cuối cùng.

Ví dụ :

```
h1 {color: blue}
```

Ở đây, H1 là bộ chọn, color: blue là khai báo.

Trong phạm vi khai báo:

{Property: Value}

Color là thuộc tính, blue là giá trị.

Mỗi quy tắc có thể tách rời nhau trong phạm vi thẻ STYLE.

### Ví dụ 2

```
<HTML>
```

```
    <HEAD>
```

```
        <STYLE TYPE="text/css">
```

```
            H1 {color:limegreen}
```

```
            H1 {font-family:Arial}
```

```
            H2 {color:limegreen}
```

```
            H2 {font-family:Arial}
```

```
        </STYLE>
```

```
    </HEAD>
```

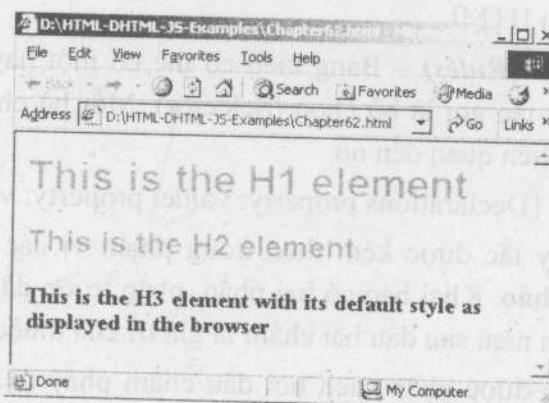
```
    <H1>This is the H1 element</H1>
```

```
    <H2>This is the H2 element</H2>
```

```
    <H3>This is the H3 element with its default style as  
displayed in the browser</H3>
```

```
</HTML>
```

**Chú ý:** text/css chỉ ra kiểu "style" được dùng ở đây là style sheet phân cấp "cascading style sheet"

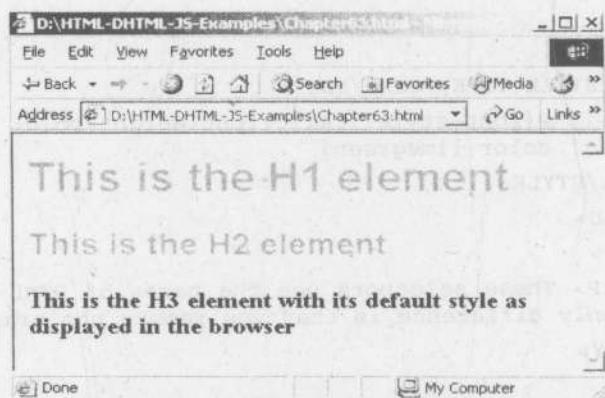


Hình 3.2. Kết quả ví dụ 2

Chúng ta cũng có thể nhóm các quy tắc. Mỗi khai báo được tách ra bởi dấu chấm phẩy.

### Ví dụ 3

```
<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      H1, H2{color:limegreen;font-family:Arial;}
    </STYLE>
  </HEAD>
  <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>
</HTML>
```



Hình 3.3. Kết quả ví dụ 3

### 2.3. Các Selector (bộ chọn) trong style sheet (hay các giả lớp)

Ta có thể dùng các giả lớp trong các selector nhưng không thể dùng chúng trong HTML. Về cơ bản, chúng dùng những thông tin bên ngoài để tác động đến việc định dạng. Ví dụ, với việc sử dụng lớp giả, các liên kết đã được ghé qua có thể hiển thị khác đi so với các liên kết chưa được ghé qua như sau:

```
A:link { color: red } /* unvisited link */
A:visited { color: blue } /* visited links */
A:active { color: lime } /* active links */
```

Selector có thể định nghĩa như là “một chuỗi ký tự xác định ra các phần tử và các quy tắc tương ứng áp dụng cho các phần tử đó”.

Có hai kiểu selector cơ bản:

– selector đơn: chỉ áp dụng cho một phần tử duy nhất.

### **- Selector đơn**

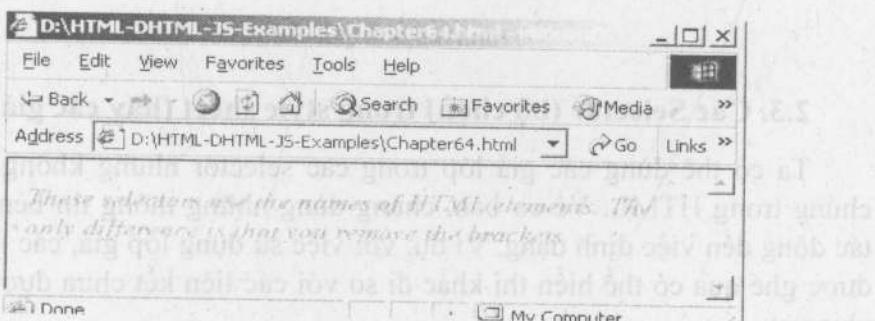
Dây là những selector dễ sử dụng nhất. Selector đơn mô tả một phần tử dù vị trí của nó ở đâu trong cấu trúc tài liệu. Bộ nhận dạng tiêu đề H1 là một điển hình. Sau đây là một số kiểu của selector đơn.

### **- Selector HTML**

Những selector này sử dụng tên của phần tử HTML và bỏ đi dấu mốc. Vì vậy, thẻ <P> trong HTML trở thành P và khi đó, nó được xem như là một selector. Ví dụ sau đây minh họa điều đó

#### **Ví dụ 4**

```
<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      P{font-style:italic;font-weight:bold;
      color:limegreen}
    </STYLE>
  </HEAD>
  <Body>
    <P> These selectors use the names of HTML elements. The
        only difference is that you remove the brackets. </P>
  </BODY>
</HTML>
```



**Hình 3.4.** Kết quả ví dụ 4

### **- Selector lớp**

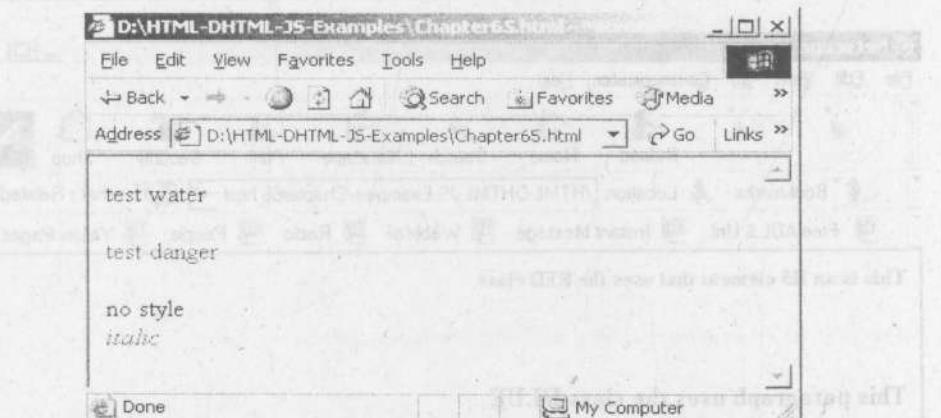
Những selector này sử dụng thuộc tính CLASS của các phần tử HTML. Mọi phần tử khi hiển thị có một thuộc tính CLASS được sử dụng để gán vào một định danh (identifier). Ta có thể gán một tên lớp duy nhất cho các phần tử khác. Ngoài ra, cũng có thể gán định danh lớp cho nhiều phần tử cùng loại khi muốn hiển thị các trạng thái khác nhau so với dạng chuẩn. Ví dụ, có thể

cho <H2> xuất hiện với nhiều màu khác nhau. Trong trường hợp đó, ta sử dụng định danh lớp cho <H2>.

Selector lớp có dấu chấm(.) đứng trước gọi là ký tự cờ (flag), theo sau là tên lớp do chúng ta chọn. Tốt nhất nên chọn những tên lớp theo mục đích của chúng, không nên chọn tên mô tả màu sắc hay kiểu của chúng. Ví dụ, cho đoạn A hiển thị chữ nghiêng, những đoạn khác thì hiển thị theo kiểu khác. Trong trường hợp đó, đoạn A có thể có bộ nhận dạng lớp là .slant.

### Ví dụ 5

```
<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      .water { color:blue }
      .danger { color:red }
    </STYLE>
  </HEAD>
  <BODY>
    <P class=water>test water
    <P class=danger>test danger
    <P> no style
    <BR>
    <EM class=danger>italic</EM>
  </BODY>
</HTML>
```



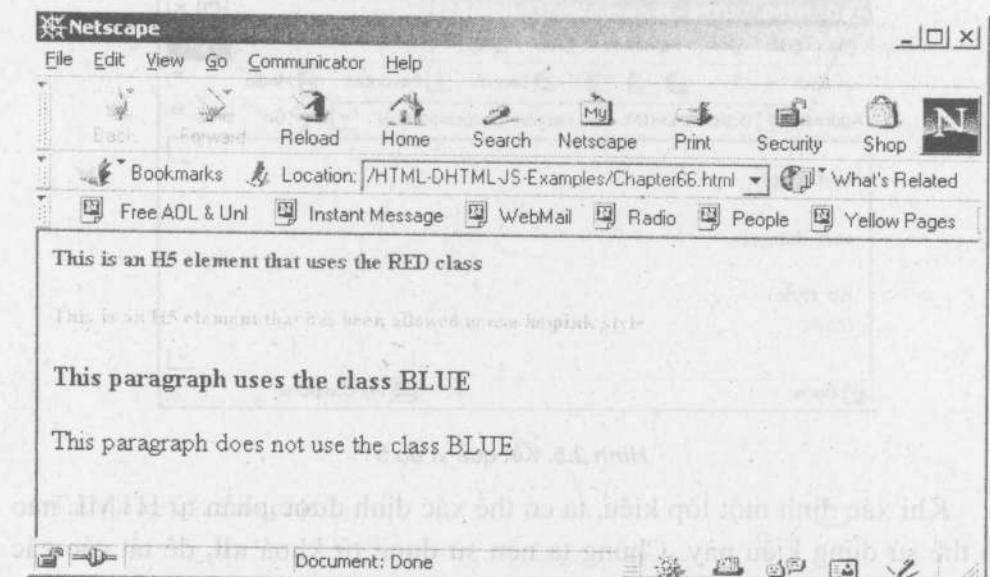
Hình 3.5. Kết quả ví dụ 5

Khi xác định một lớp kiểu, ta có thể xác định được phần tử HTML nào có thể sử dụng kiểu này. Chúng ta nên sử dụng từ khoá **all**, để tất cả các phần tử đều có thể sử dụng nó.

## Ví dụ 6

```
<HTML>
  <HEAD>
    <STYLE type="text/css">
      .HOTPINK {color:hotpink;}
      P.BLUE {color:blue; font-weight:bold;}
      H5.RED {color:red; font-weight:bold;}
    </STYLE>
  </HEAD>
  <H5 CLASS=RED>This is an H5 element that uses the RED class</H5>
  <H5 CLASS=hotpink>This is an H5 element that has been allowed to use hotpink style.</H5>
  <BODY bgColor=lavender>
    <P CLASS=BLUE>This paragraph uses the class BLUE</P>
    <P>This paragraph does not use the class BLUE</P>
    <P CLASS=hotpink>This paragraph is hotpink
  </BODY>
</HTML>
```

**Chú ý:** Xem kết quả trong Netscape



Hình 3.6. Kết quả ví dụ 6

### - Selector ID

Selector ID sử dụng thuộc tính ID của phần tử HTML. Selector ID được dùng để áp dụng một kiểu vào riêng một phần tử nào đó trên trang Web. Ví dụ, có thể sử dụng một selector ID để áp dụng một kiểu đặc biệt nào đó cho phần tử <H2>. Điều đó không có nghĩa là một kiểu tương tự được áp dụng cho một phần tử <H2> khác trên trang đó, nếu không được xác định. Tương tự với việc sử dụng kiểu nội tuyến mà nhờ đó có thể áp dụng riêng cho một phần tử nào đó. Selector ID được bắt đầu bằng dấu thăng (#).

#### Ví dụ 7

```
<HTML>
```

```
    <HEAD>
```

```
        <TITLE> ID Selectors</TITLE>
```

```
        <STYLE TYPE="text/css">
```

```
            #control {color:red}
```

```
        </STYLE>
```

```
    </HEAD>
```

```
    <BODY>
```

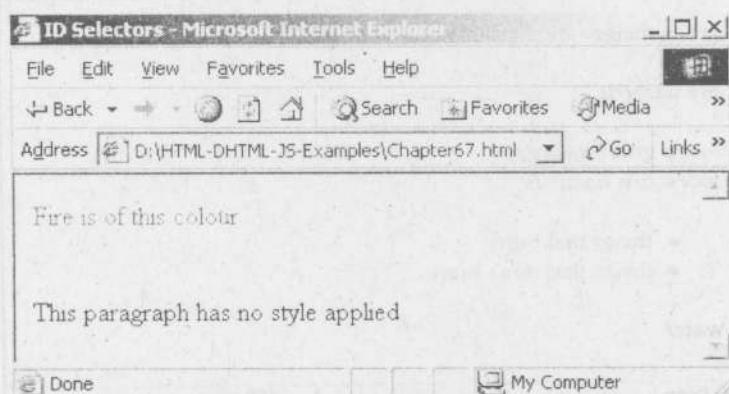
```
        <P id="control">Fire is of this color</H2>
```

```
        <BR>
```

```
        <P>This paragraph has no style applied
```

```
    </BODY>
```

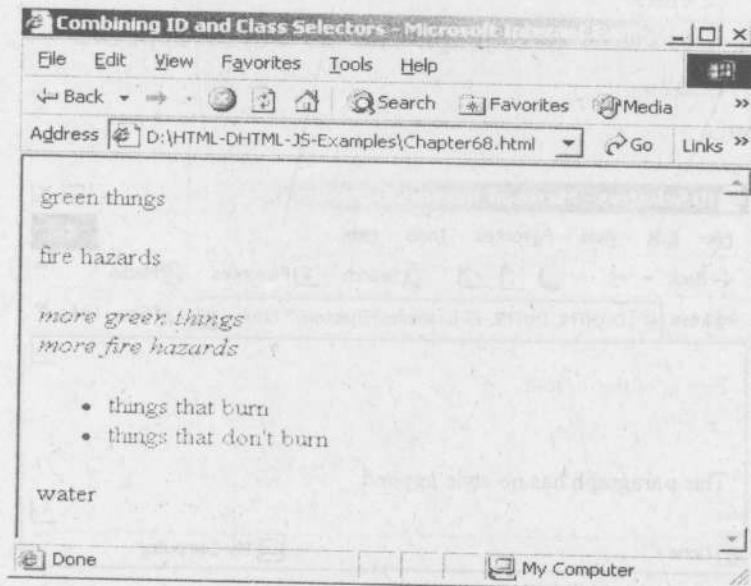
```
</HTML>
```



Hình 3.7. Kết quả ví dụ 7

## Ví dụ 8

```
<HTML>
    <HEAD>
        <TITLE> Combining ID and Class Selectors</TITLE>
        <STYLE TYPE="text/css">
            .forest {color:green}
            .danger {color:red}
            #control {color:blue}
        </STYLE>
    </HEAD>
    <BODY>
        <P class=forest>green things
        <P class=danger>fire hazards
        </P>
        <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>
```



Hình 3.8. Kết quả ví dụ 8

### - Selector ngữ cảnh

Selector ngữ cảnh liên quan đến ngữ cảnh của phần tử. Ví dụ, khi có hai phần tử cùng giá trị, phần tử chính hay phần tử cha có một phần tử con bên trong. Để thay đổi kiểu của phần tử con, ta phải sử dụng selector theo ngữ cảnh. Điều này dựa trên khái niệm kế thừa, phần tử con kế thừa kiểu được gán cho thẻ cha.

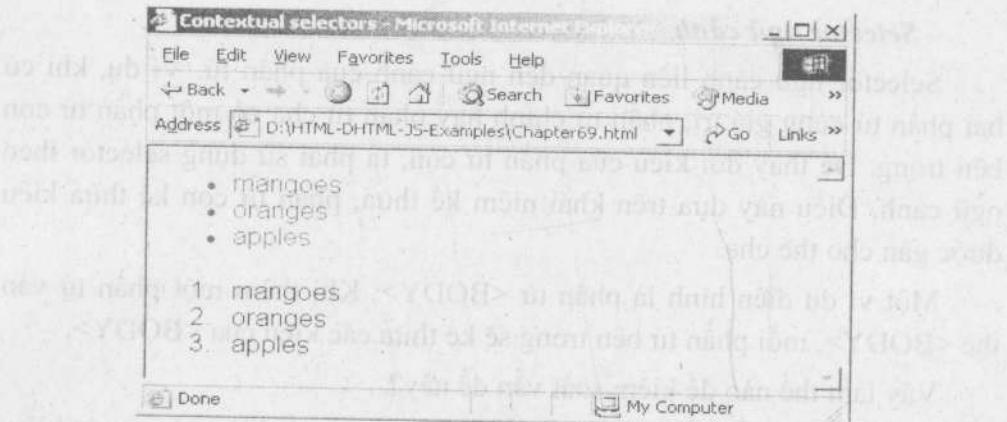
Một ví dụ điển hình là phần tử <BODY>. Khi thêm một phần tử vào thẻ <BODY>, mỗi phần tử bên trong sẽ kế thừa các kiểu của <BODY>.

Vậy làm thế nào để kiểm soát vấn đề này?

Để tắt cả các phần tử trên trang Web không xuất hiện cùng một kiểu, cần phải có sự thay đổi cụ thể đối với các phần tử con, hay nói một cách khác đó là sự nạp chồng kế thừa.

### Ví dụ 9

```
<HTML>
    <HEAD>
        <TITLE>Contextual selectors</TITLE>
        <STYLE TYPE="text/css">
            BODY {color:blue;
                   background:lavender;
                   font-family:Arial;}
            UL {color:red }
        </STYLE>
    <HEAD>
    <BODY>
        <UL>
            <LI>mangoes
            <LI>oranges
            <LI>apples
        </UL>
        <OL>
            <LI>mangoes
            <LI>oranges
            <LI>apples
        </OL>
    </BODY>
</HTML>
```



Hình 3.9. Kết quả ví dụ 9

Selector UL trong style sheet xác định các mục trong danh sách (list item) được hiển thị màu đỏ. Chúng kế thừa phông chữ Arial từ khai báo BODY, màu đỏ từ khai báo UL. Nếu xác định font-family trong khai báo UL, nó sẽ nạp chồng lên khai báo của selector BODY. Không có selector OL trong style sheet, vì thế các thuộc tính của OL kế thừa từ selector BODY.

## 2.4. Kết hợp, liên kết và chèn một style sheet vào tài liệu HTML

Có một số cách ta có thể kết hợp style sheet với HTML

1. Phần tử STYLE
2. Thuộc tính STYLE
3. Phần tử LINK

### – Phần tử STYLE

Phần tử STYLE được chèn vào phần tử <HEAD> của tài liệu, tất cả các quy tắc được định nghĩa giữa thẻ mở và thẻ đóng. Khi hiển thị các trang, chỉ tài liệu nào có nhúng STYLE mới được tác động. Ví dụ:

```
<Style Type = "text/css">
    H1 {color:maroon;}
    P {color:hotpink; font-family:Arial;}
</Style>
```

Ví dụ trên là cách sử dụng phần tử <STYLE>

### **– Thuộc tính STYLE**

Thuộc tính STYLE được sử dụng để áp dụng style sheet cho từng phần tử. Khi sử dụng thuộc tính style, ta có thể bỏ qua phần tử STYLE và đặt khai báo trực tiếp vào thuộc tính style trong thẻ mở của phần tử. Ví dụ:

```
<H2 style="color: green;  
font-family: Arial">  
</H2>
```

Ta chỉ nên dùng kiểu này khi thay đổi kiểu cho một phần tử cụ thể nào đó. Nếu muốn áp dụng cùng kiểu trên toàn bộ tài liệu thì đây không phải là cách hay. Ví dụ trên là cách sử dụng thuộc tính STYLE.

### **– Phần tử Link**

Nếu muốn sử dụng phần tử LINK, thì Style sheet phải là một tài liệu riêng. Sau đó chúng ta phải thêm địa chỉ Web của style sheet vào. Ví dụ :

```
<LINK REL = "style sheet"  
      HREF = "stylesmine.css"  
      Type = "text/css" >
```

Thuộc tính “rel=style sheet” phải được khai báo nếu không trình duyệt sẽ không tải được style sheet .

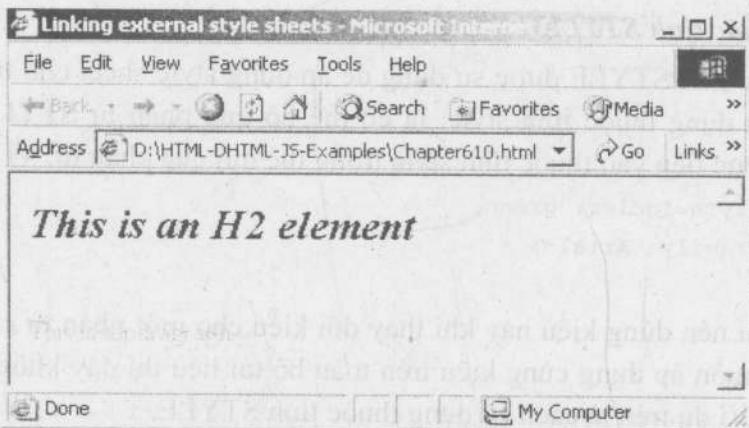
### **Ví dụ 10**

#### **Sheet1.css**

```
H2 {color:blue; font-style:italic;}  
P {color: limegreen;}
```

#### **Tệp.htm**

```
<HTML>  
  <HEAD>  
    <TITLE>Linking external style sheets</TITLE>  
    <LINK REL=STYLE SHEET TYPE="text/css"  
          HREF="sheet1.css">  
  </HEAD>  
  <H2>This is an H2 element</H2>  
  <BR>  
  <BODY>  
    <P>This is a paragraph  
  </BODY>  
</HTML>
```



Hình 3.10. Kết quả ví dụ 10

## 2.5. Thiết lập thuộc tính trong style sheet

Chúng ta có thể thiết lập nhiều thuộc tính trong style sheet. Bảng 3.1 cho thấy nhiều thuộc tính có thể được sử dụng trong style sheet.

Bảng 3.1

Thuộc tính	Tên
Các thuộc tính font	Font
	font-size
	font-style
Các thuộc tính văn bản	text-align
	text-indent
	vertical-align
Các thuộc tính hộp	border
	border-width
	border-bottom
	border-color
Các thuộc tính về vị trí	Clip
	height
	Left
	Top
	z-index

# *Chương 4*

## **NỀN TẢNG CƠ BẢN VÀ CÚ PHÁP JAVA SCRIPT**

Java Script là một ngôn ngữ kịch bản (scripting language) được dùng để tạo các script ở máy client (client-side script) và máy server (server-side script). Các script ở máy client được thực thi tại trình duyệt, các script ở máy server được thực hiện trên server. Chương này sẽ giới thiệu về ngôn ngữ Java Script và cách chèn một script vào trong tài liệu HTML.

### **1. GIỚI THIỆU VỀ JAVA SCRIPT**

HTML ban đầu được phát triển như là một định dạng của tài liệu có thể chuyển dữ liệu trên Internet. Tuy nhiên, không lâu sau đó, trọng tâm của HTML nặng tính hàn lâm và khoa học dần chuyển hướng sang người dùng thường nhật vì ngày nay người dùng xem Internet như là một nguồn thông tin và giải trí. Các trang Web ngày càng mang tính sáng tạo và đẹp mắt hơn nhằm thu hút nhiều người dùng hơn. Nhưng thực chất kiểu dáng và nội dung bên trong vẫn không thay đổi. Và người dùng hầu như không thể điều khiển trên trang Web mỗi khi nó được hiển thị.

Java Script được phát triển như là một giải pháp cho vấn đề nêu trên. Java Script là một ngôn ngữ kịch bản được Sun Microsystems và Netscape phát triển. Nó được dùng để tạo các trang Web động và tương tác trên Internet. Đối với những người phát triển HTML, Java Script rất hữu ích trong việc xây dựng các hệ thống HTML có thể tương tác với người dùng.

#### **1.1. Java Script là gì?**

Sun Microsystems đã viết ra một ngôn ngữ phức tạp và mạnh mẽ mà chúng ta đã biết đó là ngôn ngữ Java. Mặc dù Java có tính khả dụng cao nhưng nó lại phù hợp nhất đối với các nhà lập trình có kinh nghiệm và cho các công việc phức tạp hơn. Netscape Communications nhận thấy nhu cầu cần một ngôn ngữ thiết kế web có khả năng tương tác với người sử dụng

hay với các Java Applet, dễ sử dụng ngay cả với những người lập trình ít kinh nghiệm.

Live Script là một ngôn ngữ mới chỉ ở dạng phác thảo, tuy nhiên nó hấp dẫn người sử dụng vì hứa hẹn sẽ đáp ứng tốt những yêu cầu trên. LiveScript được thiết kế theo tinh thần của nhiều ngôn ngữ script đơn giản nhưng nó lại có tính khả dụng cao được thiết kế đặc biệt để xây dựng các trang Web (chẳng hạn như HTML và các form tương tác). Để giúp ‘bán chạy’ ngôn ngữ mới này, Netscape hợp tác với Sun cho ra đời ngôn ngữ Java Script. Trên thực tế, Microsoft là người tiên phong triển khai phiên bản của Java Script (còn có tên là Jscript), nhưng họ không sử dụng các đặc tả chính thức của Java Script.

Mục tiêu của Java Script là nhằm cung cấp cho các nhà phát triển Web một số khả năng và quyền điều khiển chức năng cho trang Web. Mã Java Script có khả năng nhúng trong tài liệu HTML để điều khiển nội dung của trang Web và kiểm tra sự hợp lệ của dữ liệu mà người dùng nhập vào. Khi một trang hiển thị trong trình duyệt, các câu lệnh được trình duyệt thông dịch và thực thi.

## 1.2. Định nghĩa

*Java Script là một ngôn ngữ kịch bản dựa trên đối tượng nhằm phát triển các ứng dụng Internet chạy trên phía client và phía server.*

Các ứng dụng client được chạy trong một trình duyệt như Netscape Navigator hoặc Internet Explorer, và các ứng dụng server chạy trên một Web server như Microsoft’s Internet Information Server hoặc Netscape Enterprise Server.

## 1.3. Hiệu ứng và quy tắc Java Script

Java Script là một ngôn ngữ lập trình được nhúng trong các trang HTML. Java Script nâng cao tính động và khả năng tương tác cho web-site bằng cách sử dụng các hiệu ứng của nó như thực hiện các phép tính, kiểm tra form, viết các trò chơi, bổ sung các hiệu ứng đặc biệt, tùy biến các chọn lựa đồ họa, tạo ra các mật khẩu bảo mật và hơn thế nữa.

Chúng ta có thể sử dụng Java Script để :

➤ **Tương tác với người dùng :** Chúng ta có thể viết mã để xử lý các sự kiện. Các sự kiện này có thể phát sinh bởi người dùng khi nhấp chuột hay được phát sinh từ hệ thống khi định lại kích thước của trang...

➤ **Thay đổi nội dung động** : Mã Java Script có thể dùng để thay đổi nội dung và vị trí các phần tử một cách động trên một trang nhằm đáp lại sự tương tác với người dùng.

➤ **Kiểm tra tính hợp lệ dữ liệu** : Chúng ta có thể viết mã nhằm kiểm tra tính hợp lệ của dữ liệu do người dùng nhập vào trước khi nó được gửi lên Web server để xử lý.

Giống như các ngôn ngữ khác, Java Script cũng tuân thủ một số quy tắc ngữ pháp căn bản. Việc nắm vững các quy tắc ngữ pháp này có thể giúp ta đọc được script và tự viết các script không bị lỗi.

Một số trong các luật này bao gồm:

➤ **Dùng Caps** : Java Script phân biệt chữ hoa chữ thường.

➤ **Dùng Pairs** : Trong Java Script, luôn luôn có cặp ký hiệu mở và đóng. Lỗi sẽ xuất hiện khi bỏ sót hoặc đặt sai một trong hai ký hiệu này.

➤ **Dùng Spaces (các ký tự trắng)** : Như HTML, Java Script thường bỏ qua ký tự trắng. Trong Java Script, ta có thể thêm vào các ký tự trắng hoặc các tab giúp cho việc đọc hoặc việc sửa các tệp script dễ dàng.

➤ **Dùng Chú thích (Comments)** : Các chú thích để ghi chú về chức năng của đoạn script, thời gian và người tạo ra đoạn script.

Mặc dù cả client-side Java Script và server-side Java Script đều dựa trên một ngôn ngữ nền tảng như nhau, nhưng mỗi loại còn có thêm những tính năng chuyên biệt phù hợp với môi trường mà nó chạy. Nghĩa là, client-side Java Script bao gồm các đối tượng được định nghĩa sẵn chỉ có thể sử dụng trên trình duyệt, server-side Java Script bao gồm các đối tượng và các hàm được định nghĩa sẵn chỉ có thể sử dụng trong các ứng dụng phía server (server-side applications).

#### 1.4. Các công cụ Java Script, IDE và môi trường thực thi

Các công cụ sinh mã Java Script và IDE giúp tạo ra mã Java Script rất hữu hiệu. Các công cụ này còn giúp ta nhanh chóng phát triển website của mình.

Một vài công cụ Java Script và IDE được đề cập dưới đây:

➤ **Dialog Box** : Công cụ này tự động tạo mã để sinh ra các hộp thoại tùy biến trên các trình duyệt khác nhau (alert, confirm, prompt,...) mang lại kiểu dáng chuyên nghiệp cho website.

➢ ***Pop-up Menu builder*** : Chỉ cần đưa vào các lựa chọn, công cụ này sẽ tự động tạo ra các pop-up menu trên các trình duyệt khác nhau.

➢ ***Remotes*** : Tự động sinh mã để mở ra một cửa sổ popup.

Như chúng ta đã biết, Java Script có thể được chạy trên máy khách (client) và máy chủ (server). Bên phía máy khách, trình duyệt sẽ thực thi mã lệnh Java Script trong trang web khi mở nó. Bên phía máy chủ, mã Java Script sẽ được thực thi tại máy chủ và do máy chủ thực hiện.

➢ ***Client-side Java Script***

Khi máy client yêu cầu một trang HTML, server sẽ kiểm tra xem trang đó có chứa script hay không. Nếu nó chứa các client-side script, server sẽ chuyển toàn bộ tài liệu bao gồm mã lệnh Java Script và nội dung HTML của nó cho trình duyệt. Khi trình duyệt nhận được tài liệu đó, nó thực thi các mã lệnh HTML và Java Script mà không cần bất kỳ sự tương tác nào với server.

➢ ***Java Script trên Web Server***

Chúng ta có thể nhúng các lệnh Java Script chạy trên server (server-side script) vào trong tài liệu HTML. Quá trình tạo ra các ứng dụng server-side là một quá trình gồm hai giai đoạn.

1. Các trang HTML có chứa các câu lệnh Java Script của cả client-side và server-side đều được tạo ra cùng với các file Java Script. Tất cả các file này sẽ được biên dịch thành dạng mã thực thi được gọi là bytecode.

2. Khi trình duyệt yêu cầu trang HTML, run-time engine sẽ thực thi mã lệnh server-side Java Script rồi trả trang HTML về cho trình duyệt.

Một số công dụng của script server-side bao gồm:

- Kết nối vào các cơ sở dữ liệu.
- Chia sẻ thông tin cho những người dùng của một ứng dụng.
- Truy cập vào hệ thống file trên server.

## 1.5. Nhúng Java Script vào trong trang Web

Chúng ta có thể chèn các lệnh Java Script vào trong một tài liệu HTML theo những cách sau đây:

1. Nhúng các câu lệnh trực tiếp vào trong tài liệu bằng cách sử dụng thẻ <SCRIPT>.

2. Liên kết file nguồn Java Script với tài liệu HTML.

3. Đặt các biểu thức Java Script làm giá trị cho thuộc tính của thẻ HTML.

4. Dùng như trình xử lý sự kiện trong các thẻ HTML.

Chi tiết có thể xem trong một số ví dụ cụ thể sau :

➤ **Dùng thẻ SCRIPT**

Mã Java Script cụ thể được nhúng vào trong tài liệu HTML bằng thẻ SCRIPT. Chúng ta có thể nhúng nhiều script vào trong cùng một tài liệu, mỗi script nằm trong một thẻ SCRIPT. Khi trình duyệt gặp phải một thẻ <SCRIPT> nào đó, nó sẽ đọc từng dòng một cho đến khi gặp thẻ đóng </SCRIPT>. Tiếp đến nó sẽ kiểm tra lỗi trong các câu lệnh Java Script. Nếu gặp lỗi, nó sẽ cho hiển thị lỗi đó trong chuỗi các hộp cảnh báo (alert boxes) trên màn hình. Nếu không có lỗi, các câu lệnh sẽ được biên dịch sao cho máy tính có thể hiểu được lệnh đó.

Cú pháp như sau:

```
<script language="Java Script">  
  <!--  
  Java Script statements;  
  //-->  
</script>
```

- Thuộc tính language trong thẻ script chỉ ra ngôn ngữ mà trình duyệt sẽ dùng để biên dịch script. Chúng ta cũng có thể chỉ rõ phiên bản Java Script nào sẽ được trình duyệt sử dụng. Ví dụ:

```
<script language="Java Script1.2">
```

- <!--statements //--> là các thẻ chú thích. Những thẻ này được dùng để báo cho trình duyệt bỏ qua các câu lệnh chứa trong nó. <! là thẻ mở của thẻ chú thích, //--> là thẻ đóng. Các thẻ này không bắt buộc phải có, nhưng ta nên đưa chúng vào trong các đoạn script. Chỉ có Netscape 2.0 và các phiên bản sau đó mới hỗ trợ Java Script. Các thẻ chú thích đảm bảo các phiên bản cũ hoặc các trình duyệt không hỗ trợ Java Script sẽ bỏ qua các câu lệnh được nhúng trong tài liệu HTML.

Các câu lệnh Java Script phải được kết thúc bằng dấu chấm phẩy ( ; ).

Trong ví dụ sau đây, ngôn ngữ script được đặt là Java Script. Các thẻ chú thích được dùng để các phiên bản cũ của trình duyệt bỏ qua script nằm trong các thẻ chú thích. Các phiên bản trình duyệt mới sẽ thực thi các câu lệnh Java Script.

## Ví dụ 1

```
<HTML>
  <HEAD>
    <script language = "Java Script">
      <!-- hide from older browsers>
      document.write("Welcome to the world of Java Script");
      // Script hiding ends here -->
    </script>
  </head>
  <body>
    <P>Enjoy the learning experience!!!</P>
  </BODY>
</HTML>
```

Kết quả:



Hình 4.1. Dùng Java Script

Trên lý thuyết, các câu lệnh Java Script có thể đặt ở bất kỳ nơi nào trong tài liệu HTML. Tuy nhiên, nên đặt các câu lệnh script trong phần `<head>` và `</head>`. Điều này đảm bảo tất cả các câu lệnh đều được đọc và biên dịch trước khi nó được gọi từ trong phần BODY.

### ➤ Dùng file bên ngoài

Thường các câu lệnh Java Script được nhúng trong một tài liệu HTML. Tuy nhiên, chúng ta có thể tạo ra một file riêng chứa mã Java Script. File này có thể được liên kết với một tài liệu HTML. Thuộc tính SRC (source) của thẻ SCRIPT dùng để chỉ ra file chứa Java Script mà nó cần sử dụng.

Khi xác định file nguồn, ta có thể dùng tên đường dẫn tương đối và tuyệt đối ở trong thuộc tính SRC.

```
<script language="Java Script" src="filename.js">  
</script>
```

Đó là file văn bản chứa các mã lệnh Java Script, tên file có phần mở rộng là “.js”. Nó chỉ có thể chứa các câu lệnh và các hàm Java Script. Ta không thể đưa các thẻ HTML vào trong nó.

Trong ví dụ sau đây, có hai file được tạo ra. File thứ nhất - “test.htm” là một file tài liệu HTML. File thứ hai - “test.js” là file văn bản có chứa mã Java Script. File này được liên kết với file tài liệu HTML.

### Ví dụ 2

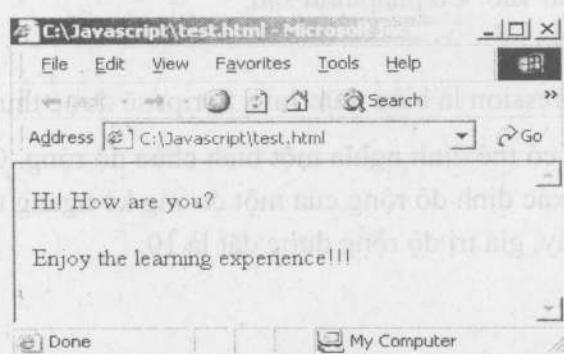
#### File HTML: (test.htm)

```
<HTML>  
  <HEAD>  
    <script src = "test.js"> </script>  
  </head>  
  <body>  
    <P>Enjoy the learning experience!!!  
  </body>  
</HTML>
```

#### File nguồn Java Script: (test.js)

```
document.write ("Hi! How are you?")
```

Kết quả



Hình 4.2. Dùng Java Script – File ngoài

**Chú ý:** Lưu các file **test.htm** và **test.js** trong cùng một thư mục.

Đây là một ví dụ đơn giản mô tả tính năng liên kết các file riêng chứa mã lệnh Java Script. Tuy nhiên, việc thực hiện chức năng liên kết các file sẽ rất có lợi khi ta muốn chia sẻ các hàm cho nhiều tài liệu HTML.. Trong trường hợp này, chúng ta có thể tạo ra một file .js với các hàm thông thường. File này được liên kết với các tài liệu cần nó. Nếu ta muốn điều chỉnh hoặc thêm vào một vài hàm, ta chỉ cần thực hiện thay đổi trong một file mà thôi thay vì phải trong nhiều tài liệu HTML.

Duôi .js phải được server ánh xạ đến kiểu MIME là application/x-Java Script. Server lúc đó sẽ gửi nó trong phần đầu (header) của giao thức HTTP.

Để ánh xạ đuôi .js với kiểu MIME ta thêm dòng code sau vào file chứa các kiểu MIME trong thư mục config của server. Sau đó ta sẽ khởi động lại server.

```
type=application/x-Java Script exts=js
```

Nếu server không ánh xạ đuôi .js đến application/x-Java Script MIME thì trình duyệt sẽ không tải đúng file Java Script được xác định trong thuộc tính SRC.

#### **➤ *Đặt các biểu thức Java Script là giá trị cho các thuộc tính của thẻ HTML***

Chúng ta có thể dùng biểu thức Java Script làm giá trị cho thuộc tính của thẻ HTML. Các giá trị đó được thực hiện một cách động mỗi khi trang được trình duyệt tải vào. Cú pháp như sau:

```
& {expression};
```

Trong đó expression là biểu thức Java Script sẽ được thực hiện.

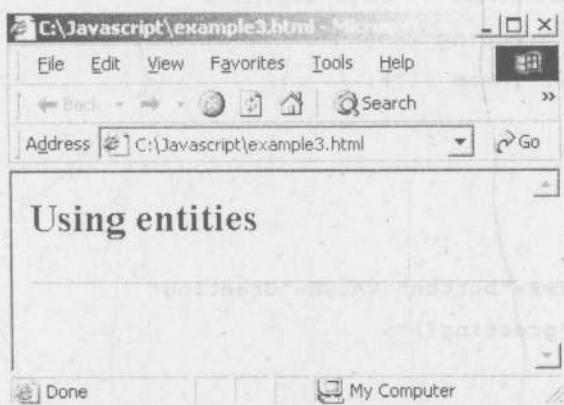
Chẳng hạn ta có thể định nghĩa một biến chứa độ rộng. Chúng ta có thể dùng biến này để xác định độ rộng của một đường kẻ ngang trên trang Web. Trong ví dụ sau đây, giá trị độ rộng được đặt là 10:

#### **Ví dụ 3**

```
<HTML>
  <HEAD>
    <SCRIPT>
      var linewidth = 10;
    </SCRIPT>
```

```
<BODY>
    <H2>Using entities</H2>
    <HR width="{linewidth};" align = "left">
</BODY>
</HTML>
```

Khi mã được thực thi, phần tử HR sẽ dùng giá trị của biến linewidth như minh họa trong hình 13.3.



Hình 4.3. Dùng các biểu thức Java Script

Một biểu thức Java Script chỉ có thể được sử dụng bên phải của phần tên/giá trị (name/value pair) trong thẻ HTML. Ví dụ, khi chúng ta dùng HR WIDTH="&{linewidth};10%" câu lệnh sẽ được dịch là:

```
<HR WIDTH="10%">
```

Nếu chúng ta sử dụng <H4>&{myTitle}</H4>, chuỗi &myTitle sẽ được hiển thị thay vì giá trị của biến myTitle.

#### ➤ *Dùng Java Script trong trình xử lý sự kiện*

Chúng ta có thể tạo một trình xử lý sự kiện cho một thẻ HTML dùng mã Java Script. Một sự kiện là một hành động được hỗ trợ bởi một đối tượng. Một trình xử lý sự kiện là đoạn mã sẽ được thực thi nhằm đáp trả một sự kiện. Cú pháp là:

```
<TAG event handler="Java Script code">
```

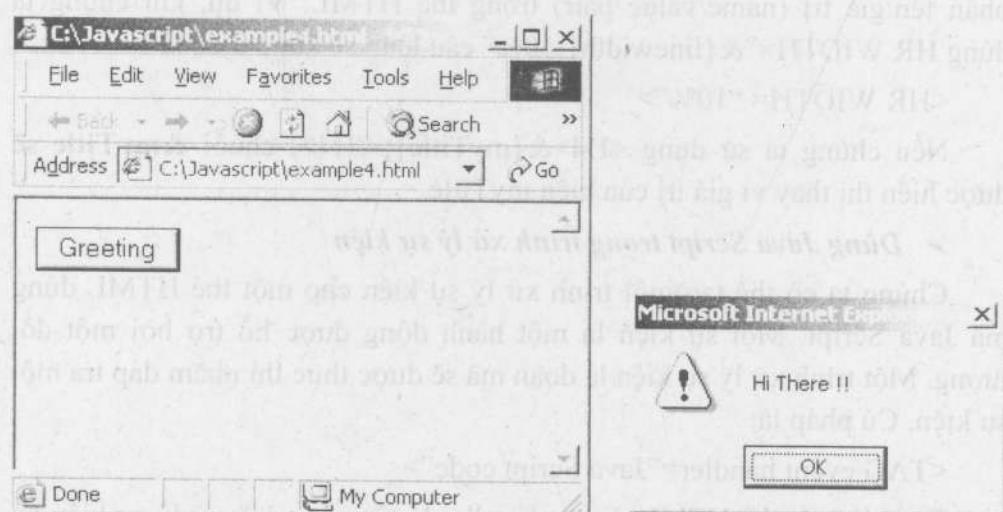
TAG là một thẻ HTML. Event handler là tên của trình xử lý sự kiện và Java Script mã là các câu lệnh Java Script được thực thi khi sự kiện được kích hoạt.

Trong ví dụ sau đây, sự kiện onClick được kích hoạt khi người dùng nhấp chuột vào phần tử BUTTON. Trình xử lý sự kiện được gọi để đáp trả sự kiện đó. Trình xử lý sự kiện có chứa mã Java Script được thực thi bởi trình duyệt.

#### Ví dụ 4

```
<HTML>
<SCRIPT LANGUAGE = "Java Script">
function greeting () {
    alert ("Hi There !!");
}
</SCRIPT>
<BODY>
<FORM>
    <INPUT TYPE="button" VALUE="Greeting"
        onClick="greeting ()">
</FORM>
</BODY>
</HTML>
```

Kết quả



Hình 4.4. Cảnh báo (alert) sử dụng Java Script

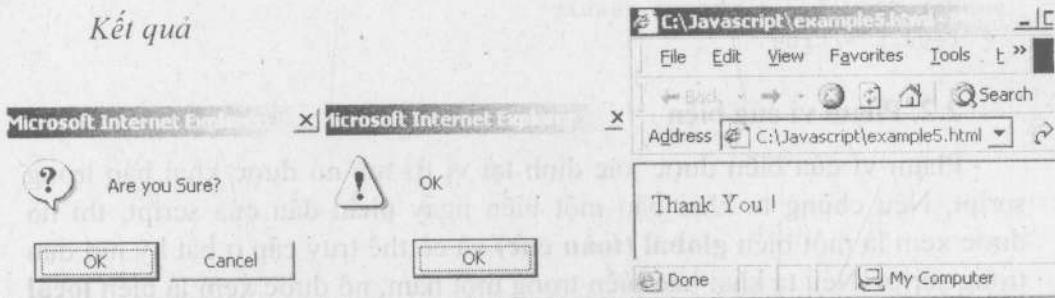
## 1.6. Ví dụ đơn giản sử dụng các hộp thông báo và phương thức write

Trong ví dụ minh họa dưới đây, hướng dẫn cách sử dụng các phương thức confirm (xác nhận), alert (cảnh báo) và write (ghi).

### Ví dụ 5

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Java Script">
confirm ("Are you Sure?");
alert("OK");
document.write(" Thank You !");
</SCRIPT>
</HEAD>
</HTML>
```

### Kết quả



Hình 4.5. Dùng Java Script – Confirm, Alert và Write sử dụng Java Script

## 2. CÁC BIẾN

Biến là một tham chiếu đến một vị trí trong bộ nhớ. Nó dùng để chứa các giá trị có thể thay đổi khi script đang được thực thi. Chúng ta phải đặt tên cho biến. Chẳng hạn ta có thể tạo một biến có tên First. Tại mỗi thời điểm thực hiện, biến có thể chứa một giá trị mới. Khi chúng ta muốn xem giá trị của biến hoặc thay đổi giá trị của nó, ta chỉ cần dùng tên của biến.

Các biến thường có các quy ước đặt tên như sau:

- Tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới ("\_").
- Tên biến có thể chứa chữ số.
- Java Script phân biệt rõ chữ hoa và chữ thường, vì vậy tên biến bao gồm các ký tự từ "A" đến "Z" (chữ hoa) và các ký tự từ "a" đến "z" (chữ thường).

## 2.1. Khai báo biến

Sử dụng từ khoá 'var' để khai báo biến. Biến sẽ tồn tại ngay sau khi ta khai báo nó. Ta có thể khởi tạo giá trị cho biến ngay khi khai báo:

```
var A = 10;
```

Cũng có thể khai báo biến bằng cách gán giá trị cho nó mà không cần từ khoá var.

```
A = 10
```

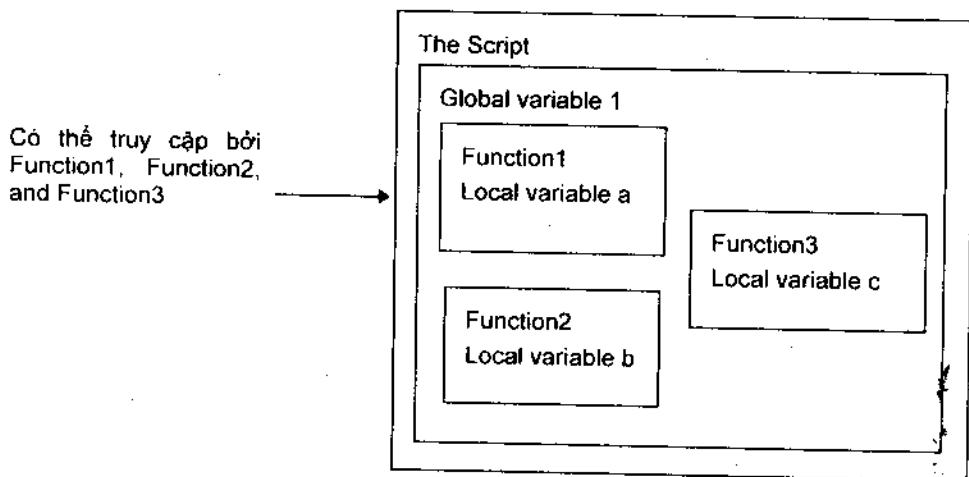
Ta có thể khai báo nhiều biến trên cùng một dòng bằng cách tách tên các biến bằng dấu phẩy.

Đây là một số ví dụ các biến:

```
var new_amount  
var answer = null  
var old_cost = 12.50  
var result = 'Unknown result'  
result = true
```

## 2.2. Phạm vi của biến

Phạm vi của biến được xác định tại vị trí mà nó được khai báo trong script. Nếu chúng ta khai báo một biến ngay phần đầu của script, thì nó được xem là một biến **global (tổng cục)** và có thể truy cập ở bất kỳ nơi đâu trong script. Nếu ta khai báo biến trong một hàm, nó được xem là biến **local (cục bộ)** và nó chỉ được sử dụng chỉ đối với hàm đó mà thôi. Những hàm khác trong script không thể truy xuất vào biến đó được. Hình 13.6 minh họa phạm vi của các biến trong script.



Hình 4.6. Phạm vi của biến (variables)

Các biến toàn cục không cần thiết phải sử dụng từ khoá var khi khai báo. Tuy nhiên, các biến cục bộ luôn luôn phải sử dụng từ khoá var khi khai báo.

### ➤ **Literals**

Các literals là những giá trị cố định mà ta có thể dùng trong script. Giá trị của literal không bị thay đổi trong quá trình thực hiện script.

### **2.3. Các kiểu dữ liệu**

Java Script có một tập hợp các kiểu dữ liệu như sau:

Kiểu dữ liệu	Mô tả
Numbers	Các số nguyên hoặc số thực. Ví dụ: 487 hoặc 25.95
Logical hoặc Boolean	True hoặc False
Strings	"Hello World"
Null	Keyword đặc biệt chỉ giá trị null.

Java Script luôn phân biệt các chữ hoa và chữ thường; null không giống như Null hoặc NULL

Dữ liệu luôn luôn thuộc một kiểu nào đó. Biến có kiểu dữ liệu phụ thuộc vào dữ liệu mà nó lưu trữ, kiểu dữ liệu của biến sẽ bị thay đổi khi giá trị của nó thay đổi.

Khi chúng ta khai báo một biến thì không cần phải chỉ ra kiểu dữ liệu của nó. Các kiểu dữ liệu sẽ tự động thay đổi nếu cần thiết trong quá trình thực thi script. Ví dụ:

```
var x = 10
```

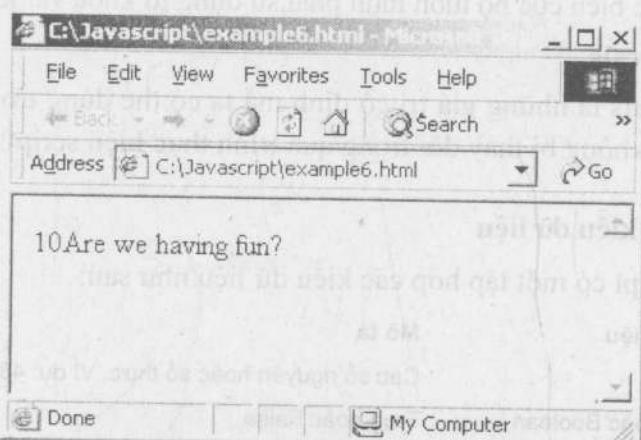
Sau đó, chúng ta có thể gán một chuỗi vào biến x trong script:

```
x = "Are you having fun?"
```

### **Ví dụ 6**

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE = "Java Script">
      var x = 10;
      document.write(x);
      var x = "Are we having fun?";
      document.write(x);
    </SCRIPT>
  </HEAD>
</HTML>
```

Kết quả :



Hình 4.7. Các biến Java Script.

Trong Java Script, ta có thể kết hợp hai biến thuộc các kiểu khác nhau. Ví dụ, nếu chúng ta kết hợp một chuỗi với một giá trị số với nhau thì Java Script sẽ chuyển đổi các giá trị số sang các chuỗi.

Ví dụ:

```
A = " This apple costs Rs." + 5
```

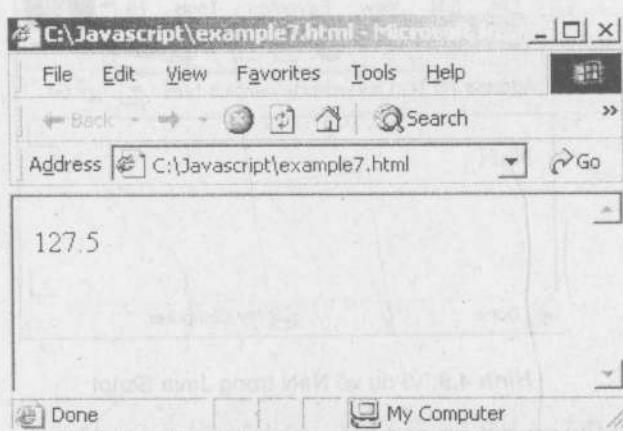
Nó sẽ cho ra kết quả là một chuỗi có giá trị "This apple costs Rs. 5"

Tuy nhiên, chúng ta không thể chuyển một chuỗi thành một số. Nếu chúng ta cộng một số thực 7.5 với một chuỗi là "12", kết quả sẽ là 127.5.

**Ví dụ 7**

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Java Script">
var A = "12" + 7.5;
document.write(A);
</SCRIPT>
</HEAD>
</HTML>
```

Kết quả :



Hình 4.8. Ví dụ sử dụng biến trong Java Script

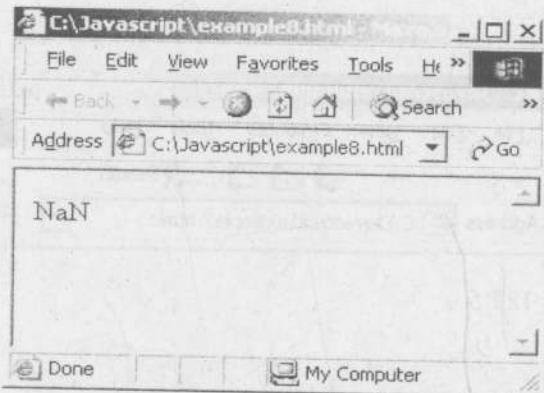
Trong Java Script có hàm parseInt() và parseFloat() có chức năng chuyển đổi các chuỗi thành các số nguyên hoặc các số thực. Chẳng hạn, hàm parseInt("15") sẽ chuyển đổi chuỗi "15" sang giá trị là một số nguyên. Hàm parseFloat("35.45") sẽ chuyển đổi một chuỗi thành một số thực là 34.45. Nếu hàm parseFloat() nhận ra một ký tự chứ không phải là một ký hiệu (+ hoặc -), một số (0 - 9), ký hiệu dấu thập phân hoặc một số mũ thì nó sẽ bỏ qua ký tự đó và tất cả các ký tự sau nó. Nếu ký tự đầu tiên không thể chuyển đổi được, hàm đó sẽ trả về giá trị "NaN" (Not a Number).

Trong ví dụ sau đây, chuỗi "a15" được chuyển vào hàm parseFloat(). Kết quả là NaN vì chuỗi đó không mở đầu bằng các ký tự có thể chuyển đổi được.

#### Ví dụ 8

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE = "Java Script">
      var x = "a15"
      a = parseFloat(x);
      document.write(a);
    </SCRIPT>
  </HEAD>
</HTML>
```

Kết quả :



Hình 4.9. Ví dụ về NaN trong Java Script

Literals có thể có bất kỳ các kiểu dữ liệu nào sau đây :

a) *Số nguyên* – Chúng có thể được biểu diễn trong hệ thập phân, hệ thập lục phân và hệ nhị phân.

b) *Số thực (Floating-point)* – Các literal số thực (Floating-point literals) phải có ít nhất một chữ số. Số đó có thể có dấu thập phân hoặc “e” hoặc “E” theo sau một số nguyên. Số nguyên có thể dương (“+”) hoặc âm (“-”). Ký hiệu e hay E được hiểu là “lũy thừa cơ số 10”. Ví dụ: 10.24, 1.20e+22, 4E-8, .1815,...

c) *Chuỗi* – là chuỗi rỗng hoặc các ký tự được đặt trong dấu ngoặc đơn ('Unexpected error') hoặc dấu ngoặc kép "Hi! How are you". Chuỗi đó phải bắt đầu và kết thúc bằng một dấu ngoặc của cùng một kiểu.

Sau đây là một số các ví dụ về literals chuỗi:

'Unexpected error'

"Hi! How are you"

'4531'

Khi dùng chuỗi, ta có thể chèn các ký tự đặc biệt vào chuỗi đó. Các ký tự đặc biệt sẽ thực hiện một công việc cụ thể. Ví dụ: chúng ta có thể thêm một ký tự chéo ngược (\) trong chuỗi đó. Dưới đây là bản liệt kê các ký tự đặc biệt và nghĩa của chúng:

Ký tự	Ý nghĩa
\b	backspace
\f	form feed
\n	new line (xuống dòng và đưa con trỏ về đầu dòng đó)
\r	Carriage return (đưa con trỏ về đầu dòng hiện tại)
\t	Tab

Những ví dụ dưới đây minh họa cách sử dụng các ký tự đặc biệt của Java Script:

➢ **Ký tự backspace**

```
<SCRIPT LANGUAGE = "Java Script">
    alert(" This is line one \b This is line two");
</SCRIPT>
```

➢ **Ký tự form feed (sang trang)**

```
<SCRIPT LANGUAGE = "Java Script">
    alert(" This is line one \f This is line two");
</SCRIPT>
```

➢ **Ký tự new line (xuống dòng)**

```
<SCRIPT LANGUAGE = "Java Script">
    alert(" This is line one \n This is line two");
</SCRIPT>
```

➢ **Ký tự carriage return (đưa con trỏ về đầu dòng hiện tại)**

```
<SCRIPT LANGUAGE = "Java Script">
    alert(" This is line one \r This is line two");
</SCRIPT>
```

➢ **Ký tự tab**

```
<SCRIPT LANGUAGE = "Java Script">
    alert(" This is line one \t This is line two");
</SCRIPT>
```

Ngoài các ví dụ nêu trên, chúng ta có thể chèn các ký tự khác trong một chuỗi bằng cách đặt trước nó ký tự chéo ngược (\). Đây được xem là **ký tự thoát** (*escaping character*).

Ký tự chéo ngược (\) được dùng để bỏ qua ý nghĩa sử dụng của ký tự đứng sau nó, vì nhiều ký tự đặc biệt được thiết kế sẵn để phục vụ một chức năng cụ thể nào đó. Ví dụ, nếu ta muốn cho hiển thị ký tự backslash, đây là một ký tự đặc biệt nên ta phải loại bỏ ý nghĩa của nó.

d) *Boolean* – Nó chỉ có thể nhận hai giá trị : True hoặc False. Kiểu dữ liệu này rất hữu dụng khi thực hiện các quyết định hoặc so sánh dữ liệu.

e) *Null* – Kiểu null chỉ có duy nhất một giá trị : null. Null ngụ ý là không có dữ liệu. Nó thực hiện chức năng là giữ chỗ trong một biến với ngụ ý là ở đó không có hữu dụng gì.

Số không hay một xâu rỗng và null là các giá trị khác nhau.

## 2.4. Các toán tử

Các toán tử thực hiện tính toán trên một hoặc nhiều biến hoặc giá trị (toán hạng) và trả về một giá trị mới. Ví dụ: toán tử + có thể cộng hai số để cho ra một số thứ ba. Các toán tử được dùng trong các biểu thức với các giá trị liên quan đến nhau nhằm thực hiện các phép toán hoặc so sánh các giá trị.

Java Script sử dụng cả các toán tử hai ngôi và các toán tử một ngôi.

➤ Toán tử hai ngôi cần hai toán hạng. Ví dụ:

$4 + 5$

trong đó 4 và 5 là các toán hạng và + là toán tử

➤ Toán tử một ngôi đòi hỏi phải có một toán hạng. Ví dụ:

$a++$  hoặc  $b++$

Các toán tử được phân loại tùy thuộc vào chức năng mà chúng thực hiện:

### 2.4.1. Toán tử số học

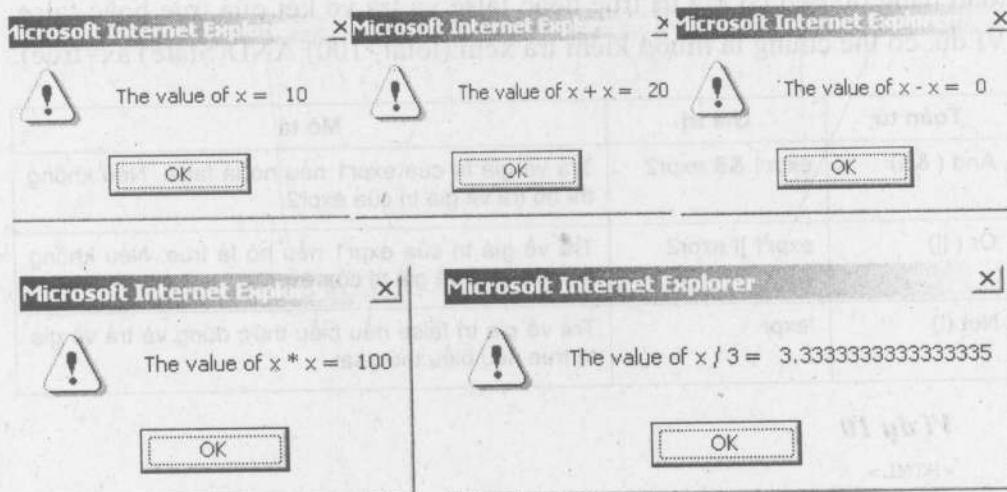
Các toán tử số học sử dụng các giá trị số (literals hay các biến) làm toán hạng của chúng và trả về một giá trị số. Các toán tử số học cơ bản là:

Toán tử	Mô tả	Ví dụ
+	Phép cộng	$A = 5 + 8$
-	Phép trừ	$A = 8 - 5$
/	Phép chia	$A = 20 / 5$
%	Phép chia lấy số dư	$10 \% 3 = 1$
++	Tăng lên một đơn vị. Toán tử này nhận một toán hạng. Giá trị của toán hạng sẽ tăng lên 1 đơn vị. Giá trị được trả về sẽ tùy thuộc vào toán tử ++ nằm sau hay nằm trước toán hạng.	$++x$ sẽ trả về giá trị của x sau khi tăng. $x++$ sẽ trả về giá trị của x trước khi tăng.
--	Giảm một đơn vị. Toán tử này nhận một toán hạng. Giá trị được trả về sẽ tùy thuộc vào toán tử -- nằm trước hay nằm sau toán hạng.	$-x$ sẽ trả về giá trị của x sau khi giảm. $x-$ sẽ trả về giá trị của x trước khi giảm.
-	Lấy số đối. Nó sẽ trả về số đối của toán hạng	Nếu $a$ là 5, thì $-a = -5$ .

### Ví dụ 9

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Java Script">
var x = 10;
alert ("The value of x = " + x);
alert ("The value of x + x = " + (x + x));
alert ("The value of x - x = " + (x - x));
alert ("The value of x * x = " + (x * x));
alert ("The value of x / 3 = " + (x / 3));
alert ("The value of x % 3 = " + (x % 3));
alert ("The value of --x = " + (--x));
alert ("The value of ++x = " + (++x));
alert ("The value of -x = " + (-x));
</SCRIPT>
</HEAD>
</HTML>
```

Dưới đây là một số hộp thông báo:



Hình 4.10. Các toán tử số học trong Java Script

#### 2.4.2. Các toán tử so sánh

Toán tử so sánh thường so sánh các toán hạng của nó và trả về một giá trị logic dựa trên phép so sánh đó đúng hay không. Các toán hạng có thể là

các giá trị số hoặc chuỗi. Khi so sánh chuỗi, nó sẽ dựa trên thứ tự các ký tự trong bảng chữ cái.

Toán tử	Mô tả	Ví dụ
$= =$	Bằng. Trả về giá trị true nếu các toán hạng bằng nhau.	$a == b$
$!=$	Không bằng. Trả về giá trị true nếu các toán hạng không bằng nhau.	$Var2 != 5$
$>$	Lớn hơn. Trả về giá trị true nếu toán hạng trái lớn hơn toán hạng phải.	$Var1 > var2$
$>=$	Lớn hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái lớn hơn hoặc bằng toán hạng phải.	$Var1 >= 5$ $Var1 >= var2$
$<$	Nhỏ hơn. Trả về giá trị true nếu toán hạng trái nhỏ hơn toán hạng phải.	$Var2 < var1$
$<=$	Nhỏ hơn hoặc bằng. Trả về giá trị true nếu toán hạng trái nhỏ hơn hoặc bằng toán hạng phải.	$Var2 <= 4$ $Var2 <= var1$

#### 2.4.3. Toán tử logic

Toán tử logic được dùng để kết hợp nhiều toán tử so sánh thành một biểu thức điều kiện. Một toán tử logic thường có hai toán hạng, mỗi một toán hạng đó đều có giá trị true hoặc false và trả về kết quả true hoặc false. Ví dụ, có thể chúng ta muốn kiểm tra xem (total>100) AND(StateTax=true).

Toán tử	Giá trị	Mô tả
And ( $&&$ )	$expr1 \&\& expr2$	Trả về giá trị của $expr1$ nếu nó là false. Nếu không thì nó trả về giá trị của $expr2$ .
Or ( $  $ )	$expr1    expr2$	Trả về giá trị của $expr1$ nếu nó là true. Nếu không thì nó sẽ trả về giá trị của $expr2$ .
Not ( $!$ )	$!expr$	Trả về giá trị false nếu biểu thức đúng và trả về giá trị true nếu biểu thức sai.

#### Ví dụ 10

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="Java Script">
      var x = 10;
      var y = 5;
      alert ("The value of x is " + x + "The value of y is " + y);
      alert("x AND y = " + (x && y));
    </SCRIPT>
  </HEAD>
<BODY>
</BODY>

```

```

    alert("x OR y = " + (x || y));
    alert("NOT x = " + (!x));

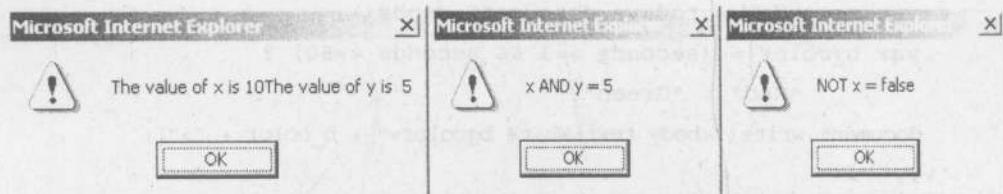
```

</SCRIPT>

</HEAD>

</HTML>

Kết quả :



Hình 4.11. Toán tử logic trong Java Script

#### 2.4.4. Toán tử chuỗi

Toán tử chuỗi nhận hai toán hạng và tạo ra một chuỗi mới bằng cách ghép hai chuỗi ban đầu lại với nhau.

Ví dụ:

```

x = 'yellow';
y = 'green';
z = x + y + 'white'; which means z is "yellowgreenwhite"
w = y + 9, which means w is "green9"

```

#### 2.4.5. Toán tử đánh giá

Một số toán tử ít sử dụng trong Java Script và không được xếp vào loại cụ thể nào. Những toán tử này được liệt kê dưới đây:

##### ➤ Toán tử điều kiện

(condition) ? trueVal : falseVal gán một giá trị xác định cho một biến nếu điều kiện đúng, và ngược lại gán một giá trị khác nếu điều kiện là false.  
Ví dụ:

```
status = (age >= 18) ? "adult" : "minor"
```

Nếu tuổi lớn hơn hoặc bằng 18, biến status được gán giá trị "adult". Nếu không, nó sẽ có giá trị "minor".

Trong ví dụ sau đây, phương thức getSeconds() được dùng để lấy số giây từ đối tượng Date và gán nó với biến "seconds". Nếu giá trị "seconds" lớn hơn hoặc bằng 3 hoặc nhỏ hơn hoặc bằng 50, thì màu nền của tài liệu chuyển sang màu đỏ (red). Ngược lại màu nền là màu xanh lục (green).

## Ví dụ 11

```
<html>
<head>
<script language="Java Script">
<!--
var todays_date = new Date();
var seconds = todays_date.getSeconds();
var b_color = (seconds >=3 && seconds <=50) ?
    "Red" : "Green";
document.write("<body text=White bgcolor=" + b_color + ">");
//-->
</script>
</head>
<h1>
<hr>
    Welcome to Aptech Limited.
<hr> </h1>
</body>
</html>
```

Kết quả :



Hình 4.12. Toán tử điều kiện trong Java Script

## > **typeof**

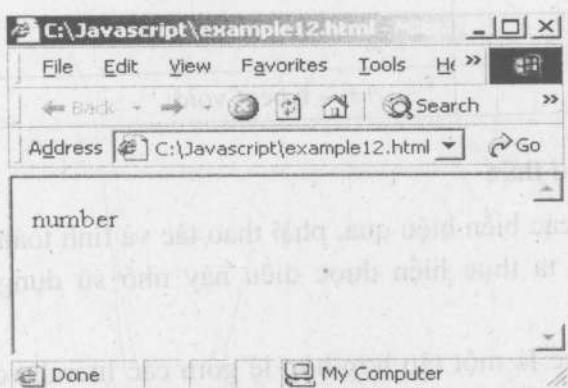
Toán tử typeof trả về chuỗi cho biết tên kiểu dữ liệu của toán hạng

### **Ví dụ 12**

```
<html>
  <head>
    <script language="Java Script">
      <!--
      Var x = 5;
      document.write(typeof(x));
      //-->
    </script>
  </head>
</html>
```

Toán tử typeof sẽ trả về: number

Kết quả:



Hình 4.13. Java Script: Toán tử typeof

### **2.4.6. Mức ưu tiên của các toán tử**

Khi có nhiều toán tử trong cùng một biểu thức, độ ưu tiên của toán tử xác định thứ tự thực hiện của toán tử đó. Một biểu thức được đọc từ trái sang phải và được thực hiện từ các toán tử có độ ưu tiên cao đến các toán tử có độ ưu tiên thấp hơn. Nếu chúng ta muốn thay đổi trật tự thực hiện của các toán tử, ta phải sử dụng các dấu ngoặc. Bảng dưới đây liệt kê độ ưu tiên của các toán tử từ thấp đến cao:

### 3. KIẾU TOÁN TỬ

Kiểu toán tử	Các toán tử đơn
Gán	= += -= *= /= %= <<= >>= >>>= &= ^=  =
Điều kiện	?:
logic -or	
logic -and	&&
bitwise-or	
bitwise-xor	^
bitwise-and	&
bằng/không bằng	== !=
quan hệ	< <= > >=
dịch bit	<< >> >>>
cộng/trừ	+ -
nhân/chia	* / %
Phủ định/tăng	! ~ - ++ -- typeof void

#### 3.1. Các biểu thức

Để sử dụng các biến hiệu quả, phải thao tác và tính toán chúng mỗi khi cần thiết. Chúng ta thực hiện được điều này nhờ sử dụng các biểu thức (expression).

Một biểu thức là một tập hợp hợp lệ gồm các literals, các biến và các toán tử để tính toán và trả về một giá trị đơn. Giá trị này có thể là một số, một chuỗi hay bất kỳ một giá trị logic nào đó.

Có hai dạng biểu thức. Ta có thể gán một giá trị vào một biến, chẳng hạn như `a = 10`, nó trở thành một câu lệnh gán. Lúc khác, ta có thể viết một biểu thức như `x*y`. Đây là một biểu thức nhưng nó không thực hiện phép gán.

Dưới đây là một số dạng biểu thức có sẵn trong Java Script.

- **Số học:** trả về giá trị là một số.
- **Logic:** trả về một giá trị boolean.
- **Chuỗi:** trả về giá trị là một chuỗi.

Các biểu thức liên kết các biến và literals với nhau thông qua toán tử.

### > Các biểu thức quy tắc

Biểu thức quy tắc là một mẫu tìm kiếm dùng để tìm kiếm các chuỗi ký tự cùng dạng trong một xâu. Dùng biểu thức quy tắc có thể tìm kiếm theo mẫu trong các chuỗi ký tự do người dùng nhập vào. Ví dụ, tạo ra một mẫu tìm kiếm gồm từ « cat » và sẽ tìm kiếm tất cả sự xuất hiện của từ này trong một xâu nào đó. Trong Java Script, một biểu thức quy tắc cũng được xem là một đối tượng.

Mẫu tìm kiếm của biểu thức quy tắc có thể bao gồm:

a) *Các mẫu đơn giản (Simple pattern)* : Thực hiện việc tìm kiếm chính xác theo những ký tự được chỉ ra trong mẫu. Ví dụ, ta có một mẫu tìm kiếm như sau:

/cat/

Nó sẽ tìm kiếm từ ‘cat’ trong một chuỗi và sẽ trả về từ ‘cat’ trong câu: “The cat sat on the mat” hoặc “Please hurry or we will not catch the train”.

Ký tự ( / ) chỉ ra vị trí bắt đầu và kết thúc của biểu thức cần tìm kiếm:

b) *Kết hợp các ký tự đơn giản và đặc biệt* : được dùng khi muốn tìm kiếm các chuỗi cùng dạng. Ví dụ, muốn tìm kiếm một chuỗi có một ký tự xuất hiện nhiều lần hay có nhiều ký tự trắng. Thông thường, dữ liệu do người dùng nhập vào thường có nhiều từ. Ví dụ, biểu thức /xy\*z/ sẽ tìm kiếm các chuỗi bắt đầu bằng x, có thể có nhiều ký tự y hay không có bên trong nó, kết thúc bởi z.

Trong chuỗi xxzzzyzxyyyzz, biểu thức sẽ trả về chuỗi con ‘xyyyz’.

Dưới đây là bảng liệt kê một số ký tự đặc biệt có thể sử dụng trong các biểu thức quy tắc.

Ký tự	Nghĩa	Ví dụ
\	Giải thoát nghĩa hằng (literal) của ký tự	/b/ tương ứng với ký tự 'b'. \b/ có thể kết hợp với bất kỳ từ nào bắt đầu bằng với b. /a*/ tương ứng với 0 hoặc nhiều ký tự a. /a*/ tương ứng với 'a*.'
^	Tương ứng ký tự đầu chuỗi nhập hay đầu dòng	/^A/ sẽ tìm kiếm "An apple."

\$	Tương xứng ký tự cuối chuỗi nhập hay cuối dòng	/\$/ sẽ tìm kiếm các ký tự t trong "eat" mà không phải "eater".
*	Tương xứng ký tự phía trước là 0 hoặc nhiều lần	/o*ch/ tương xứng 'oooooch'
+	Tương xứng ký tự phía trước là 1 hoặc nhiều lần	/r+/ tương xứng 'r' trong "break" và tất cả ký tự r trong "brrrrreak"
?	Tương xứng ký tự phía trước là 0 hoặc 1 lần.	/e?le?/ tương xứng le 'el' trong "angel" và 'le' trong "bangle."
.	(Đầu chấm thập phân) tương xứng bắt ký ký tự đơn ngoại trừ ký tự newline	
(x)	Tương xứng 'x' và ghi nhớ ký tự tương xứng đó	Ví dụ, /(boo)/ tương xứng 'boo' trong "boo said Casper" và ghi nhớ 'boo'
x y	Tương xứng 'x' hoặc 'y'	Ví dụ, /green ripe/ tương xứng là 'green' trong "green mangoes" và 'ripe' trong "ripe mangoes."
{n,m}	Tương xứng ít nhất n và nhiều nhất m số lần xuất hiện của ký tự phía trước n và m là số nguyên dương	Ví dụ, /o{1,3}/ không tương xứng trong "cold", ký tự 'o' trong "cold," hai ký tự oo đầu tiên trong "coold," và ba ký tự o đầu tiên trong "cooooold" Mặc dù có nhiều hơn ba ký tự o xuất hiện nhưng nó chỉ trả về ba ký tự xuất hiện đầu tiên
[xyz]	Tương xứng bắt ký nào nằm trong dấu ngoặc. Dấu nối được dùng để xác định một khoảng	[vxyz] tương tự như [v-yz] [0-9]
[^xyz]	Tương xứng với bất kỳ ký tự nào không nằm trong dấu ngoặc Dấu nối được dùng để xác định một khoảng	[^xyz] tương tự như [^x-z]
\d	Tương xứng ký tự số	/\d/ sẽ tương xứng 7 trong chuỗi "This is Bond7"
\s	Tương xứng với ký tự trắng đơn bao gồm space, tab, form feed, line feed. Tương tự [\f\n\r\t\v]	\\$w/ tương xứng ' fun' trong "good fun."
\t	Tương xứng một tab	
\w	Tương xứng ký tự alphanumeric bao gồm ký tự gạch dưới Tương tự [A-Za-z0-9].	\w/ tương xứng 'm' trong "mangoes", '6' trong "62.50" và '2' trong "2D."

## > Tạo một biểu thức quy tắc

Một biểu thức quy tắc là một mẫu tìm kiếm để tìm kiếm dữ liệu cùng dạng. Java Script xem một biểu thức quy tắc như một đối tượng. Vì vậy, phải tạo một biểu thức quy tắc trước khi sử dụng chúng. Chúng ta có thể tạo một biểu thức quy tắc bằng một trong hai cách sau:

– *Sử dụng khởi tạo đối tượng (Object initializer)* : Cách này trước đây được gọi là tạo đối tượng bằng cách sử dụng các ký hiệu nguyên dạng. Sau đó nó được chuyển thành object initializer để giống với thuật ngữ của C++. Nếu muốn tạo ra một thể hiện của một đối tượng, phải dùng một object initializer.

Cú pháp của việc khởi tạo đối tượng:

```
Objectname = {expression}
```

Trong đó **Objectname** là tên của đối tượng mới.

**Expression** là một khuôn mẫu để tạo đối tượng.

Ví dụ,

```
re = /xy+z/
```

Trong định nghĩa trên, một biểu thức quy tắc “xy+z” được tạo và gán cho đối tượng re. Bây giờ, có thể dùng đối tượng re để tìm kiếm các mẫu theo yêu cầu.

Khi khởi tạo đối tượng, biểu thức quy tắc được dịch khi script được đánh giá. Nếu biểu thức quy tắc không thay đổi, thì sử dụng khởi tạo đối tượng sẽ hiệu quả.

– *Gọi hàm khởi tạo của đối tượng RegExp* : Java Script cung cấp đối tượng biểu thức định nghĩa trước đó là, RegExp. Một hàm khởi tạo được dùng để tạo một kiểu đối tượng và định nghĩa các thuộc tính của đối tượng. Ví dụ, chúng ta có thể tạo một đối tượng gọi là “employee”. Các thuộc tính của đối tượng là empID, join\_dt, salary.

```
function employee(empID, join_dt, salary)
{
    this.empID = empID
    this.join_dt = join_dt
    this.salary = salary
}
```

Sau khi hàm được tạo, phải dùng hàm để tạo một thẻ hiện của đối tượng bằng toán tử new. Ví dụ:

```
employee1=new employee("100", "11/11/02", 3000)
```

Khi chúng ta dùng hàm khởi tạo, biểu thức được dịch trong thời gian thực thi. Nếu biểu thức quy tắc thay đổi hoặc nếu nó phụ thuộc vào dữ liệu nhập vào từ người dùng thì sử dụng hàm khởi tạo là hợp lý nhất.

Ví dụ, hàm getdetails() tìm kiếm một mẫu dữ liệu được nhập từ người dùng. Dữ liệu nhập bởi người dùng rất đa dạng. Chọn mẫu tìm kiếm là (\w+) \s (\d\*). Nghĩa là, một hoặc nhiều ký tự bất kỳ xuất hiện sau một ký tự trắng hoặc xuất hiện bất kỳ một ký tự số nào. Dấu cộng (+) chỉ ra một hoặc nhiều ký tự xuất hiện, dấu sao (\*) chỉ ra 0 hoặc nhiều ký tự xuất hiện.

```
function getdetails()
{
    re = /(\w+)\s(\d+)/
    re.exec();
    window.alert(RegExp.$1 + ", your age is " + RegExp.$2);
}
```

#### > Sử dụng biểu thức quy tắc

Khi tạo ra một biểu thức quy tắc, chúng ta có thể đặt nó để dùng. Một số phương thức trong bảng dưới đây có thể dùng với biểu thức quy tắc.

Phương thức	Mô tả
Exec	Tìm kiếm một mẫu tương xứng trong một chuỗi. Nó trả về một mảng thông tin.
Test	Kiểm tra tương xứng trong một chuỗi. Trả về giá trị đúng hoặc sai.
Match	Tìm kiếm tương xứng trong một chuỗi. Trả về một mảng thông tin hoặc giá trị null nếu sai.
Search	Kiểm tra sự tương xứng trong một chuỗi. Trả về giá trị chỉ số của tương xứng nếu tồn tại, -1 nếu bị sai.
Replace	Tìm kiếm sự tương xứng trong một chuỗi và thay thế chuỗi con tìm kiếm tương xứng bằng một chuỗi con thay thế khác.
Split	Dùng để tách một chuỗi thành một mảng các chuỗi con.

Để dùng một phương thức, phải xác định đối tượng được sử dụng. Cú pháp là:

```
objectname.method = function_name
```

Sau đó có thể gọi phương thức trong ngữ cảnh của đối tượng. Cú pháp là:

objectname.methodname(parameters)

Chúng ta có thể dùng các cờ với biểu thức quy tắc. Hai cờ “g” và “i” được chọn tùy ý. Sau đó, chúng ta có thể dùng riêng hoặc dùng cả hai cờ. Cờ “g” được dùng để chỉ dẫn tìm kiếm toàn cục. Cờ “i” dùng để chỉ dẫn tìm kiếm có phân biệt chữ hoa và chữ thường. Ví dụ :

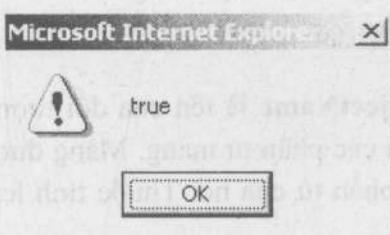
```
re = /\w+\s/g; //use a global search
```

### Ví dụ 13

Đoạn mã dưới đây dùng để kiểm tra phương thức tìm kiếm một mẫu trong chuỗi. Nếu mẫu được tìm thấy, giá trị trả về “True” và ngược lại thì trả về “false”.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Java Script">
re = /Time/
str = re.test ("Time and Tide wait for none");
window.alert(str);
</SCRIPT>
</HEAD>
</HTML>
```

Kết quả :



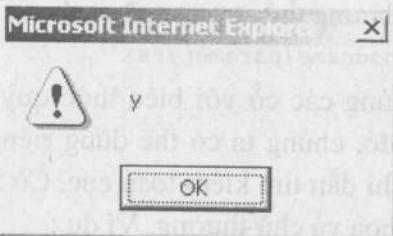
Hình 4.14. Java Script: Biểu thức

### Ví dụ 14

Đoạn mã sau đây tìm kiếm sự xuất hiện của ký tự x, y hoặc z.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Java Script">
re = /[xyz]/
str = re.exec("It is very coooooold");
window.alert(str);
</SCRIPT>
</HEAD>
</HTML>
```

Kết quả :



Hình 4.15. Java Script: Các biểu thức (1)

### 3.2. Mảng

Có những lúc ta muốn lưu nhiều giá trị vào trong một biến, khi đó phải sử dụng mảng. Mảng được dùng để lưu một tập hợp các biến có cùng tên. Chỉ số của mảng dùng để phân biệt các biến này. Trong Java Script chỉ số của mảng bắt đầu từ 0.

Tuy nhiên, Java Script không có kiểu dữ liệu mảng. Nhưng nó có một đối tượng mảng được xây dựng sẵn. Để dùng mảng trong chương trình, phải sử dụng đối tượng mảng và các phương thức của nó.

#### ➤ Tạo mảng

Cú pháp sau đây dùng để tạo một mảng:

```
arrayObjectName = new Array([element0, element1, ..., elementN])
```

Trong đó **arrayObjectName** là tên của đối tượng mảng, **elementN** là danh sách các giá trị của các phần tử mảng. Mảng được khởi tạo với các giá trị xác định như là các phần tử của nó. Thuộc tính **length** lưu số lượng các phần tử có trong mảng.

#### ➤ Thêm các phần tử

Chúng ta có thể đưa các phần tử vào một mảng ngay khi chúng ta tạo nó. Chúng ta cũng có thể đưa dữ liệu vào một mảng bằng cách gán giá trị cho các phần tử của nó. Ví dụ,

```
emp[0] = "Ryan Dias"  
emp[1] = "Graham Browne"  
emp[2] = "David Greene"
```

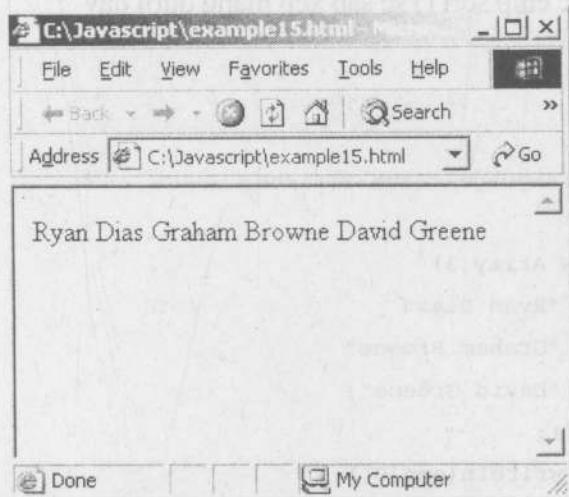
Ryan Dias	Graham Browne	David Greene
-----------	---------------	--------------

Đoạn mã trên tạo ra một mảng và gán giá trị cho từng phần tử của nó.

### Ví dụ 15

```
<html>
  <head>
    <script language="Java Script">
      <!--
        emp = new Array(3)
        emp[0] = "Ryan Dias"
        emp[1] = "Graham Browne"
        emp[2] = "David Greene";
        document.writeln(emp[0]);
        document.writeln(emp[1]);
        document.writeln(emp[2]);
      //-->
    </script>
  </head>
</html>
```

Kết quả :



Hình 4.16. Java Script: Mảng

#### ➤ Truy cập các phần tử mảng

Có hai cách để truy cập các phần tử của mảng. Chúng ta có thể:

- Chỉ ra tên của phần tử. Ví dụ, `emp ["Ryan Dias"]`
- Chỉ ra chỉ số của phần tử. Ví dụ, `emp [0]`.

## ➤ Sử dụng các phương thức của mảng

Dễ thao tác trên mảng, chúng ta phải dùng các phương thức của đối tượng mảng.

Phương thức	Mô tả
Join	Kết hợp các phần tử của mảng thành một chuỗi
Pop	Trả về phần tử cuối cùng của mảng, sau khi xoá nó từ mảng.
Push	Thêm một hoặc nhiều phần tử vào cuối mảng. Trả lại phần tử cuối cùng thêm vào.
Reverse	Đảo ngược các phần tử của mảng: phần tử đầu tiên của mảng trở thành phần tử cuối cùng và phần tử cuối cùng trở thành phần tử đầu tiên.
Shift	Xoá phần tử đầu tiên của mảng và trả về phần tử đó.
Sort	Sắp xếp các phần tử của mảng.

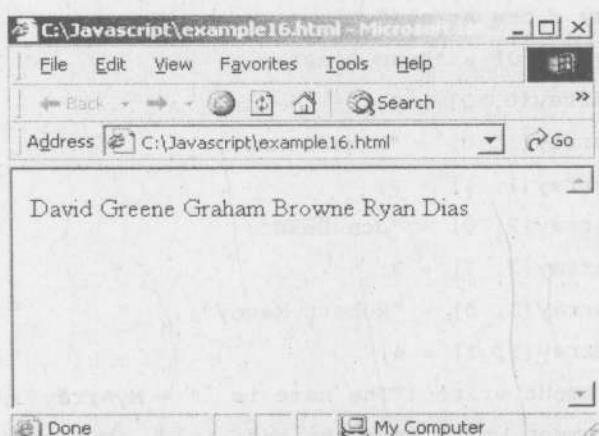
### • Phương thức sort

Phương thức emp.sort () sẽ sắp xếp mảng dưới đây

#### Ví dụ 16

```
<html>
  <head>
    <script language="Java Script">
      <!--
      emp = new Array(3)
      emp[0] = "Ryan Dias"
      emp[1] = "Graham Browne"
      emp[2] = "David Greene";
      emp.sort();
      document.writeln(emp[0]);
      document.writeln(emp[1]);
      document.writeln(emp[2]);
      //-->
    </script>
  </head>
</html>
```

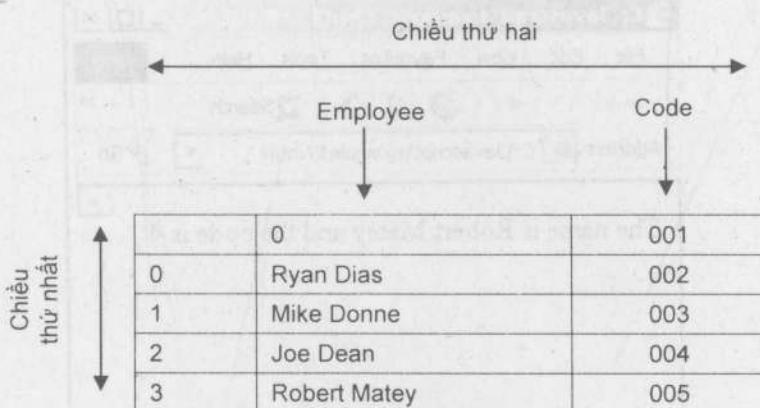
Kết quả :



Hình 4.17. Java Script: Sắp xếp các mảng (Sorting Arrays)

### > Mảng nhiều chiều

Một mảng có thể có nhiều hơn một chiều. Ví dụ, ta có thể tạo ra một mảng hai chiều để lưu trữ mã nhân viên và tên của nhân viên đó.



Hình 4.18. Mảng hai chiều

Đoạn mã dưới đây tạo ra một mảng hai chiều và cho hiển thị giá trị của một trong những phần tử trong mảng.

### Ví dụ 17

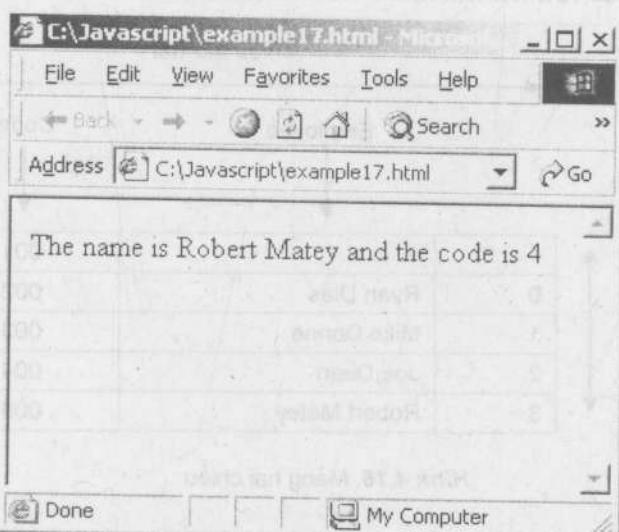
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Java Script">
<!--
```

```

MyArray = new Array(5,5);
MyArray[0, 0] = "Ryan Dias";
MyArray[0, 1] = 1;
MyArray[1, 0] = "Mike Donne";
MyArray[1, 1] = 2;
MyArray[2, 0] = "Joe Dean";
MyArray[2, 1] = 3;
MyArray[3, 0] = "Robert Matey";
MyArray[3, 1] = 4;
document.write ("The name is " + MyArray[3, 0]);
document.write("and the code is " + MyArray[3,1]);
-->
</SCRIPT>
</HEAD>
</HTML>

```

Kết quả :



Hình 4.19. Mảng hai chiều (Two dimensional Array)

### 3.3. Các câu lệnh điều kiện

Câu lệnh điều kiện được dùng để kiểm tra một điều kiện. Kết quả của điều kiện xác định câu lệnh hoặc khối lệnh sẽ được thực thi. Trong Java Script các câu lệnh sau đây được dùng để kiểm tra một điều kiện:

### **3.3.1. Câu lệnh if... else**

Câu lệnh này dùng để kiểm tra điều kiện, nó thực hiện việc tính toán trên một biểu thức. Nếu điều kiện là đúng (true) thì khối lệnh được thực thi.

```
if (condition)
{ statements; }
```

Ta cũng có thể chỉ ra khối lệnh cần thực hiện khi điều kiện là sai (false) bằng việc dùng mệnh đề else

```
if (condition)
{ statements; }
else
{ statements2; }
```

Nếu điều kiện là sai khối lệnh sau else được thực thi.

### **3.3.2. Lệnh switch**

Khi có nhiều tùy chọn If...else thì tốt nhất là sử dụng lệnh switch. Lệnh này còn được xem là lệnh case, lệnh switch thực thi một trong các khối lệnh tuỳ thuộc vào giá trị của biểu thức. Nếu không tìm thấy một giá trị nào trong danh sách các case của nó, khối lệnh trong phần default sẽ được thực hiện. Lệnh break dùng để thoát ra khỏi câu lệnh switch.

```
switch (expression) {
    case label:
        statement;
        break;
    case label:
        statement;
        break;
    ...
    default: statement;
}
```

Ví dụ :

```
switch (dayofweek) {
    case 1:
        statements;
        break;
    case 2:
        statements;
```

```

        break;
case 3:
    statements;
    break;
case 4:
    statements;
    break;
case 5:
    statements;
    break;
default: statements;
}

```

### **3.4. Các lệnh vòng lặp**

Các cấu trúc điều khiển việc thực hiện lặp đi lặp lại trong một chương trình được gọi là vòng lặp. Có nhiều loại vòng lặp:

- Vòng lặp thực hiện lặp đi lặp lại các lệnh cho đến khi điều kiện là False.
- Vòng lặp thực hiện lặp đi lặp lại các lệnh cho đến khi điều kiện là True.
- Vòng lặp thực hiện lặp đi lặp lại các lệnh theo một số lần nhất định.

#### **3.4.1. Vòng lặp “for”**

Vòng lặp for sẽ thực hiện lặp đi lặp lại khối lệnh cho đến khi điều kiện là false. Số lần thực hiện của vòng lặp thường được điều khiển thông qua một biến đếm.

Lệnh for bao gồm ba phần, cách nhau bởi dấu chấm phẩy. Cả ba phần đó đều không bắt buộc phải có và chúng điều khiển việc thực hiện của vòng lặp for.

```

for (lệnh khởi tạo; điều kiện; lệnh tăng)
{ statements; }

```

**Lệnh khởi tạo** - được thực hiện duy nhất một lần và thường dùng để khởi tạo biến đếm

**Lệnh điều kiện** - điều kiện của vòng lặp.

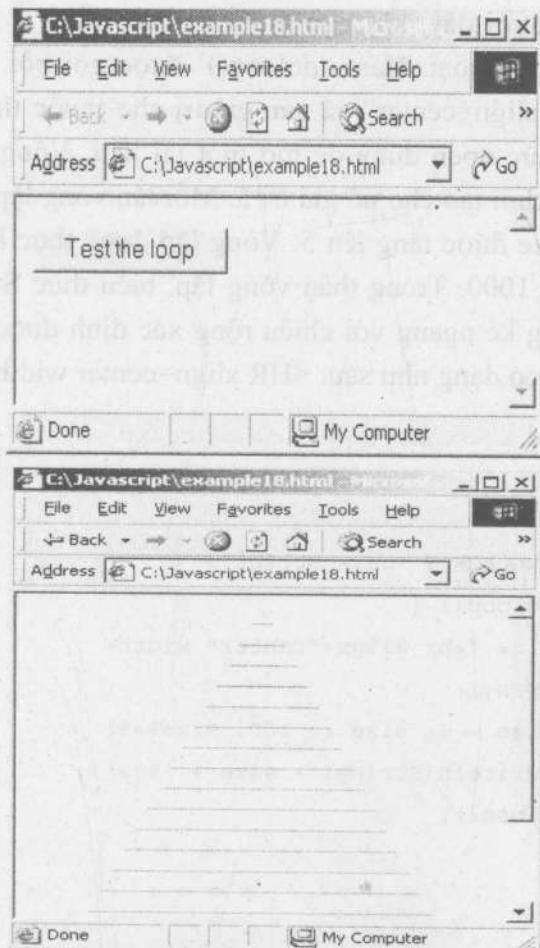
**Lệnh tăng** - thay đổi biến đếm trong vòng lặp.

Trong ví dụ dưới đây, khi người dùng nhấp chuột ở một nút, sự kiện onclick sẽ được kích hoạt. Hàm “doloop()” được gọi bởi sự kiện onclick. Hàm doloop đặt “align=center” và gán giá trị cho thuộc tính width của thẻ <HR>. Phương thức open dùng để mở một tài liệu. Vòng lặp for sử dụng biến đếm size và khởi tạo cho nó giá trị 5. Mỗi lần vòng lặp được thực hiện, giá trị của biến size được tăng lên 5. Vòng lặp được thực hiện đến khi size có giá trị lớn hơn 1000. Trong thân vòng lặp, biểu thức String1 được thực hiện và một đường kẻ ngang với chiều rộng xác định được in ra màn hình. Biểu thức String1 có dạng như sau: <HR align=center width=size%>

### Ví dụ 18

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Java Script">
function doloop() {
var String1 = '<hr align="center" width=';
document.open();
for (var size = 5; size <= 100; size+=5)
    document.writeln(String1 + size + "%>');
document.close();
}
</SCRIPT>
</HEAD>
<BODY>
<FORM>
<INPUT type="button"
       value="Test the loop"
       onclick = "doloop()" >
</FORM>
</BODY>
</HTML>
```

Kết quả



Hình 4.20. Vòng lặp for

### 3.4.2. Do...while

Vòng lặp Do...while được dùng để thực thi một khối lệnh cho đến khi điều kiện là false. Cú pháp là:

```
do  
{ statements;  
while (condition)
```

#### - Lệnh while

Lệnh while là một cấu trúc lặp khác trong Java Script. Nó được dùng để thực hiện một khối các câu lệnh khi điều kiện là true. Nếu có nhiều câu lệnh thực hiện trong thân của vòng lặp chương trình, phải sử dụng cặp dấu { và } để chứa các câu lệnh trong đó.

Khác biệt chính giữa vòng lặp while và do...while là các lệnh trong thân vòng lặp while có thể không được thực hiện một lần nào vì có thể ngay từ ban đầu điều kiện đã là false. Tuy nhiên vòng lặp do...while bao giờ cũng được thực hiện ít nhất một lần. Cú pháp là:

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

#### - Câu lệnh break & continue

Vòng lặp while loop và for sẽ kết thúc thực hiện khi điều kiện là false. Tuy nhiên ta cũng có thể kết thúc vòng lặp nếu muốn. Lệnh break dùng để kết thúc việc thực thi của một câu lệnh. Khi được sử dụng trong một vòng lặp, lệnh break làm dừng ngay vòng lặp đó và không thực hiện thêm nữa.

Một lệnh đặc biệt khác cũng có thể được sử dụng trong vòng lặp là lệnh continue. Continue dừng ngay lần lặp hiện tại và quay lại kiểm tra điều kiện để thực hiện lần lặp tiếp theo.

#### 3.4.3. Vòng lặp for...in

Câu lệnh for ... in được dùng để duyệt các thuộc tính của một đối tượng hay các phần tử của một mảng. Ví dụ, khi muốn thực hiện một khối các câu lệnh cho mỗi phần tử của mảng.

Cú pháp là:

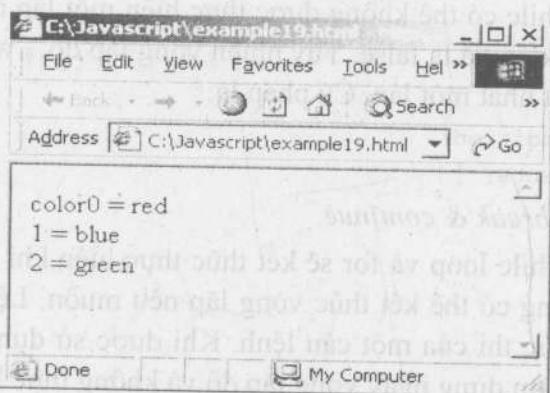
```
for (variable in object)
{ statements; }
```

Trong ví dụ dưới đây, một mảng "color" được tạo. Các phần tử của mảng là "red", "blue" và "green". Vòng lặp for .. in được dùng để duyệt qua mảng màu và hiển thị các phần tử trong nó.

#### Ví dụ 19

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Java Script">
color = new Array("red", "blue", "green");
var record = "color";
for (var prop in color)
(record += prop + " = " + color[prop] + "<BR>")
record += "<br>"
document.write(record)
</SCRIPT>
</HEAD>
</HTML>
```

Kết quả:



Hình 4.21. Vòng lặp For..in

### 3.5. Hàm (function)

Java Script có nhiều hàm định nghĩa trước mà chúng ta sẽ dùng trong script. Ví dụ, nếu muốn kiểm tra người dùng có phải nhập vào một số hay không, có thể sử dụng hàm isNaN (Not a Number) để thực hiện điều này. Trong phần này chúng ta sẽ xem xét một số hàm Java Script định nghĩa sẵn.

#### ➤ Hàm eval

Hàm eval được dùng để đánh giá một chuỗi và không cần tham chiếu đến bất kỳ một đối tượng cụ thể nào.

#### **eval(string)**

Chuỗi có thể là một biểu thức Java Script, câu lệnh, hoặc một nhóm các câu lệnh. Trong biểu thức có thể bao gồm các biến và thuộc tính của một đối tượng.

Nếu chuỗi được xác định là một biểu thức, hàm eval sẽ tính toán biểu thức. Nếu nó là một hoặc một tập các lệnh, nó sẽ thực thi các câu lệnh. Ví dụ, để tạo biểu thức theo mẫu như chuỗi sau - - "3 \* 5", có thể dùng eval để tính toán ra kết quả.

#### ➤ Hàm isNaN

Hàm isNaN được dùng để kiểm tra xem đối số truyền vào có phải là một số hay không.

`isNaN(testValue)`

Tạo hàm do người dùng định nghĩa

Một hàm Java Script khá giống với «thủ tục » ("procedure") hay « chương trình con » ("subroutine") trong ngôn ngữ lập trình. Một hàm là một tập các câu lệnh, thực hiện các tác vụ xác định. Chúng ta có thể truyền các giá trị (tham số) cho hàm. Hàm cũng có thể trả về một giá trị.

Các thành phần của một hàm là:

- Từ khoá function.
- Tên hàm.
- Danh sách các đối số của hàm cách nhau bởi dấu phẩy (,) đặt trong cặp ngoặc đơn( ).
- Các câu lệnh của hàm đặt trong cặp dấu mốc {}.

```
function funcName(argument1,argument2,etc)
{ statements; }
```

Một hàm có thể không có đối số, nhưng chúng ta vẫn phải có cặp dấu ngoặc đơn ( ).

```
funcName();
```

Các hàm không thể lồng nhau.

#### ➤ Định nghĩa hàm

Trước khi chúng ta gọi hàm, cần phải định nghĩa nó trong chương trình. Định nghĩa một hàm là một quá trình khai báo tên của hàm và các lệnh sẽ được thực thi khi gọi hàm.

#### ➤ Gọi một hàm

Để thực thi một hàm, ta phải gọi nó. Để gọi một hàm, phải chỉ ra tên hàm và danh sách các tham số nếu có.

#### Ví dụ 20

```
<html>
<head>
<script language="Java Script">
<!--
function hello( )
{
    document.write ('Hello.');
    document.write ('Welcome to the hello( ) function.');
    return;
}
```

```

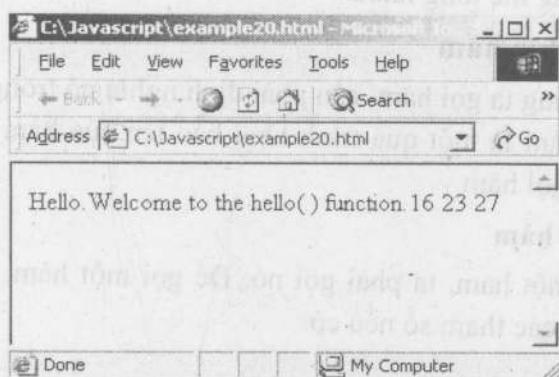
function sum_up (one, two)
{
    var result = one + two;
    return result;
}
function sum_all ( )
{
    var loop=0, sum=0;
    for (loop = arguments.length-1; loop >=0; loop--)
        sum += arguments[loop];
    return sum;
} // Add it up now

hello ( );

var total = sum_up(7, 9);
document.write (total + ' ' + sum_up (8, 15));
document.write (' ' + sum_all (1, 5, 8, 7, 6));
//-->
</script>
</head> </html>

```

Kết quả:



Hình 4.22. Định nghĩa và gọi một hàm

### Câu lệnh return

Câu lệnh này được dùng để trả về một giá trị. Dùng lệnh return trong một hàm là không bắt buộc vì không phải tất cả các hàm đều trả về một giá trị cụ thể.

Cú pháp là:

```

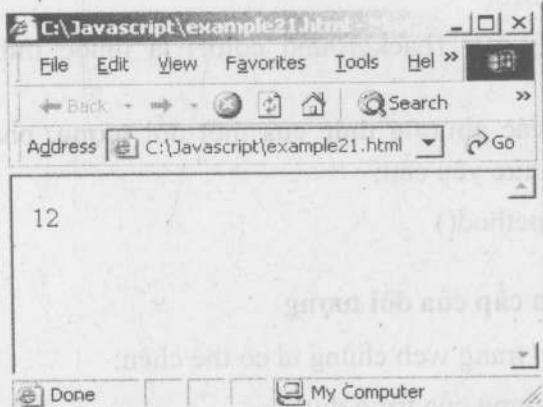
return value;
Or
return ( value ) ;

```

## Ví dụ 21

```
<html>
  <head>
    <script language="Java Script">
      function testreturn(x) {
        var i = 0;
        while (i < 6) {
          if (i == 3)
            break;
          i++;
        }
        document.write (i*x);
        return (i*x);
      }
    </script>
  </head>
<BODY>
  <SCRIPT>
    testreturn(4);
  </SCRIPT>
</BODY>
</html>
```

Kết quả :



Hình 4.23. Giá trị trả về hàm

Tại bất kỳ lúc nào, cũng có thể thoát ra khỏi hàm đơn giản, chỉ cần sử dụng lệnh return mà không cần đến bất kỳ cấu trúc phức tạp nào. Quyền điều khiển ngay lập tức trả về cho câu lệnh đứng sau lệnh gọi hàm.

## 4. CÁC ĐỐI TƯỢNG CƠ BẢN TRONG Java Script

Khi tạo ra một trang web, có thể sử dụng các đối tượng do trình duyệt, ngôn ngữ kịch bản và các thành phần HTML, cung cấp. Trong chương này, chúng ta cùng thảo luận về những đối tượng khác nhau có thể được chèn vào trong một tài liệu HTML.

### 4.1. Các đối tượng Java Script

Một đối tượng là một gói dữ liệu toàn diện. Các thuộc tính (biến) dùng để định nghĩa đối tượng và các phương thức (hàm) tác động tới dữ liệu đều nằm trong đối tượng. Ví dụ, một chiếc xe hơi là một đối tượng. Các thuộc tính của chiếc xe hơi là cấu tạo, kiểu dáng và màu sắc của nó. Hầu hết các chiếc xe hơi đều có một vài phương thức chung như go(), brake(), reverse().

```
carobj.make = "Fiat"  
carobj.model = "Uno"  
carobj.color = "red"
```

Để truy cập các thuộc tính của đối tượng, chúng ta phải chỉ ra tên đối tượng và thuộc tính của nó:

```
objectName.propertyName
```

Ví dụ:

```
Document.bgcolor
```

Trong đó: bgcolor (background color) là thuộc tính của đối tượng document.

Để truy cập các phương thức của một đối tượng, phải chỉ ra tên đối tượng và phương thức yêu cầu:

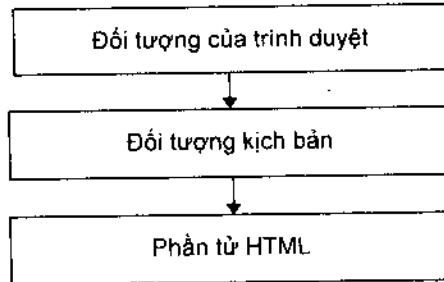
```
objectName.method()
```

### 4.2. Cây phân cấp của đối tượng

Khi tạo ra một trang web chúng ta có thể chèn:

- Các đối tượng của trình duyệt
- Các đối tượng có sẵn (thay đổi phụ thuộc vào ngôn ngữ kịch bản được sử dụng).
- Các phần tử HTML

Đĩ nhiên, có thể tạo ra các đối tượng để sử dụng theo yêu cầu của mình.



Hình 4.24. Cây phân cấp đối tượng

Khi tài liệu HTML được hiển thị trong trình duyệt, một cây phân cấp đối tượng được tạo ra dựa trên các phần tử trong trang. Các đối tượng trình duyệt ví dụ như văn bản, cửa sổ, khung, vị trí,... nằm trên cùng của cây phân cấp đối tượng. Sau đó là các đối tượng Java Script. Đây là các đối tượng được cung cấp bởi Java Script ví dụ như đối tượng Date, Array,... Các phần tử HTML nằm ở sau cùng và chính là các thẻ HTML tạo nên văn bản hiện hành.

Sự hiểu biết về cây phân cấp này rất quan trọng bởi vì các đối tượng được truy cập theo sự phân cấp. Ví dụ, để truy cập một đối tượng form bạn cần phải chỉ ra tên form và đối tượng chứa trong đó như văn bản, câu lệnh sẽ là:

```
document.form1
```

Để truy cập các thuộc tính của một phần tử văn bản, Text1 trong một đối tượng form:

```
document.form.text1.value = "Having fun"
```

### 4.3. Câu lệnh this

Câu lệnh ‘this’ không chỉ là một thuộc tính nội tại. Giá trị của nó chỉ ra đối tượng hiện hành và có thể có các thuộc tính chuẩn như tên, độ dài, và giá trị được áp dụng phù hợp. Câu lệnh ‘this’ chỉ được dùng trong phạm vi của một hàm hay các tham chiếu khi gọi hàm.

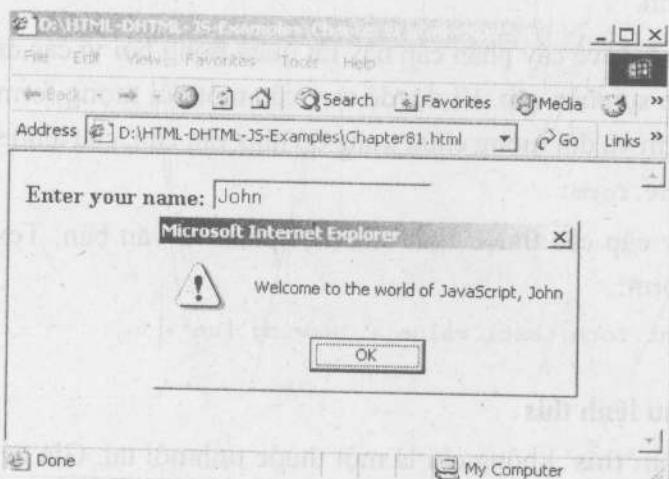
```
this [.property]
```

Nếu nó không có đối số thì nó sẽ thông qua đối tượng hiện hành. Tuy nhiên, chúng ta nên gán vào một thuộc tính hợp lệ để đưa ra kết quả.

### Ví dụ 22

```
<HTML>
<HEAD>
    <script language="Java Script">
        function dispname(name) {
            alert("Welcome to the world of Java Script, " + name);
        }
    </script>
</HEAD>
<FORM>
    <B>Enter your name:</B>
    <INPUT TYPE = "text" NAME = "text1" SIZE = 20
    onChange="dispname(this.form.text1.value)">
</FORM>
</HTML>
```

Kết quả



Hình 4.25. Kết quả của ví dụ 22

### Ví dụ 23

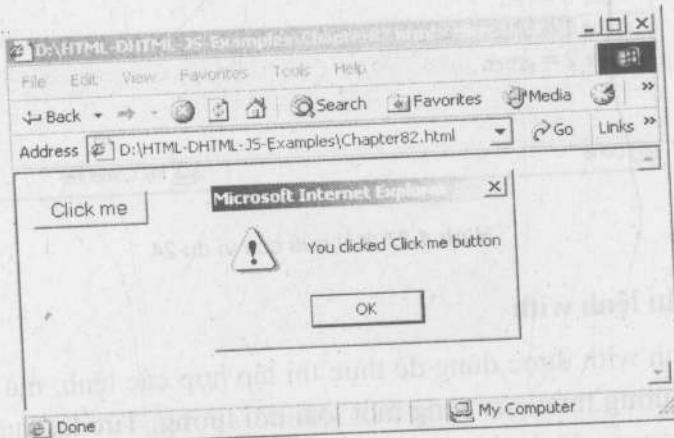
```
<html>
<head>
    <script language="Java Script">
```

```

function show(value) {
    alert("You clicked " + value + " button");
}
</script>
</head>
<FORM>
<B>Enter your name:</B>
<INPUT TYPE = "button" NAME = "button1" value = "Click me" SIZE = 20 onClick="show(this.form.button1.value)">
</FORM>
</html>

```

Kết quả



Hình 4.26. Kết quả của ví dụ 23

#### 4.4. Câu lệnh for...in

Câu lệnh for...in được dùng để lặp mỗi thuộc tính của đối tượng hoặc mỗi phần tử của một mảng. Ví dụ, chúng ta muốn thực thi một khối các câu lệnh cho mỗi phần tử trong một mảng.

```

for (variable in object)
{
    statements;
}

```

#### Ví dụ 24

```

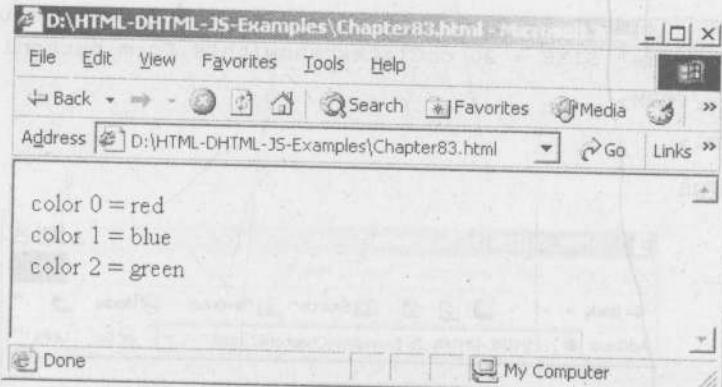
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Java Script">
color = new Array("red", "blue", "green");

```

```

        for (var prop in color) {
            var record = "color ";
            record += prop + " = " + color[prop] + "<BR>";
            document.write(record)
        }
    </SCRIPT>
</HEAD>
</HTML>

```



Hình 4.27. Kết quả của ví dụ 24

#### 4.5. Câu lệnh with

Câu lệnh with được dùng để thực thi tập hợp các lệnh, mà các lệnh này dùng các phương thức của cùng một loại đối tượng. Tức là, thuộc tính được gán cho đối tượng đã được xác định trong câu lệnh with.

```

with (object)
{
    statements;
}

```

Trong ví dụ dưới đây, đối tượng Math được xem là đối tượng mặc định.

#### Ví dụ 25

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE = "Java Script">
        var a, b, c;
        var r=10;
        with (Math) {
            a = PI * r * r;
            b = r * cos(PI);
            c = r * sin(PI/2);
        }
    </SCRIPT>
</HEAD>

```

```

        document.write (a + "<BR>");  

        document.write (b + "<BR>");  

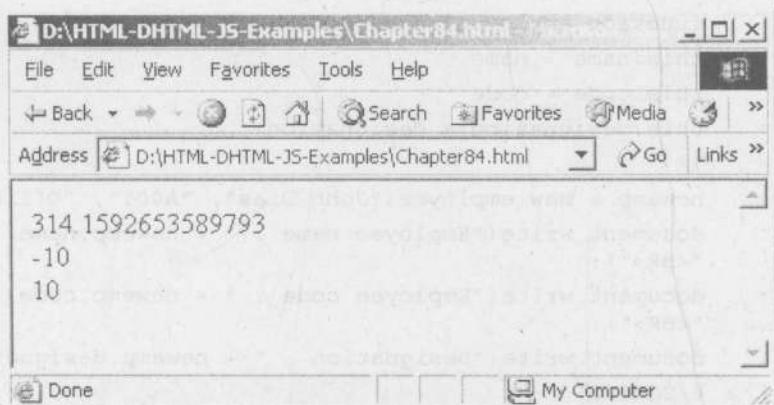
        document.write (c + "<BR>");  

    </SCRIPT>  

</HEAD>  

</HTML>

```



Hình 4.28. Kết quả của ví dụ 25

#### 4.6. Câu lệnh new

Toán tử new được dùng để tạo ra một thực thể mới của một loại đối tượng. Loại đối tượng này có thể là đối tượng sẵn có hoặc do người dùng định nghĩa. Ví dụ như đối tượng mảng.

```
objectName = new objectType (param1 [,param2] ... [,paramN])
```

Trong đó:

- objectName là tên của thực thể đối tượng mới.
- ObjectType là một hàm quyết định loại của đối tượng. Ví dụ Array.
- Param[1, 2, ... ] là các giá trị thuộc tính của đối tượng.

Hãy xem ví dụ dưới đây, chúng ta có thể tạo một loại đối tượng có tên gọi là employee, các thuộc tính là name, code, designation.

```

function employee (name, code, designation) {  

    this.name = name  

    this.code = code  

    this.designation = designation  

}

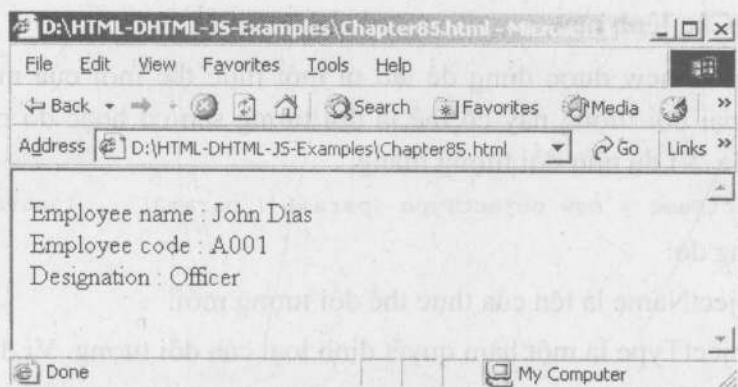
```

Để tạo ra một thực thể mới của đối tượng employee, chúng ta dùng:

```
newemp = new employee("John Dias", "A001", "Officer")
```

### Ví dụ 26

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="Java Script">
      function employee(name, code, designation) {
        this.name = name
        this.code = code
        this.designation = designation
      }
      newemp = new employee("John Dias", "A001", "Officer");
      document.write("Employee name : " + newemp.name +
      "<BR>");
      document.write("Employee code : " + newemp.code +
      "<BR>");
      document.write("Designation : " + newemp.designation);
    </SCRIPT>
  </HEAD>
</HTML>
```



Hình 4.29. Kết quả của ví dụ 26

## 4.7. Câu lệnh eval

Hàm eval được dùng để đánh giá một chuỗi mã lệnh mà không cần tham chiếu đến bối cảnh đối tượng cụ thể nào.

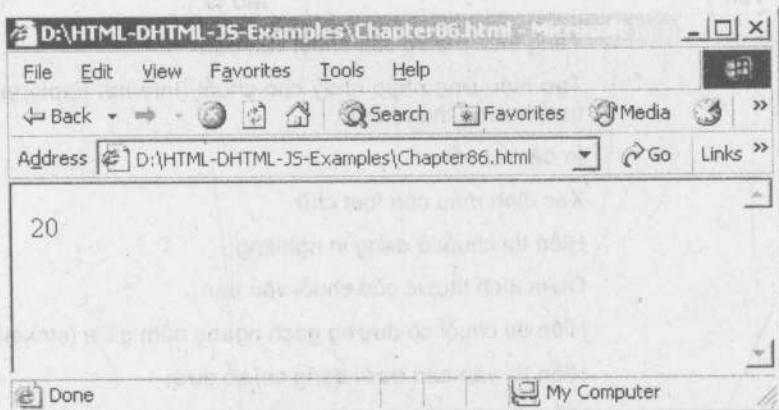
```
eval(string)
```

Chuỗi có thể là một biểu thức Java Script, một câu lệnh hoặc một nhóm câu lệnh. Biểu thức có thể bao gồm nhiều biến và nhiều thuộc tính của một đối tượng.

Nếu chuỗi trên là một biểu thức thì hàm eval định giá trị biểu thức đó. Nếu chuỗi đó là một hoặc nhiều câu lệnh thì hàm eval sẽ thực thi các câu lệnh. Chúng ta có thể dùng hàm eval để đưa ra kết quả.

### Ví dụ 27

```
<HTML>
<SCRIPT LANGUAGE = "Java Script">
    var x = 5;
    var z = 10;
    document.write(eval("x + z + 5"));
</SCRIPT>
</HTML>
```



Hình 4.30. Kết quả của ví dụ 27

## 4.8. Đối tượng String

Đối tượng string được dùng để thao tác và làm việc với chuỗi văn bản. Chúng ta có thể tách nó ra thành các chuỗi con và biến đổi chuỗi đó thành các chuỗi hoa hoặc thường trong một chương trình. Cú pháp tổng quát là:

stringName.propertyName

Hoặc

stringName.methodName

Có 3 phương thức khác nhau để tạo ra chuỗi.

– Dùng lệnh var và gán cho nó một giá trị. Ví dụ :

var newstr = " This is my script"

– Dùng một toán tử (=) có gán với một tên biến. Ví dụ :

newstr = "This is my script"

– Dùng hàm khởi tạo String (string). Ví dụ :

var newstr = String ("This is my script")

#### 4.8.1. Các thuộc tính

Tên	Mô tả
length	Trả lại độ dài của chuỗi

#### 4.8.2. Phương thức

Tên	Mô tả
Big	Tăng kích thước của chuỗi văn bản
Blink	Tạo hiệu ứng nhấp nháy cho chuỗi (Internet Explorer không hỗ trợ phương thức này)
Bold	In đậm chuỗi
Fontcolor	Xác định màu của font chữ
Italics	Hiển thị chuỗi ở dạng in nghiêng
Small	Giảm kích thước của chuỗi văn bản
Strike	Hiển thị chuỗi có đường gạch ngang nằm giữa ( <del>strikethrough</del> )
Sub	Hiển thị văn bản dưới dạng chỉ số dưới
Sup	Hiển thị văn bản dưới dạng chỉ số trên
ToLowerCase	Chuyển chuỗi thành ký tự thường
ToUpperCase	Chuyển chuỗi thành ký tự hoa

Ví dụ dưới đây hiển thị một vài phương thức và công dụng của chúng:

#### Ví dụ 28

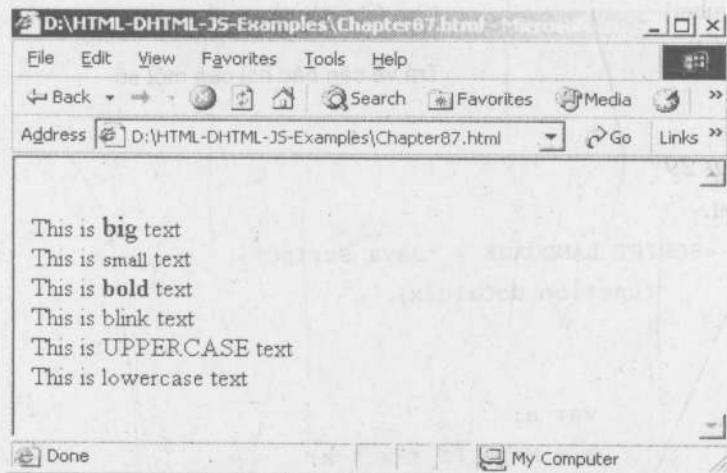
<HTML>

```
<HEAD>
<script language = "Java Script">
var bstr = "big";
var sstr = "small";
var blstr = "bold";
var blkstr = "blink"
```

```

var ucase = "Uppercase";
var lcase = "Lowercase";
document.write ("<BR>This is "+ bstr.big() + " text");
document.write ("<BR>This is "+ sstr.small() +" text");
document.write ("<BR>This is "+ blstr.bold() + " text");
document.write ("<BR>This is "+ blkstr.blink() + " text");
document.write ("<BR>This is "+ ucase.toUpperCase() + " text");
document.write ("<BR>This is "+ lcase.toLowerCase() + " text");
</script>
</HEAD>
</HTML>

```



Hình 4.31. Kết quả của ví dụ 28

## 4.9. Đối tượng Math

Đối tượng Math có các thuộc tính và phương thức biểu thị các phép tính toán học nâng cao.

### 4.9.1. Thuộc tính

Thuộc tính	Mô tả
PI	Giá trị của $\pi$ , bằng khoảng 3,1415.
LN10	Giá trị của $\lg$ (logarit cơ số 10), bằng khoảng 2,302
E	Giá trị của hằng số Euler, bằng khoảng 2,718. Hằng số Euler được dùng như số cơ sở cho các ln

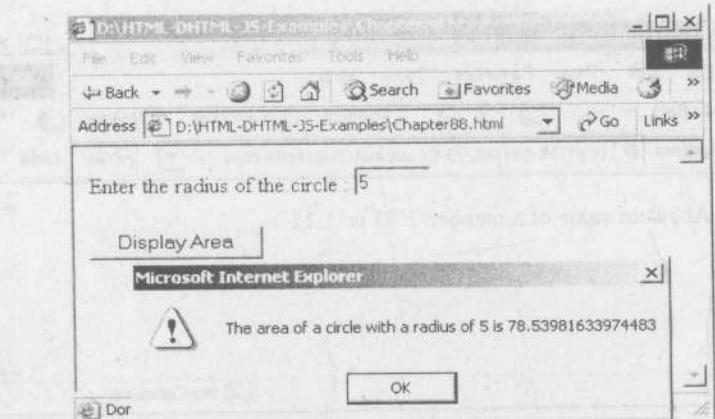
#### 4.9.2. Phương thức

Phương thức	Mô tả
Abs (number)	Trả về giá trị tuyệt đối của một số
Sin (number)	Trả về giá trị sin của của một số (tính bằng radian)
Cos (number)	Trả về giá trị cosin của của một số (tính bằng radian)
Tan (number)	Trả về giá trị tang của của một số (tính bằng radian)
Min (number1, number2)	Trả về giá trị nhỏ nhất của hai số number1 và number2
Max (number1, number2)	Trả về giá trị lớn nhất của hai số number1 và number2
Round (number)	Làm tròn đối số tới số nguyên gần nhất
Sqrt (number)	Trả về căn bậc hai của một số

#### Ví dụ 29

<HTML>

```
<SCRIPT LANGUAGE = "Java Script">
    function doCalc(x)
    {
        var a;
        a = Math.PI * x * x;
        alert ("The area of a circle with a radius of " + x
        + " " + "is" + " " + a);
    }
</SCRIPT>
<BODY bgColor = white>
<FORM>
    Enter the radius of the circle :
    <INPUT TYPE = TEXT size = 5 name = "rad">
    <BR><BR>
    <INPUT type = button value = "Display Area"
    onclick="doCalc(rad.value)">
</FORM>
</BODY>
</HTML>
```



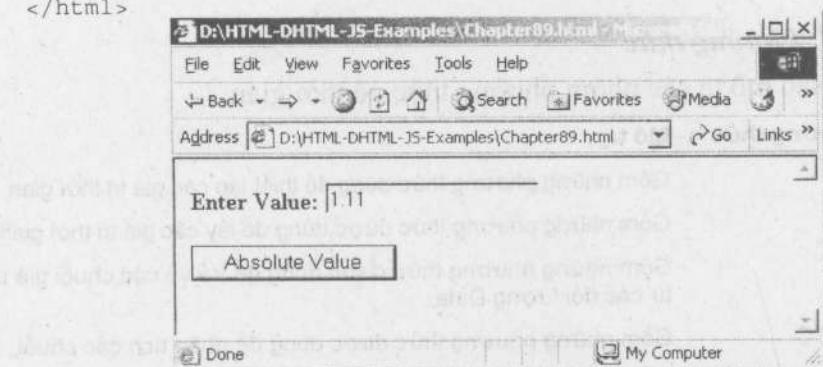
Hình 4.32. Kết quả của ví dụ 29

### Ví dụ 30

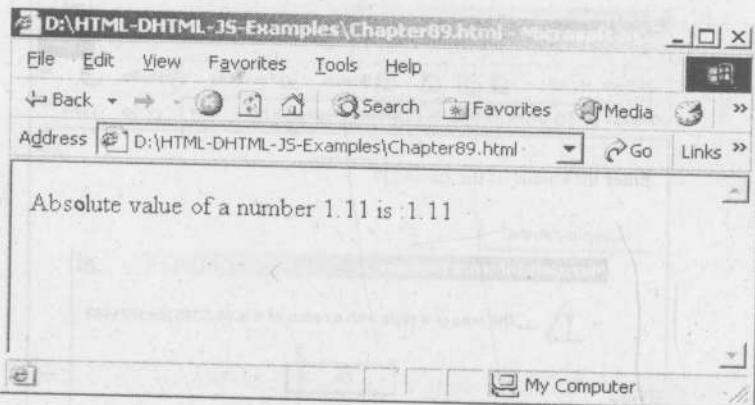
```

<html>
    <head>
        <script language="Java Script">
            function show(value)
            {
                with(Math)
                    document.write("Absolute value of a number "+
                        value+ " is :" +abs(eval(value)));
            }
        </script>
    </head>
    <FORM>
        <B>Enter Value:</B>
        <INPUT TYPE = "text" NAME = "text1" SIZE = 20>
        <P>
        <INPUT TYPE = "button" NAME = "button1" value="Absolute
        Value" onClick="show(this.form.text1.value)"
    </FORM>
</html>

```



Hình 4.33a. Kết quả của ví dụ 30



Hình 4.33b. Kết quả của ví dụ 30(sau khi nhấn nút lệnh)

## 4.10. Đối tượng Date

Date là một đối tượng có sẵn chứa thông tin về ngày và giờ. Đối tượng Date không có thuộc tính nào. Tuy nhiên, nó có nhiều phương thức dùng để thiết lập, lấy và xử lý các thông tin về thời gian.

Đối tượng Date lưu trữ thời gian theo số mili giây tính từ 1/1/1970 00:00:00.

Tạo một đối tượng Date

```
DateObject = new Date (parameters)
```

Trong đó:

- DateObject là một biến lưu trữ một đối tượng thời gian mới.
- Parameters có thể là:
  - Không có gì – Nếu không có tham số, nó sẽ trả về thời gian hiện tại của hệ thống. Ví dụ: today = new Date () .
  - String (chuỗi) – biểu diễn ngày giờ theo dạng: “MM DD, YYYY, hh:mm:ss”. Ví dụ : tdate = new Date (“ July 29, 1998,10:30:00”)

### 4.10.1. Phương thức

Bảng sau mô tả các nhóm phương thức về thời gian:

Nhóm phương thức	Mô tả
Set	Gồm những phương thức dùng để thiết lập các giá trị thời gian.
Get	Gồm những phương thức được dùng để lấy các giá trị thời gian.
To	Gồm những phương thức được dùng để trả về các chuỗi giá trị từ các đối tượng Date.
parse và UTC	Gồm những phương thức được dùng để phân tích các chuỗi.

Bảng sau dùng các số nguyên để biểu diễn các giá trị:

Các giá trị	Số nguyên
Giây và phút	0 đến 59
Giờ	0 đến 23
Ngày (trong tuần)	0 đến 6
Ngày (trong tháng)	1 đến 31
Tháng	0 đến 11 (tháng một đến tháng mười hai)
Năm	Từ 1900 trở đi

Chúng ta hãy tìm hiểu các nhóm phương thức get, set, to và parse:

### ➤ Nhóm phương thức get

Phương thức	Mô tả
getDate	Trả về ngày trong tháng từ đối tượng Date (1-31)
getDay	Trả về ngày trong tuần từ đối tượng Date (0-6)
getHours	Trả về giờ từ đối tượng Date (0 – 23)
getMinutes	Trả về phút từ đối tượng Date ( 0-59)
getSeconds	Trả về giây từ đối tượng Date (0 -59)
getMonth	Trả về tháng từ đối tượng Date (0 – 11)
getYear	Trả về năm từ đối tượng Date (năm (-)1900)
getTime	Trả về số mili giây của thời gian hiện tại (tính từ 1/1/1970)
getTimeZoneOffset	Trả về chênh lệch bằng phút giữa giờ địa phương và giờ chuẩn (GMT)

### ➤ Nhóm phương thức set

Phương thức	Mô tả
setDate	Thiết lập ngày trong tháng cho đối tượng Date (0 –31)
setHours	Thiết lập giờ cho đối tượng Date (0-23)
setMinutes	Thiết lập phút cho đối tượng Date (0-59)
setSeconds	Thiết lập giây cho đối tượng Date (0-59)
setTime	Thiết lập giá trị thời gian (tính bằng mili giây) cho đối tượng Date
setMonth	Thiết lập tháng cho đối tượng Date (1-12)
setYear	Thiết lập năm cho đối tượng Date, năm phải lớn hơn 1900 (năm (-) 1900)

## ➤ Nhóm phương thức to

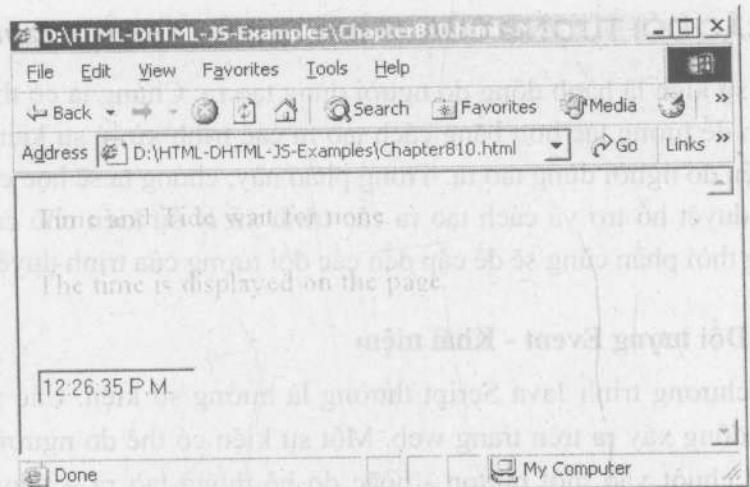
Phương thức	Mô tả
toGMTString	Chuyển một đối tượng Date từ một chuỗi thời gian sang dạng GMT
toLocaleString	Chuyển một đối tượng Date từ một chuỗi sang dạng thời gian địa phương

## ➤ Nhóm phương thức Parse & UTC

Phương thức	Mô tả
Date.parse(date string )	Số mili giây trong một date string (chuỗi thời gian) tính từ 1/1/1970
Date.UTC (year, month, day, hours, min., secs.)	Số mili giây của một đối tượng thời gian tính từ 1/1/1970

### Ví dụ 31

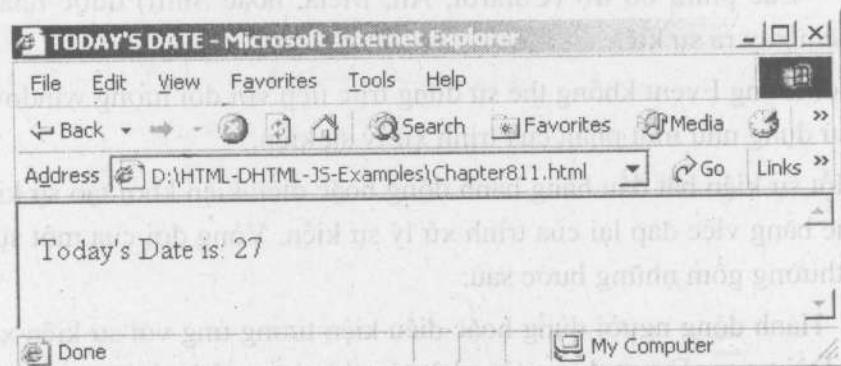
```
<HTML>
<HEAD>
<SCRIPT language="Java Script">
<!--
function disptime() {
    var time = new Date()
    var hour = time.getHours()
    var minute = time.getMinutes()
    var second = time.getSeconds()
    var temp = "" + ((hour > 12) ? hour - 12 : hour)
    temp += ((minute < 10) ? ":0" : ":") + minute
    temp += ((second < 10) ? ":0" : ":") + second
    temp += (hour >= 12) ? " P.M." : " A.M."
    document.MyPage.digits.value = temp
    id = setTimeout("disptime()",1000)
}
//-->
</SCRIPT>
</HEAD>
<BODY onLoad="disptime()">
<P><font color = hotpink size = 4>Time and Tide wait for none.
<BR><BR>
<P> The time is displayed on the page.
<BR><BR><BR>
<FORM NAME="MyPage">
<INPUT TYPE="text" NAME="digits" SIZE=12 VALUE="">
</FORM>
</BODY>
</HTML>
```



Hình 4.34. Kết quả của ví dụ 31

### Ví dụ 32

```
<HTML>
  <HEAD>
    <TITLE> TODAY'S DATE </TITLE>
  </HEAD>
  <BODY>
    <script language="Java Script">
      mydate=new Date()
      document.write("Today's Date is: "+ mydate.getDate());
    </script>
  </BODY>
</HTML>
```



Hình 4.35. Kết quả của ví dụ 32

## 5. CÁC ĐỐI TƯỢNG CỦA TRÌNH DUYỆT TRONG Java Script

Các sự kiện là hành động do người dùng tạo ra. Chúng ta có thể làm cho trang web dễ tương tác hơn bằng cách tạo ra các trình xử lý sự kiện, đáp ứng các sự kiện do người dùng tạo ra. Trong phần này, chúng ta sẽ học các sự kiện mà trình duyệt hỗ trợ và cách tạo ra các trình xử lý sự kiện cho các sự kiện này, đồng thời phần cũng sẽ đề cập đến các đối tượng của trình duyệt.

### 5.1. Đối tượng Event - Khái niệm

Các chương trình Java Script thường là hướng sự kiện. Các sự kiện là các hành động xảy ra trên trang web. Một sự kiện có thể do người dùng tạo ra – click chuột vào một button – hoặc do hệ thống tạo ra – thay đổi kích thước của trang.

Hầu hết các trình duyệt đều hỗ trợ một đối tượng Event. Mỗi sự kiện có một đối tượng Event tương ứng. Đối tượng Event cung cấp thông tin về sự kiện - loại sự kiện và vị trí của con trỏ tại thời điểm xảy ra sự kiện. Khi một sự kiện được phát sinh, nó được gửi đi như một đối số đến trình xử lý sự kiện. Dĩ nhiên, phải có một trình xử lý sự kiện trong trường hợp này.

Ví dụ, khi người dùng nhấp chuột, sự kiện **onmousedown** được phát sinh. Đối tượng Event chứa những thông tin sau:

- ✓ Loại sự kiện - Trong trường hợp này là nhấp chuột
- ✓ Vị trí x và y của con trỏ khi nhấp chuột
- ✓ Nút chuột nào được nhấn
- ✓ Các phím hỗ trợ (Control, Alt, Meta, hoặc Shift) được nhấn vào thời điểm xảy ra sự kiện.

Đối tượng Event không thể sử dụng trực tiếp với đối tượng window. Nó được sử dụng như một phần của trình xử lý sự kiện.

Một sự kiện bắt đầu bằng hành động hoặc điều kiện khởi tạo sự kiện và kết thúc bằng việc đáp lại của trình xử lý sự kiện. Vòng đời của một sự kiện thông thường gồm những bước sau:

1. Hành động người dùng hoặc điều kiện tương ứng với sự kiện xảy ra.
2. Đối tượng Event được cập nhật tức thì nhằm phản ánh trạng thái của sự kiện.

3. Sự kiện được kích hoạt.
4. Trình xử lý sự kiện tương ứng được gọi.
5. Trình xử lý sự kiện thực hiện hành động của nó và trả về điều khiển cho chương trình.

## 5.2. Các sự kiện Java Script

Tập hợp các sự kiện tương ứng với các phần tử khác nhau trên trang là một phần của mô hình đối tượng tài liệu (Document Object Model), không phải của Java Script. Nghĩa là, các sự kiện được một phần tử nào đó hỗ trợ có thể khác nhau trên các trình duyệt.

Sau đây là một số sự kiện thường được hầu hết các đối tượng hỗ trợ:

### ➤ **onClick**

Sự kiện onClick được tạo ra bất cứ khi nào người dùng nhấp chuột lên các phần tử form nào đó (button, checkbox, radio button và phần tử select), hoặc lên các hyperlink.

#### *Ví dụ 33*

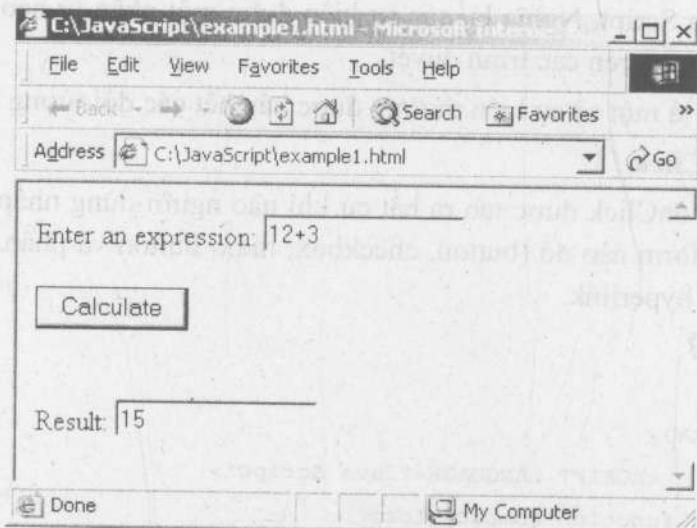
```
<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="Java Script">
            function compute(form)
            {
                if (confirm("Are you sure?"))
                    form.result.value = eval(form.expr.value)
                else
                    alert("Please come back again.")
            }
        </SCRIPT>
    </HEAD>
    <BODY>
        <FORM>
            Enter an expression:
            <INPUT TYPE="text" NAME="expr" SIZE=15 ><BR><BR>
            <INPUT TYPE="button" VALUE="Calculate"
            ONCLICK="compute(this.form)">
        </FORM>
    </BODY>
</HTML>
```

```

<BR><BR><BR>
Result
<INPUT TYPE="text" NAME="result" SIZE=15 >
<BR>
</FORM>
</BODY>
</HTML>

```

Kết quả đoạn mã trong ví dụ 33 được minh họa ở hình 4.36.



Hình 4.36. Kết quả của ví dụ 33

#### ➤ **onChange**

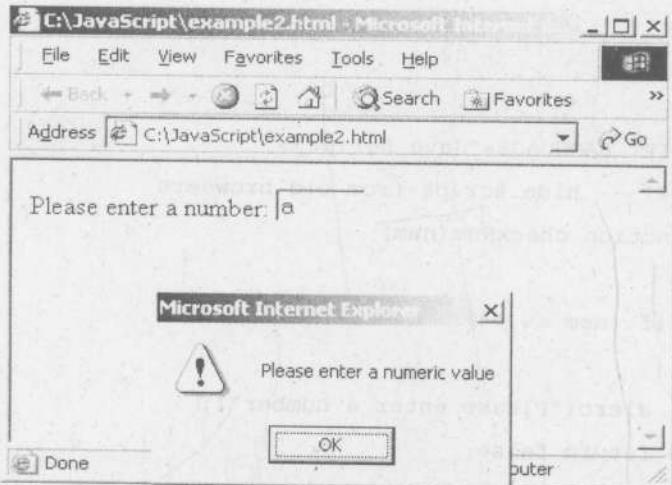
Sự kiện onChange xảy ra bất cứ khi nào một phần tử form thay đổi. Điều này có thể xảy ra khi nội dung của một trường văn bản thay đổi, hoặc khi một chọn lựa trong danh sách chọn lựa thay đổi. Tuy nhiên, sự kiện onChange không được tạo ra khi một radio button hoặc một checkbox được nhấp. Thay vào đó, sự kiện onClick sẽ được tạo ra.

Sự kiện onChange được gửi đi khi một phần tử hoàn tất việc thay đổi. Vì vậy, khi một textbox đang được hiệu chỉnh, sự kiện onChange chỉ được phát sinh sau khi việc hiệu chỉnh đã hoàn tất, và khi người dùng thoát khỏi textbox đó.

### *Ví dụ 34*

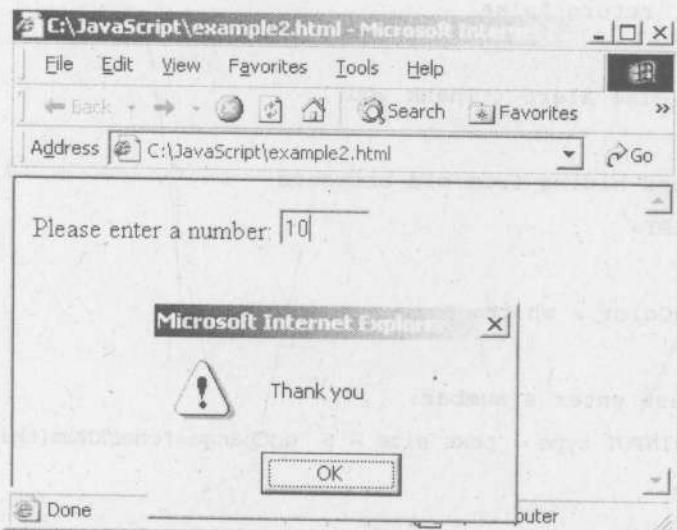
```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Java Script">
    <!-- hide script from old browsers
    function checkNum(num)
    {
        if (num == "")
        {
            alert("Please enter a number");
            return false;
        }
        if (isNaN (num))
        {
            alert("Please enter a numeric value");
            return false;
        }
        else alert ("Thank you");
    }
    // end hiding from old browsers -->
</SCRIPT>
</HEAD>
<BODY bgColor = white>
<FORM>
    Please enter a number:
    <INPUT type = text size = 5 onChange="checkNum(this.value)">
</FORM>
</BODY>
</HTML>
```

Hình 4.37 và 4.38 minh họa kết quả của đoạn mã trong ví dụ 34.



Hình 4.37. Kết quả của ví dụ 34

Nếu chúng ta nhập vào một giá trị số:



Hình 4.38. Kết quả của ví dụ 34

#### ➤ onFocus

Sự kiện onFocus được gửi đi bắt cứ khi nào một phần tử form trở thành phần tử hiện thời. Chỉ khi một phần tử nhận được focus nó mới nhận được input từ người dùng. Điều này có thể xảy ra khi người dùng nhấp chuột lên

phần tử, hoặc sử dụng phím Tab hoặc Shift+Tab (di chuyển tới các phần tử trên form).

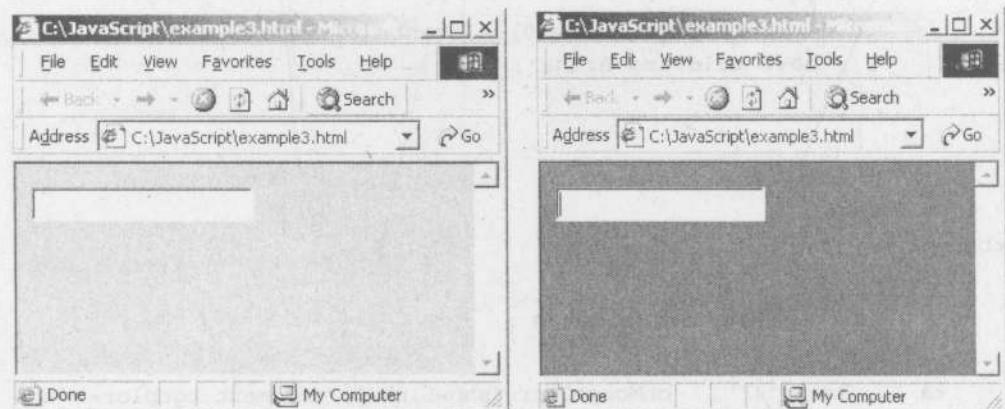
#### ➤ **onBlur**

Blur ngược với focus. Khi người dùng rời khỏi một phần tử trên form, sự kiện onBlur được kích hoạt. Đối với một số phần tử, nếu nội dung của nó cũng bị thay đổi, thì sự kiện onChange cũng được kích hoạt.

#### **Ví dụ 35**

```
<HTML>
  <BODY BGCOLOR="lavender">
    <FORM>
      <INPUT type = text name = text1
        onBlur= " (document.bgColor='aqua') "
        onFocus= " (document.bgColor='dimgray') " >
    </FORM>
  </BODY>
</HTML>
```

Khi textbox nhận được focus, màu nền sẽ chuyển sang DIMGRAY, khi mất focus (blur), màu nền sẽ chuyển sang AQUA.



**Hình 4.39.** Kết quả của ví dụ 35 – Blur (hình trái) and focus (hình phải)

#### ➤ **onMouseOver**

Sự kiện onMouseOver được tạo ra bất cứ khi nào con trỏ chuột di chuyển lên trên một phần tử.

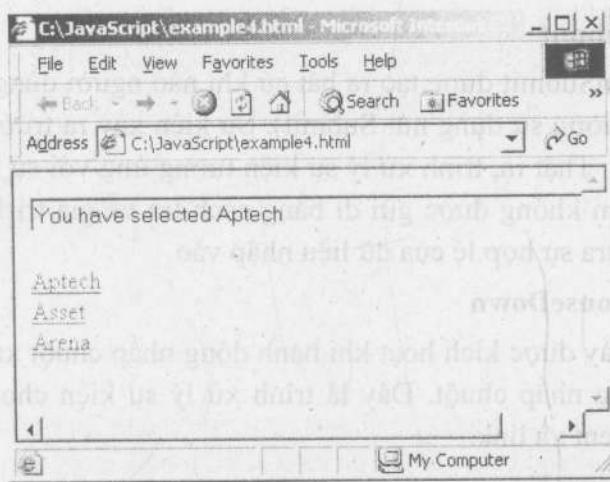
## ➤ onmouseout

Sự kiện onmouseout được tạo ra bất cứ khi nào con trỏ chuột di chuyển ra khỏi phần tử đó.

### Ví dụ 36

```
<html>
<head>
<script language = "Java Script">
    var num =0
    function showLink(num)
    {
        if (num==1)
        {
            document.forms[0].elements[0].value= "You
have selected Aptech";
        }
        if (num==2)
        {
            document.forms[0].elements[0].value = "You
have selected Asset";
        }
        if (num==3)
        {
            document.forms[0].elements[0].value = "You
have selected Arena";
        }
    }
</script>
</head>
<body>
<form>
    <input type=text size=60 >
</form>
<a href="#" onMouseOver="showLink(1)"document.bgcolor=
green">Aptech</a><br>
<a href="#" onMouseOver="showLink(2)">Asset</a><br>
<a href="#" onMouseOver="showLink(3)">Arena</a><br>
</body>
</html>
```

Khi di chuột qua Aptech, kết quả được hiển thị như sau :



Hình 4.40. Kết quả của ví dụ 36

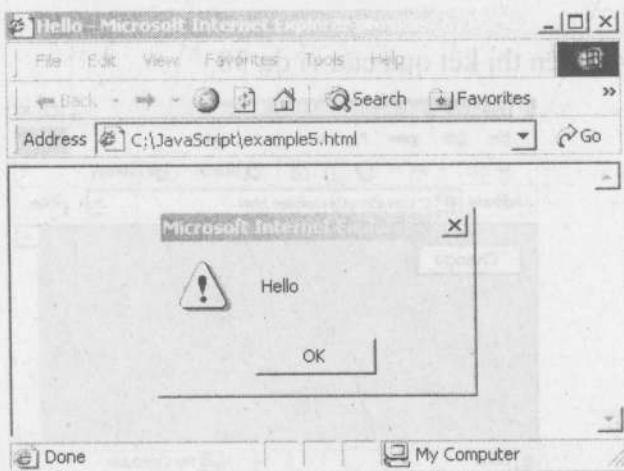
#### ➤ OnLoad

Sự kiện onLoad (áp dụng cho đối tượng body) được phát sinh khi đã tải xong tài liệu. Nó cũng được phát sinh khi một ảnh đã tải xong.

#### Ví dụ 37

```
<HTML>
<HEAD>
    <TITLE>Hello </TITLE>
</HEAD>
<BODY onLoad="alert('Hello')">
</BODY>
</HTML>
```

Kết quả :



Hình 4.41. Kết quả của ví dụ 37

### ➤ **onSubmit**

Sự kiện onSubmit được tạo ra bất cứ khi nào người dùng truyền dữ liệu từ form đi (thường sử dụng nút Submit). Sự kiện xảy ra trước khi form thật sự được gửi đi. Thật ra, trình xử lý sự kiện tương ứng với sự kiện này có thể ngăn chặn form không được gửi đi bằng cách trả về giá trị false. Cách này dùng để kiểm tra sự hợp lệ của dữ liệu nhập vào.

### ➤ **onMouseDown**

Sự kiện này được kích hoạt khi hành động nhấp chuột xảy ra. Nghĩa là, khi người dùng nhấp chuột. Đây là trình xử lý sự kiện cho các đối tượng button, document và link.

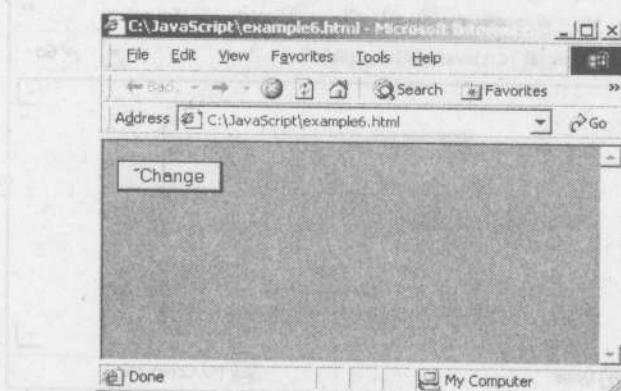
### ➤ **onMouseUp**

Sự kiện này được kích hoạt khi hành động thả chuột xảy ra. Nghĩa là, khi người dùng thả chuột. Đây là trình xử lý sự kiện cho đối tượng button, document và link.

### Ví dụ 38

```
<HTML>
<BODY BGCOLOR="lavender">
<FORM>
    <INPUT type = button name = text1 value="Change Color"
        onmousedown= " (document.bgColor='aqua') "
        onmouseup= " (document.bgColor='limegreen') "
</FORM>
</BODY>
</HTML>
```

Hình 4.42 hiển thị kết quả của ví dụ 38.



Hình 4.42. Kết quả của ví dụ 38

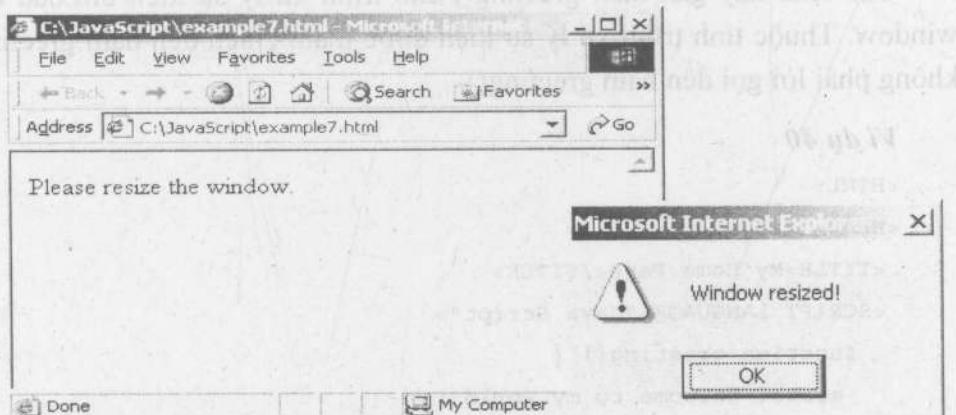
### ➤ onResize

Sự kiện này được kích hoạt khi hành động thay đổi kích thước cửa sổ xảy ra. Nghĩa là, khi người dùng hoặc một script làm thay đổi kích thước một cửa sổ hay frame. Đây là trình xử lý sự kiện cho các đối tượng Window.

#### Ví dụ 39

```
<html>
<head>
<script language="Java Script">
    window.onresize= notify;
    function notify()
    {
        alert("Window resized!");
    }
</script>
</head>
<body>
    Please resize the window.
</body>
</html>
```

Khi thay đổi kích thước cửa sổ, kết quả xuất hiện như sau:



Hình 4.43. Kết quả của ví dụ 39

### 5.3. Trình xử lý sự kiện

Khi một sự kiện được khởi tạo, chúng ta có thể tạo một đoạn mã Java Script để đáp ứng lại sự kiện. Đoạn mã này được gọi là trình xử lý sự kiện. Trình xử lý sự kiện có thể là một câu lệnh đơn, một tập hợp các câu lệnh hoặc một hàm.

```
<INPUT TYPE="button"
       NAME="docode"
       onClick="DoOnClick();">
```

Khi nhấp chuột vào một button, sự kiện onClick được khởi tạo. Sự kiện onClick gọi hàm DoOnClick và thực thi những câu lệnh bên trong hàm.

### 5.3.1. Trình xử lý sự kiện cho các thẻ HTML

Để khởi tạo trình xử lý sự kiện cho thẻ HTML, chúng ta phải chỉ định thẻ và thuộc tính trình xử lý sự kiện. Sau đó, gán mã Java Script. Đoạn mã phải được đặt trong cặp dấu nháy kép.

```
<TAG eventHandler="Java Script Code">
```

Các đối số chuỗi phải được đặt trong cặp dấu nháy đơn.

```
<INPUT TYPE="button" NAME="Button1" VALUE="Open Sesame!"
       onClick="window.open('mydoc.html', 'newWin')">
```

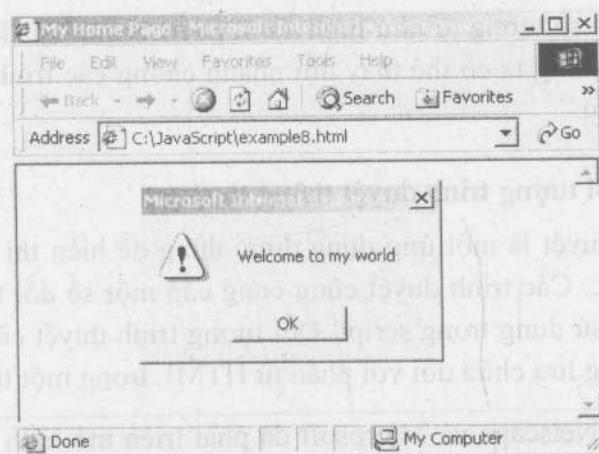
Thay vì sử dụng nhiều câu lệnh Java Script, hàm sẽ giúp cho việc thiết kế chương trình tốt hơn. Chúng ta sẽ gọi hàm khi cần thiết. Hơn nữa các hàm đó có thể được dùng bởi các phần tử khác.

Câu lệnh này gán hàm greeting() cho trình xử lý sự kiện onLoad của window. Thuộc tính trình xử lý sự kiện được tham chiếu đến hàm greeting, không phải lời gọi đến hàm greeting().

#### Ví dụ 40

```
<HTML>
<HEAD>
  <TITLE>My Home Page</TITLE>
  <SCRIPT LANGUAGE="Java Script">
    function greeting() {
      alert("Welcome to my world");
    }
  </SCRIPT>
</HEAD>
<BODY onLoad="greeting()">
</BODY>
</HTML>
```

Kết quả:



Hình 4.44. Kết quả ví dụ 40

### 5.3.2. Trình xử lý sự kiện như là những thuộc tính

Chúng ta cũng có thể gán một hàm cho một trình xử lý sự kiện của một đối tượng. Cú pháp như sau:

```
object.eventhandler = function;
```

Ví dụ,

```
window.onload = greeting;
```

Chúng ta xem lại ví dụ trên và sử dụng trình xử lý sự kiện như những thuộc tính:

#### Ví dụ 41

```
<HTML>
<HEAD>
    <TITLE>My Home Page</TITLE>
    <SCRIPT LANGUAGE="Java Script">
        <!--
            function greeting() {
                alert("Welcome to my world");
            }
            window.onload = greeting;

        // -->
    </SCRIPT>
</HEAD>
</HTML>
```

Kết quả sẽ tương tự như hình 4.44. Điểm mạnh của kỹ thuật này là tính linh hoạt. Chúng ta có thể thay đổi nhanh chóng các trình xử lý sự kiện khi được yêu cầu.

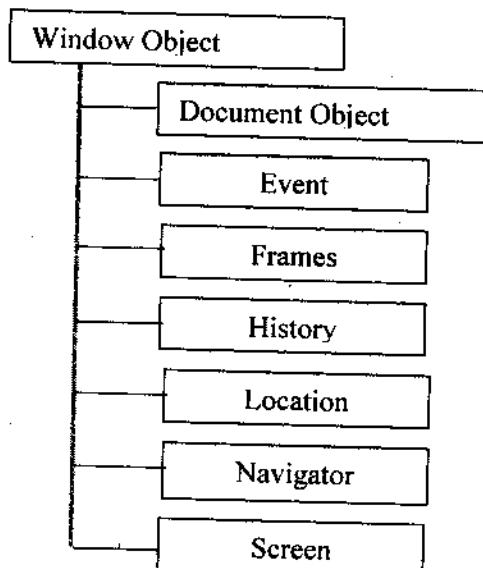
#### 5.4. Đối tượng trình duyệt thông thường

Trình duyệt là một ứng dụng được dùng để hiển thị nội dung của một trang HTML. Các trình duyệt cũng cung cấp một số đối tượng có thể được truy cập và sử dụng trong script. Đối tượng trình duyệt cũng hoạt động như các đối tượng lưu chứa đối với phần tử HTML trong một trang Web.

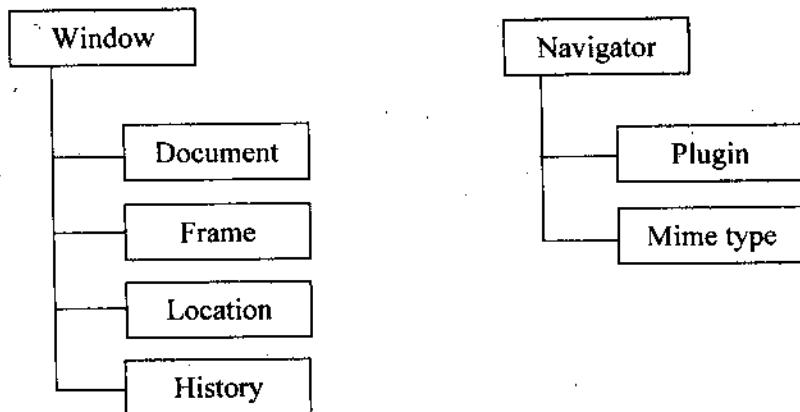
**Lưu ý:** Netscape và Microsoft đã phát triển mô hình đối tượng tài liệu riêng của mình (DOM). Hai mô hình này có thể không tương thích với nhau.

Đối tượng window thể hiện cửa sổ của trình duyệt. Tất cả những đối tượng khác trong mô hình được kế thừa từ đối tượng window và được truy cập thông qua đối tượng window. Có thể có thêm các đối tượng window khác. Các đối tượng này thể hiện các cửa sổ con và được xem như các Frame. Những đối tượng cửa sổ này được truy cập thông qua tập hợp các Frame.

Trong phần này chúng ta sẽ tìm hiểu các đối tượng mà Internet Explorer và Netscape Navigator cung cấp cùng với một số thuộc tính và phương thức liên quan đến chúng.



Hình 4.45. Đối tượng trình duyệt IE



**Hình 4.46.** Đối tượng trình duyệt Netscape

#### 5.4.1. Mô hình đối tượng (DOM)

Một đặc điểm quan trọng của Java Script là ngôn ngữ dựa trên đối tượng. Điều này hỗ trợ cho người sử dụng phát triển chương trình theo môđun và có thể sử dụng lại. Java Script không hoàn toàn là ngôn ngữ hướng đối tượng, nhưng nó hỗ trợ một vài đặc điểm hướng đối tượng.

Một đối tượng có thể được định nghĩa như một thực thể đơn bao gồm các thành phần dữ liệu gọi là các **thuộc tính (Property)** và thành phần chức năng gọi là các **phương thức (Method)**.

Một thuộc tính là một giá trị gắn với một đối tượng. Tất cả các đối tượng Java Script chuẩn đều có những thuộc tính như vậy. Ví dụ:

**Document.bgcolor**

#### 5.4.2. Đối tượng Window

Thể hiện các cửa sổ của trình duyệt và có thể được dùng để lấy thông tin về trạng thái của cửa sổ. Nó cũng được sử dụng để hiển thị đối tượng document và truy cập các sự kiện xảy ra trong cửa sổ và những đặc điểm của trình duyệt ảnh hưởng đến cửa sổ.

Khi trình duyệt mở tài liệu HTML nó thường tạo ra một đối tượng window. Tuy nhiên, nếu một tài liệu có sử dụng một hay nhiều Frame thì trình duyệt tạo ra một đối tượng window cho tài liệu ban đầu và tạo thêm các đối tượng window nữa cho mỗi Frame. Các đối tượng window mới được tạo ra đó là các cửa sổ con của cửa sổ ban đầu, chúng có thể bị ảnh

hướng bởi các hành động xảy ra trên cửa sổ cha. Ví dụ, nếu chúng ta đóng cửa sổ ban đầu thì tất cả các cửa sổ con đều bị đóng.

### Các thuộc tính

Tên	Mô tả
document	Thể hiện tài liệu HTML trong cửa sổ của trình duyệt.
event	Thể hiện trạng thái của một sự kiện, ví dụ như phần tử mà trong đó sự kiện xuất hiện, trạng thái của các phím trên bàn phím, vị trí của chuột, trạng thái của các nút trên chuột.
history	Lưu trữ thông tin của những URL đã được viếng thăm.
location	Lưu trữ thông tin của URL hiện tại.
name	Thiết lập hoặc khôi phục tên của window hoặc Frame.
navigator	Lưu trữ thông tin của trình duyệt.
screen	Lưu trữ thông tin về kích cỡ và dạng đầy đủ của màn hình mà người dùng sử dụng.
status	Thiết lập hoặc khôi phục thông báo trên thanh trạng thái của cửa sổ trình duyệt.

### Những phương thức

Tên	Mô tả
Alert	Hiển thị một hộp chứa thông báo do ứng dụng định nghĩa.
Blur	Làm cho đối tượng mất focus và kích hoạt sự kiện onblur.
Close	Đóng cửa sổ trình duyệt hiện thời hoặc ứng dụng HTML.
Focus	Đặt một điều khiển là focus và thực thi đoạn lệnh được chỉ ra trong sự kiện onload.
navigate	Tải URL được chỉ định vào cửa sổ hiện thời.
open	Mở cửa sổ mới và tải tài liệu được chỉ ra bởi một URL hoặc mở một tài liệu trống nếu không có một URL nào được chỉ định.

**Chú ý:** Các thuộc tính của window, phương thức và tập hợp các tên là các từ khoá dùng riêng không được dùng làm tên biến và các thủ tục thông thường.

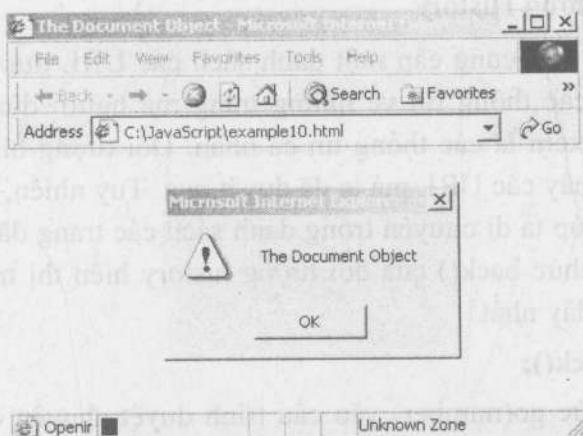
#### 5.4.3. Đối tượng Document

Thể hiện tài liệu HTML trong cửa sổ trình duyệt và được dùng để lấy thông tin về tài liệu, kiểm tra và sửa đổi các phần tử HTML và văn bản trong tài liệu và xử lý sự kiện.

## Ví dụ 42

```
<HTML>
<HEAD>
    <TITLE>The Document Object</TITLE>
    <script language="Java Script">
        alert(document.title);
    </SCRIPT>
</HEAD>
</HTML>
```

Kết quả :



Hình 4.47. Đổi tượng trình duyệt Netscape

### Các thuộc tính

Tên	Mô tả
aLinkColor	Thiết lập hay lấy màu của tất cả các liên kết động trong tài liệu.
bgColor	Thiết lập hay lấy màu nền của đối tượng document.
Body	Chỉ ra phần đầu và phần cuối của thân tài liệu.
fgColor	Thiết lập hay lấy màu (văn bản) nền trên cùng của tài liệu.
linkColor	Thiết lập hay lấy màu của các liên kết tài liệu.
location	Chứa các thông tin về URL hiện tại.
Title	Chứa tiêu đề của tài liệu.
URL	Thiết lập hay lấy URL cho tài liệu hiện thời.
vlinkColor	Thiết lập hay lấy màu của các liên kết mà người dùng đã viếng thăm.

## Các phương thức

Tên	Mô tả
Clear	Xoá tài liệu hiện thời.
Close	Đóng một luồng dữ liệu xuất và hiển thị dữ liệu nhận được.
Open	Mở một tài liệu để nhận dữ liệu từ phương thức write hoặc writeln.
Write	Viết một hay nhiều biểu thức HTML đến một tài liệu trong cửa sổ được chỉ định.
WriteLn	Viết một hay nhiều biểu thức HTML được theo sau bởi một dấu xuống dòng đến một tài liệu trong cửa sổ được chỉ định.

## 5.5. Đối tượng History

Đối tượng này cung cấp một danh sách các URL được duyệt qua gần đây nhất. Bởi các thông tin về những trang mà người dùng đã duyệt qua trước đó được xem là các thông tin cá nhân. Đối tượng history không cho chúng ta nhìn thấy các URL mà ta đã duyệt qua. Tuy nhiên, nó cung cấp các phương thức giúp ta di chuyển trong danh sách các trang đã duyệt trước đó. Ví dụ phương thức back() của đối tượng history hiển thị trang tài liệu vừa mới duyệt gần đây nhất.

### history.back();

Phương thức go(number) yêu cầu trình duyệt chuyển đến một tài liệu xác định trong danh sách các trang đã duyệt qua.

Nếu là số âm, ví dụ như history.go(-2) thì trình duyệt sẽ chuyển từ tài liệu hiện thời đến các tài liệu trước nó trong danh sách các trang đã duyệt qua (ví dụ này cũng giống như khi ta nhấn nút Back 2 lần).

Nếu là số dương, trình duyệt sẽ chuyển từ tài liệu hiện thời đến các tài liệu sau nó trong danh sách các trang đã duyệt qua.

## Các phương thức

Tên	Mô tả
back	Nạp URL trước đó trong danh sách các trang đã duyệt qua.
forward	Nạp URL đứng sau URL hiện tại trong danh sách các trang đã duyệt qua
go	Nạp một URL xác định trong danh sách các trang đã duyệt qua.

## 5.6. Đối tượng Location

Đối tượng này lưu thông tin về URL hiện tại. Nó cung cấp phương thức cho phép URL hiện tại có thể được tải lại.

### Các thuộc tính

Tên	Mô tả
Hash	Thiết lập hay lấy phần dữ liệu sau dấu # trong thuộc tính href
host	Thiết lập hay lấy tên máy chủ và số cổng của URL.
hostname	Thiết lập hay lấy phần tên của máy chủ của URL.
href	Thiết lập hay lấy toàn bộ URL như một chuỗi.

### Các phương thức

Tên	Mô tả
assign	Nạp một tài liệu HTML mới.
reload	Nạp lại một trang hiện thời.
replace	Thay thế tài liệu hiện thời bằng cách tải một tài liệu khác với URL xác định.

# *Chương 5*

## **THIẾT KẾ GIAO DIỆN**

### **1. MỤC ĐÍCH CỦA WEBSITE CẦN THIẾT KẾ**

Bước đầu tiên trong công đoạn thiết kế một Website là cần xác định chính xác, chắc chắn về những nội dung sẽ đưa vào trong Website. Không có chủ định và mục tiêu rõ ràng thì Website sẽ trở nên lan man, sa lầy và cuối cùng không đạt hiệu quả, mục đích ban đầu đặt ra. Thiết kế cẩn thận và định hướng rõ là chìa khoá dẫn đến thành công cho việc xây dựng một Website. Trước khi xây dựng một Website, chúng ta nên:

- Xác định đối tượng độc giả của Website. Website có mục đích rõ ràng.
- Thiết lập các chủ đề chính của website.
- Thiết kế các khối thông tin chủ yếu mà website sẽ cung cấp.

Nên bắt đầu với việc xác định nguồn tài nguyên về nội dung, hình ảnh thông tin mà chúng ta cần đến để tạo ra website phù hợp với mục đích được đề ra - đó là nguồn thông tin sẽ duy trì cho website hoạt động sau này.

#### **1.1. Xác định mục tiêu cơ bản của website**

Trước tiên cần có một sự khái quát ngắn và rõ ràng cho các mục tiêu của website, điều này sẽ giúp đỡ rất nhiều cho công việc thiết kế. Đó là điểm xuất phát để chúng ta mở rộng đến các mục tiêu chính và cũng là một công cụ hữu hiệu để đánh giá sự thành công của một website. Xây dựng một website là cả một quá trình liên tục, không đơn thuần chỉ là một dự án duy nhất, một lần với các thông tin tĩnh. Việc biên tập, quản lý và duy trì kỹ thuật dài hạn nhất định phải bao trùm kể hoạch xây dựng website.

Thiếu điều này, tương lai của một website sẽ giống như những nhà văn, nhà báo, đầy lòng say mê ban đầu, nhưng cuối cùng không đạt được thành tựu nào.

## **1.2. Xác định độc giả**

Bước tiếp theo trong công việc thiết kế là xác định các độc giả chính của website, để chúng ta có thể thiết kế cấu trúc phù hợp với nhu cầu, mong muốn của họ. Sự hiểu biết, trình độ, sở thích cũng như yêu cầu của độc giả rất khác nhau. Một hệ thống được thiết kế tốt sẽ thích hợp cho một dài rộng trình độ, nhu cầu độc giả. Ví dụ, nếu mục đích website của chúng ta là chuyển tải các thông tin về tình hình nội bộ công ty, các tài liệu về nhân lực, hay các tài liệu khác dùng cho việc in các hướng dẫn thì sẽ thích hợp cho các độc giả hàng ngày và cho cả các độc giả truy nhập không thường xuyên.

### **1.2.1. Độc giả (websurfer)**

Các trang chủ (homepage) cho các chương trình duyệt web cũng nên tương tự như các bìa tạp chí. Mục tiêu là tính tổng thể với hình ảnh sinh động, chủ đề, nội dung gây được sự chú ý. Tất cả các liên kết trên trang chủ này chỉ nên liên kết đến các trang bên trong website. Chúng ta cũng phải tạo được thông báo ngắn gọn, súc tích những nội dung trong website mà độc giả thường quan tâm.

### **1.2.2. Độc giả mới và độc giả không thường xuyên**

Độc giả nhóm này phụ thuộc vào một cấu trúc site rõ ràng, dễ truy nhập đến sự tổng quát về cấu trúc toàn bộ thông tin trên website. Người mới truy nhập sẽ có cảm giác bị đe dọa bởi một cấu trúc menu phức tạp, ngại đi sâu vào website nếu trang đầu không có hình ảnh hấp dẫn và không được sắp xếp rõ ràng. Theo Jakob Nielsen ở Sun Microsystems, có dưới 10% độc giả cuộn màn hình xuống dưới phần đầu của một trang web. Các độc giả không thường xuyên lại chú trọng đến trang khai quát, cấu trúc có phân lớp và hình ảnh đồ họa, biểu tượng giúp họ nhớ, kết nối đến vị trí thông tin họ cần trong website. Một từ điển các khái niệm kỹ thuật, từ viết tắt và danh sách các vấn đề thường được xảy ra (FAQ - frequently asked questions) có thể rất hữu dụng cho các độc giả mới và không thường xuyên.

### **1.2.3. Các độc giả chuyên nghiệp, thường xuyên**

Số độc giả này phụ thuộc vào website của chúng ta để nhận thông tin nhanh chóng và chính xác. Các độc giả chuyên nghiệp rất sot ruột với menu nhiều lớp có đồ họa chất lượng kém mà chỉ cung cấp từ 2 đến 6 lựa chọn

một lần. Họ trông đợi các menu không đồ họa (text thuần tuý), có thể kéo thả được và thời gian nạp xuống nhanh chóng. Các kiểu tạo dáng đồ họa dễ gây sự khó chịu. Các độc giả chuyên nghiệp thường có chủ định rõ ràng, do vậy sẽ đánh giá cao các menu text chi tiết, các nét chính của website hoặc các chỉ số site phong phú giúp họ tìm kiếm, thu nhận thông tin nhanh chóng.

#### *1.2.4. Các độc giả nước ngoài*

Chúng ta đang thiết kế cho World Wide Web, vì vậy các độc giả có thể là một người trong cùng thành phố, nhưng cũng có thể là một người đang ở các quốc gia khác như Mỹ, Nhật Bản... Để đáp ứng tối đa lượng độc giả trên các quốc gia khác nhau, nội dung trang web cần được biên dịch, ít nhất cũng là các trang chủ đạo, tránh dùng các từ địa phương hoặc các khái niệm kỹ thuật, viết tắt trong bản giới thiệu hay các trang giải thích. Ví dụ, không viết tắt ngày trên các trang web, vì đối với một người Mỹ, thì "10/5/97" sẽ được hiểu là ngày 5 tháng 10 năm 1997, nhưng đa số độc giả các quốc gia khác sẽ hiểu đó là ngày 10 tháng 5 năm 1997.

## **2. CHIẾN LƯỢC THIẾT KẾ**

Mọi hình thức trình bày thông tin đều bị không chế bởi các yếu tố được xác định bởi mục tiêu của Website, bởi môi trường chúng ta chọn và bởi bản thân các độc giả đối với thông tin chuyền tải, mà cụ thể là phụ thuộc vào hai yếu tố: cấu trúc trình bày và thời gian truy nhập đối với độc giả.

### **2.1. Đạo chơi trên Web(browsing)**

Trong thế giới rộng lớn của World Wide Web, các websurfer là các độc giả lang thang, không lý do, những người có thể đọc website mà không yêu cầu bất cứ sự thúc ép nào, hoặc một ý định nào. Chúng tôi không có ý định trình bày các kỹ thuật để lôi kéo các độc giả nhóm này vào các website với mục đích giải trí, thương mại. Một mục đích khác của các website chính là các site cho các doanh nghiệp, giáo dục, dạng "intranet", nơi mà độc giả tìm đọc với mục đích rõ ràng hơn nhiều.

## **2.2. Đào tạo**

Các ứng dụng đào tạo trên cơ sở công nghệ web luôn tuân theo trình tự về khía cạnh thiết kế, ít khi đi lạc để từ trang giới thiệu chính. Không nên làm độc giả và cả mục đích của chúng ta bị lộn xộn bởi các mối liên kết ngoài thông tin chủ chốt. Giới hạn các liên kết bằng nút "Tiếp tục", hay "Quay về trang trước" đảm bảo mọi độc giả sẽ nhìn thấy cùng một giáo trình, cho phép dự đoán chính xác hơn thời gian truy nhập của người đọc. Đại đa số giáo trình già thiết thời gian truy nhập dưới một giờ, hoặc sẽ được phân đoạn thành các phần ít hơn một giờ. Chúng ta cũng nên thông báo cho người đọc về lượng thời gian của bài giảng, hoặc cũng lưu ý họ đừng đi xa khỏi phần chính của bài giảng nếu bài giảng đó cần phải trả tiền để đọc.

Các ứng dụng đào tạo loại này thường yêu cầu sự log-in của độc giả, và cũng thường sử dụng câu hỏi dạng form có trả lời đúng/sai hoặc theo dạng lựa chọn câu trả lời từ một danh sách. Thông tin về độc giả, bảng điểm được lưu trữ trong cơ sở dữ liệu được liên kết với website.

## **2.3. Dạy học**

Trong các ứng dụng dạy học dựa trên công nghệ web, thông tin được trình bày thường tinh tế và có chiều sâu hơn là trong các ứng dụng đào tạo. Các mối liên kết là mặt mạnh của web, tuy nhiên chúng có thể gây khó khăn cho các học sinh từ trang trình bày chính. Nếu cho phép người đọc liên kết đến các tài nguyên web khác ngoài website của chúng ta thì nên nhóm các liên kết trong trang này cách biệt khỏi phần thông tin chính.

Thông thường người đọc muốn in thông tin trên web và sẽ đọc chúng sau này. Chúng ta nên cung cấp cho họ một phiên bản "in" riêng, trong đó các trang riêng biệt, ngắn sẽ được gộp lại thành một trang dài.

## **2.4. Giáo dục**

Những trang web có phong cách thiết kế quá thu gọn, quá trình tự sẽ không phù hợp với những độc giả muốn tự học hỏi, tìm hiểu khám phá. Các độc giả nhóm này thường có trình độ cao. Thiết kế một cấu trúc uyển chuyển, có tương tác, không đơn điệu sẽ lý tưởng đối với các độc giả này, do rất khó đoán định chính xác chủ đề nào sẽ được quan tâm nhất. Thiết kế cũng phải cho phép truy nhập nhanh đến một phạm vi rộng các chủ đề và

phong phú với các liên kết đến các thông tin có liên quan trên website của chúng ta hay trên các website khác. Các danh sách dạng text của các liên kết cũng rất thích hợp cho các mục lục, bìa chỉ số vì chúng được nạp xuống nhanh, đầy đủ thông tin, nhưng không sinh động nên dễ gây sự nhàm chán vì thế cần có hình ảnh đồ họa thiết kế đẹp, thay đổi cùng các minh họa đi kèm thông tin. Thời gian truy nhập không thể dự đoán được, nhưng thường ngắn hơn các site cho đào tạo, giáo dục vì độc giả thường có nhu cầu tra cứu thông tin nhanh chóng tức thời. Lựa chọn cho in ấn cũng là bắt buộc phải có cho các độc giả này.

## 2.5. Tham khảo

Các website tham khảo được thiết kế tốt cho phép người đọc nhanh chóng đi thẳng vào vấn đề, tìm nội dung họ cần và sau đó dễ dàng in hoặc lưu giữ thông tin tìm thấy. Thông thường thông tin không phải là các "câu chuyện", do đó cấu trúc của nó hoàn toàn không có trình tự. Cấu trúc menu, nội dung nhất định phải được tổ chức cẩn thận để hỗ trợ tìm kiếm, thu nhận nhanh, dễ lưu giữ các file, in ấn khi cần. Cần giữ các hình ảnh đồ họa nhỏ để thời gian nạp xuống nhanh, và chúng ta cũng nên nghiên cứu, áp dụng phần mềm tìm kiếm thay vì chỉ cung cấp một danh sách các liên kết. Thời gian liên kết càng ngắn càng tốt.

# 3. THIẾT KẾ GIAO DIỆN

## 3.1. Khái quát

Các độc giả của web không chỉ xem thông tin, họ tương tác với nó theo cách thức mới không như các tiền lệ trong việc thiết kế tài liệu giấy. Giao diện người dùng đồ họa (GUI) của hệ thống, cộng thêm các tương tác ẩn dụ, hình ảnh và các quan niệm được sử dụng để chuyển tải mọi tính năng, thông tin lên màn hình, và trải qua thời gian, các đặc thù trực quan của các thành phần giao diện đồ họa và sự tương tác chức năng hoá đã tạo nên nét đặc trưng "nhìn thấy và cảm nhận" của các trang web cùng các mối liên kết hypertext. Thiết kế đồ họa và các hình ảnh "ký hiệu" trực giác không chỉ để làm đẹp trang web - đồ họa trở thành một phần được tích hợp bởi kinh nghiệm bởi độc giả đối với Website của chúng ta. Trong các tài liệu có kèm

theo ảnh, không thể hoàn toàn tách rời thiết kế đồ họa với thiết kế giao diện.

### **3.2. Trang Web đối lập với thiết kế tài liệu thông thường**

Đại đa số nhận thức hiện tại của chúng ta về thông tin có cấu trúc bắt nguồn từ việc tổ chức các quyển sách, tạp chí xuất bản định kỳ đã được in và công tác chi số hoá thư viện, các hệ thống phân loại được phát triển quanh các ấn bản đó. "Các tiêu chuẩn về hình thức" của các cuốn sách trong thế giới nói tiếng Anh đã được thiết lập cũng như công nhận rộng rãi; các hướng dẫn chi tiết cho việc tạo một cuốn sách có thể tìm đọc trong các quyển như The Chicago Manual of Style. Mọi đặc điểm cho một cuốn sách, từ mục lục đến chú thích, chỉ số đã được rút ra qua hàng thế kỷ, và người đọc trước kia cũng đã phải đổi diện với các vấn đề tổ chức giống như người đọc bây giờ đối với các tài liệu đa phương tiện (hypermedia).

Cuốn Kinh thánh năm 1456 của Gutenberg thường được trích dẫn như cuốn sách hiện đại đầu tiên, tuy thế cũng phải sau khi phát triển mạnh việc xuất bản cùng hơn 100 năm kinh nghiệm thì việc đánh số trang, chỉ số, mục lục cũng như các trang bìa mới trở thành đặc điểm phổ biến cho các cuốn sách. Các tài liệu web tương tự như vậy, cũng phải trải qua sự phát triển và sự chuẩn hoá cách thức thông tin được tổ chức, thực hiện ở dạng điện tử.

### **3.3. Các tiền lệ trong in ấn**

Mặc dù các tài liệu đa phương tiện trên mạng đưa ra các thách thức mới cho những nhà thiết kế thông tin, nhưng nói chung, những hướng dẫn cần thiết để thiết kế, tạo lập, lắp ráp, hiệu chỉnh và tổ chức một media phong phú, đa dạng về cẩn bản không khác so với thực tế công việc của nhà xuất bản. Đa số trang web được tạo ra phù hợp với các quy ước cho hình thức biên tập và tổ chức văn bản của "The Chicago Manual of Style". Các phương pháp tạo ra tài liệu rõ ràng, bao quát, nhất quán có thể đọc trong các hướng dẫn như "Xerox Publishing Standards: A Manual of style and Design". Không nên bị "lạc hướng" theo các trang web lạ mà bỏ qua các chuẩn mực cơ bản của thiết kế đồ họa cũng như biên tập.

### **3.4. Thiết kế các trang web độc đáo**

Các trang World Wide Web khác sách và các tài liệu khác ở một nét cơ bản: các mối liên kết siêu văn bản cho phép người đọc truy nhập đến một trang web đơn lẻ không cần đến lời nói đầu hay tựa đề. Điều này có nghĩa là các đầu trang, chân trang của trang web sẽ phức tạp, nhiều thông tin hơn các trang in trên giấy.

Đối với những trang sách, việc lặp lại các thông tin bản quyền, tác giả, thời gian xuất bản là điều không cần thiết, nhưng một trang web riêng lẻ thường cần đến những thông tin này vì các trang đơn lẻ có thể chỉ là một phần nhỏ của cá website mà độc giả nhìn thấy. Vấn đề tạo các trang web độc đáo không chỉ riêng đối với các trang web. Các báo chuyên đề, tạp chí, đa số các báo ngày đều lặp lại thời gian phát hành, số bản tại phần đầu hay cuối từng trang vì họ biết độc giả thường cắt các bài báo, hoặc photocopy các trang từ tạp chí và cần thông tin trích dẫn để theo dõi nguồn gốc nguyên bản của các bài.

Để tạo các website vừa dễ sử dụng, vừa đầy đủ nội dung, chiến lược thiết kế tốt nhất là áp dụng nhất quán một số quy ước thiết kế cơ bản trong tất cả các trang web. Thiết kế các thành phần cơ bản của một tài liệu không phức tạp và việc đó càng trở nên đơn giản khi áp dụng công nghệ Internet. Nó giống với điều được giảng trong một lớp báo chí: ai, cái gì, khi nào và ở đâu.

- *Ai*

Ai đang nói? Câu hỏi này thật cơ bản và thông tin thường được đàm bảo đến nỗi các tác giả của web thường bỏ qua phần tin quan trọng nhất mà một độc giả cần đến để đánh giá nguồn gốc, lai lịch của tài liệu: ai đang nói với tôi? Dù trang web có xuất phát từ một cá nhân hay từ một viện nghiên cứu, cũng luôn phải thông báo cho người đọc ai đã tạo trang web đó. Các website truyền bá thông tin không chính xác, không rõ nguồn gốc, thậm chí mang tính lừa gạt nhiều khi lan truyền nhanh chóng, lẩn át cuộc điều tra và kết luận chính xác.

- *Cái gì*

Mọi tài liệu đều cần tiêu đề dễ hiểu để gây sự chú ý của độc giả, nhưng trong nhiều trường hợp đặc biệt đối với các trang web, yếu tố cơ bản này lại trở nên cực kỳ quan trọng. Tiêu đề là điều đầu tiên các độc giả của các trang World Wide Web nhìn thấy khi các trang được nạp dần xuống. Trong các

trang nhiều hình ảnh đồ họa, tiêu đề là cái duy nhất mà người ta có thể nhìn thấy trong thời gian chờ ảnh được nạp lên trang web. Ngoài ra, tiêu đề là phần văn bản sẽ được chương trình duyệt sử dụng như một "đánh dấu trang" nếu độc giả chọn nó để đưa vào danh sách ghi nhớ. Một tiêu đề không sát thực, tối nghĩa, hoặc một tiêu đề quá kỹ thuật sẽ không giúp người đọc nhớ được tại sao họ lại cho trang web này vào danh sách lưu trữ.

#### -- *Khi nào*

Tính hợp thời là một yếu tố quan trọng trong việc đánh giá toàn bộ một tài liệu. Chúng ta làm thông tin trong thời đại mà các tài liệu giấy đã đám bảo: báo, tạp chí, và hầu như các quan hệ thư từ văn phòng đã được xác định ngày tháng. Do đó chúng ta cần đề ngày tháng cho mọi trang web và thay đổi mỗi khi tài liệu được cập nhật. Điều này đặc biệt quan trọng đối với các văn bản online dài và phức tạp mà chỉ được cập nhật từng phần, tuy vậy điều này cũng có thể không đủ nổi bật cho các độc giả vãng lai, không thường xuyên. Thông tin về doanh nghiệp, các bản hướng dẫn, thông tin về sản phẩm và các tài liệu kỹ thuật dưới dạng trang web nên có thêm thời gian được sửa chữa, cập nhật lại.

#### -- *Nơi nào*

World Wide Web là một địa điểm duy nhất có chiều thông tin cực lớn nhưng lại có ít chỉ dẫn rõ ràng về vị trí thực sự của tài liệu. Nhấn vào một liên kết web, chúng ta có thể kết nối đến một web server ở Sydney (Úc) hay Chicago (Mỹ) hoặc một nơi nào đó trên Internet rộng lớn. Nếu bạn không thông thạo lắm trong việc phân tích địa chỉ Internet (URL), thật khó khăn để xác định tài liệu đang thực sự nằm ở đâu. Sau cùng đây chính là World Wide Web, và câu hỏi tài liệu đến từ đâu đôi khi không thể tách khỏi câu hỏi tài liệu này của ai. Chúng ta nên luôn luôn thông báo cho độc giả chúng ta đang ở đâu, cùng với (nếu nó thích hợp) các thông tin về doanh nghiệp, viện, cơ quan của mình.

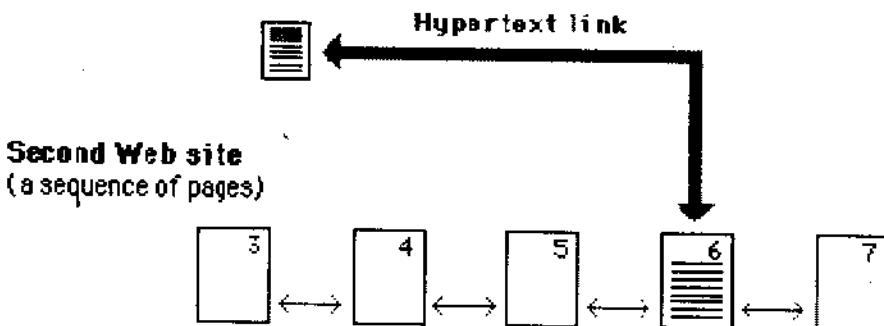
Nên kết hợp địa chỉ của trang chủ (home URL) ít nhất ở các trang chính trong website để tạo thuận tiện cho việc thực hiện các kết nối đến chúng ta. Một khi độc giả lưu các trang như một file text hoặc in ra giấy, các mối liên kết có thể sẽ bị mất đi.

Các phiên bản mới của một số trình duyệt chính có thêm tính năng tự động thêm địa chỉ vào các bản in, nhưng rất nhiều độc giả không sử dụng

các đặc tính nâng cao này. Và bây giờ, đa số có hàng "núi" bản in các trang web, không dễ gì tìm lại địa chỉ của một tài liệu nào đó.

Hãy sử dụng nhất quán tiêu đề, các thông tin phụ trợ như tên sở hữu, thời điểm cập nhật và ít nhất một liên kết đến trang chủ trong các trang web của chúng ta. Nên đặt địa chỉ của trang chủ lên một số trang chính trong website. Thêm các thành phần cơ bản này, chúng ta đã thực hiện được 90% con đường để cung cấp cho các độc giả một giao diện web có thể hiểu được.

## First Web site



### 3.5. Thiết kế giao diện cơ bản

#### 3.5.1. Thiết kế hướng tới người sử dụng

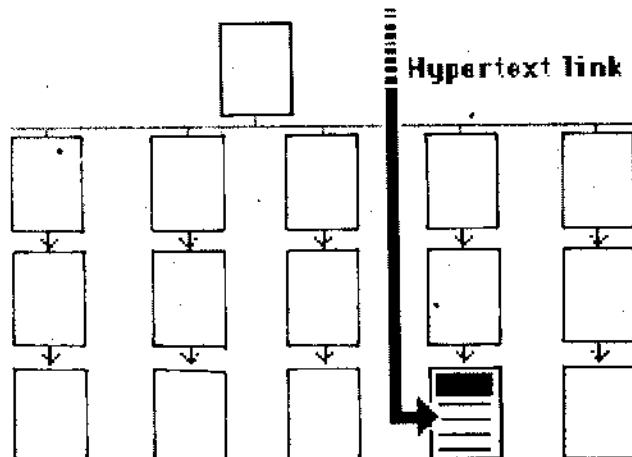
Giao diện đồ họa cho người sử dụng (GUI) được thiết kế để người dùng điều khiển được trực tiếp máy tính của họ. Ngày nay, người sử dụng yêu cầu một mức độ hoàn hảo về thiết kế giao diện đồ họa, kể cả các trang web. Mục đích là cung cấp các nội dung cần thiết cho tất cả độc giả.

Đây là nơi nghiên cứu về nhu cầu và tâm lý học khách hàng, đó là những yếu tố quyết định. Không thể thiết kế cho một độc giả mà không biết đến các yêu cầu của người đó. Chúng ta nên tạo các kịch bản mẫu cho các nhóm độc giả đang tìm kiếm thông tin trên website. Một độc giả có kinh nghiệm tìm một mẫu tin nhất định có được giúp đỡ hay cảm trở bởi thiết kế của chúng ta? Một độc giả mới có bị lúng túng trước một hệ thống menu phức tạp? Thủ nghiệm các thiết kế và nhận các phản hồi từ độc giả là

phương pháp tốt nhất để tìm ra các ý tưởng thiết kế cho phép độc giả nhận được cái họ muốn từ website của chúng ta.

### 3.5.2. Các giúp đỡ định hướng rõ ràng

DEAD END DOCUMENT



(Được thu gọn từ hình ảnh lấy tại: [home.netscape.com](http://home.netscape.com))

Với thực tại của công nghệ web, đa số độc giả tương tác với các trang web bằng cách thực hiện các liên kết giữa các tài liệu. Vấn đề chủ yếu của giao diện trong các website là độc giả không ý thức được họ đang ở đâu trong tổ chức thông tin. Các biểu tượng nhất quán, dễ hiểu, các lược đồ đồ họa đồng nhất và bản khái quát (đồ họa hay văn bản), màn hình tổng hợp có thể cho độc giả sự tin tưởng là họ có thể tìm thấy những điều cần thiết mà không lãng phí thời gian. Độc giả phải luôn có khả năng quay trở lại trang chủ và các điểm chủ chốt trên website. Các liên kết cơ bản này nên có trên mọi trang web của chúng ta, nó thường là các nút ấn đồ họa với hai mục đích: tạo các mối liên kết cơ bản và giúp tạo một biểu tượng đồ họa thông báo cho độc giả biết họ vẫn còn đang ở trong Website của chúng ta. Ví dụ, trên website của Netscape, thanh biểu tượng sau đây có tại cuối mỗi trang web:

Thanh nút ấn hay được sử dụng (cho nhiều lựa chọn trong một diện tích nhỏ), có thể đoán trước (có ở tất cả các chân trang), và tạo một đặc tính đồng nhất cho mọi trang web của Netscape site.

### **3.5.3. Không có trang cuối cùng (dead-end)**

Các trang web nên có ít nhất một liên kết. Các trang "dead-end" - các trang không móc nối đến các trang khác trong cùng site - sẽ gây ra sự thất vọng cho độc giả, làm mất khả năng đưa độc giả đến với các trang khác trong website của chúng ta.

Các trang web thường được đưa ra không có lời tựa đầu: độc giả thường tạo hay đi theo các liên kết thẳng đến các trang cất sâu trong cấu trúc của website. Do vậy họ có thể không bao giờ nhìn thấy trang chủ (Homepage) hoặc các thông tin mở đầu trên website của chúng ta. Nếu các trang phía dưới của site không có các liên kết quay lên về trang chủ hoặc quay lại menu, độc giả sẽ không còn cơ hội truy nhập đến các phần còn lại của website.

Cho phép truy nhập trực tiếp mục đích là cung cấp cho độc giả thông tin họ cần với ít bước nhất và với thời gian ngắn nhất. Điều này có nghĩa là chúng ta cần thiết kế cấu trúc thông tin hiệu quả nhất, giảm tối đa các bước qua hệ thống menu. Các nghiên cứu về giao diện cho thấy độc giả thích các menu có từ 5 đến 7 liên kết, và một lượng nhỏ trang màn hình nhiều lựa chọn hơn là một lượng lớn trang với các menu đơn giản.

Bảng sau đây cho thấy, không cần nhiều mức menu để tạo nên một số lượng lớn lựa chọn:

Số menu lồng nhau	Số phần tử trong menu			
	5	7	8	10
1	5	7	8	10
2	25	49	64	100
3	125	313	512	1000

*Dài thông và ảnh hưởng độc giả không chịu đựng thời gian trễ dài.*

Các nghiên cứu về con người cho thấy đối với phần lớn các công việc tính toán, ngưỡng của sự mất tác dụng là khoảng 10 giây. Các thiết kế trang web không thích hợp với tốc độ truy nhập mạng của độc giả sẽ gây ra sự

thất vọng. Đối với độc giả chỉ là một người dạo chơi web bình thường sử dụng modem tốc độ 28.8 kbps qua đường điện thoại, nếu đặt một ảnh lớn (kích thước) lên trang web, độc giả sẽ không đủ kiên nhẫn chờ đến khi ảnh được nạp xuống. Tuy nhiên, nếu chúng ta xây dựng một website cho nội bộ (intranet) trường học, doanh nghiệp, nơi mà mọi người truy nhập web server với tốc độ mạng LAN hoặc cao hơn, chúng ta nên sử dụng nhiều ảnh và multimedia.

### **3.5.4. Đơn giản và nhất quán**

Độc giả sẽ không ấn tượng với sự phức tạp không lý do, đặc biệt các độc giả phụ thuộc vào website của chúng ta về thời gian hoặc thông tin chính xác, liên quan đến công việc. Các biểu tượng nên đơn giản, quen thuộc sẽ dễ với độc giả, ví dụ nếu chúng ta muốn một biểu tượng cho việc thiết kế thông tin, nên chọn quyển sách hay thư viện, mà không nên chọn hình ảnh tàu vũ trụ hay vô tuyến truyền hình.

Adobe (do Studio Archetype thiết kế) là một mô hình xuất sắc cho thiết kế website. Các trang sử dụng đồ họa một cách rộng rãi với các trợ giúp định vị (navigation), áp dụng nhất quán ở tất cả các trang trên website. Một khi chúng ta biết các liên kết cơ bản nằm trên đầu trang, ranh giới trở thành vô hình và việc định vị sẽ dễ dàng hơn.

Để đạt được tối đa việc chức năng hóa và tính rõ ràng, thiết kế trang web nên được xây dựng trên các đơn vị mẫu nhất quán, tất cả đều sử dụng chung một kiểu đồ họa, phong cách biên tập và mô hình tổ chức. Mục đích là tính đồng nhất, có thể đoán trước, để độc giả cảm thấy thoải mái khi khai thác website của chúng ta, và chắc chắn họ biết tìm nội dung cần tìm. Các biểu tượng đồ họa trên trang web cung cấp các tín hiệu trực quan về sự tiếp nối của thông tin. Biểu tượng đồ họa trên tất cả đầu trang của Adobe site tạo nên một giao diện độc giả nhất quán và cũng là nét nhận dạng duy nhất cho site của Adobe.

Đối với các trang không sử dụng đồ họa nhưng sử dụng nhất quán các tiêu đề, các chân trang và các liên kết đến trang chủ, các trang liên quan sẽ thì vẫn tạo cảm giác cho độc giả là họ đang trong khung cảnh website của chúng ta.

Để tạo hiệu quả cho một hệ thống "không đứt đoạn - seamless", chúng ta nên xem xét đến việc đưa các thông tin quan trọng vào website, gộp nó vào hệ thống hơn là tạo liên kết đưa độc giả ra khỏi site của chúng ta (nếu không vi phạm bản quyền cho việc sao chép thông tin).

The screenshot shows the Adobe Illustrator website. At the top, there's a navigation bar with links for HOME, UP, MAP, INDEX, SEARCH, and PURCHASE. Below the navigation is a banner with the text "what's new" and "Adobe Illustrator". The main content area features a large image of a woman's face. On the left, there's a circular graphic with the text "6.0.1 update" and "Get it now!". Below this, there's a section titled "Now available: Adobe Illustrator(R) 6.0.1 update!" followed by a bulleted list of features. At the bottom of the page, there's a download link for the update.

Now available: **Adobe Illustrator(R) 6.0.1 update!**

Includes new plug-ins for greater file format versatility.

- Enhanced support for Photoshop saving options as "Rich-Ready" CDR files
- Native Macromedia FreeHand™ file format reader that reads version 5.0.1 creatures
- Enhanced Photoshop/EPS and TIFF file format plug-in support in these formats at a selectable resolution, including grayscale, when running in +Photoshop 3.0 mode
- Enhanced PDF plug-in that maintains layers, group, and workflow information when an Adobe Illustrator file is saved as a PDF file and then re-opened in Adobe Illustrator. This plug-in is supported only by version 6.0.1

Download the 6.0.1, Freehand, Photoshop, and TIFF formats plug-ins separately below. Or, register online at [www.adobe.com](http://www.adobe.com) and order the 6.0.1 update and get all three plug-ins on CD-ROM or download. You may only be shipped one download.

(Được thu nhỏ từ ảnh của [www.adobe.com](http://www.adobe.com))

### 3.5.5. Tính ổn định thiết kế

Để thuyết phục được độc giả rằng những thông tin mà chúng ta cung cấp là chính xác, đáng tin cậy, thì cần phải thiết kế website cẩn thận, giống như chúng ta tạo các mối liên hệ với các doanh nghiệp khác, đòi hỏi sự biên tập và các trình độ thiết kế cao. Một site trông luộm thuộm, với thiết kế trực quan nghèo nàn, trình độ biên tập kém sẽ không truyền được sự tin cậy cho các độc giả.

Tính ổn định chức năng trong thiết kế web có nghĩa là giữ các thành phần giao tiếp của website làm việc ổn định. Tính ổn định chức năng có hai thành phần - đặt các vật đúng chỗ ngay từ đầu khi thiết kế website, sau đó

giữ chúng hoạt động nhịp nhàng theo thời gian. Các website tốt tự nó đã có tác động qua lại với nhiều liên kết đến các trang trong site đó và có các liên kết đến các site khác. Trong khi thiết kế, cần kiểm tra thường xuyên các liên kết để đảm bảo chúng còn tồn tại, các nội dung trên web thay đổi rất nhanh, cả trên website của chúng ta và các website khác. Cần có lịch kiểm tra lại các liên kết và kiểm tra nội dung của nó có còn thích hợp không.

### ***3.5.6. Phản hồi và đối thoại***

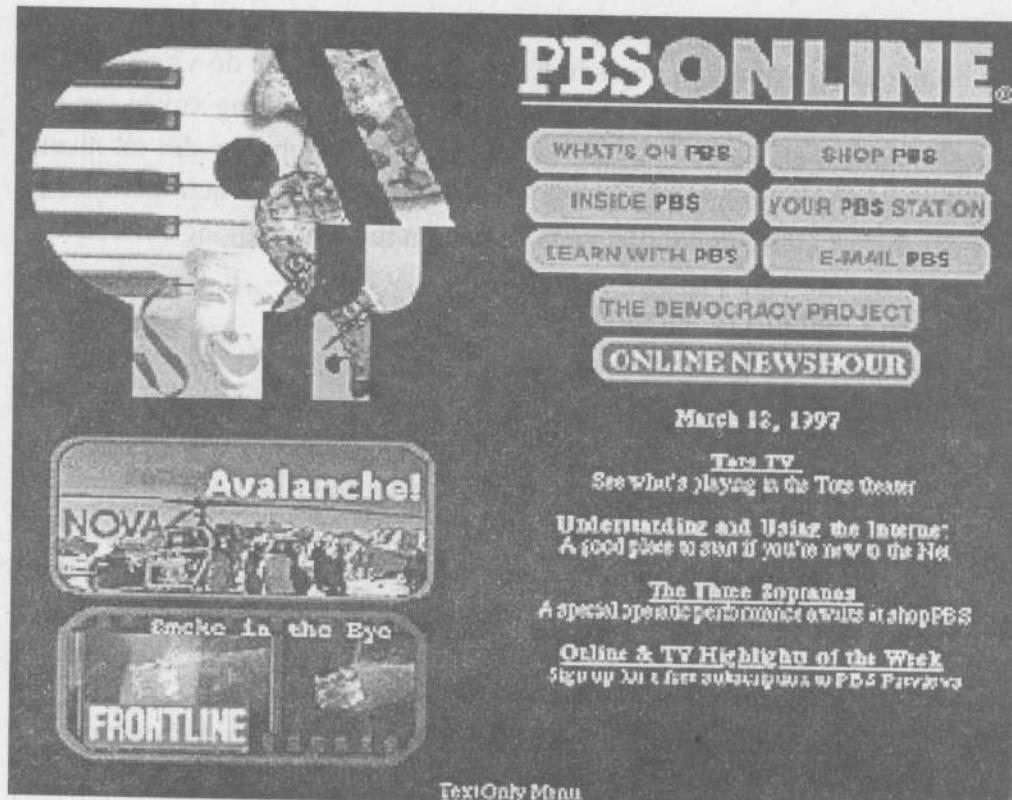
Thông qua hình ảnh đồ họa, các nút bấm, các liên kết đặt một nơi duy nhất, thiết kế web của chúng ta nên đưa ra khả năng xác nhận vị trí, lựa chọn của độc giả. Phản hồi cũng có nghĩa là bước chuẩn bị cho việc trả lời, đáp ứng các đòi hỏi, góp ý của độc giả. Các website thiết kế tốt luôn cung cấp mối liên hệ trực tiếp tới ban biên tập hoặc "webmaster" phụ trách kỹ thuật của site. Lên kế hoạch đảm bảo quan hệ liên tục với các độc giả là quan trọng sống còn đối với sự thành công lâu dài của một doanh nghiệp.

### ***3.5.7. Thiết kế cho các trình duyệt khác***

Không phải tất cả độc giả đều có thể xem được các ảnh đồ họa chúng ta đưa ra trên trang web, rất nhiều độc giả không có chương trình duyệt đồ họa. Một trong những ưu điểm của web và HTML là khả năng thay thế thông báo (nhận ALT trong HTML) để các độc giả với web browser không có khả năng đồ họa (lynx) vẫn hiểu được chức năng của hình ảnh trên trang web. Sử dụng chương trình đặc biệt, các độc giả khiếm thị có thể nghe các thông báo thay thế mà chúng ta cho hỗ trợ cùng hình ảnh đồ họa, do đó không hoàn toàn mất hẳn nội dung của bức ảnh, phím đồ họa đi kèm trang web.

### ***3.5.8. Điều khiển***

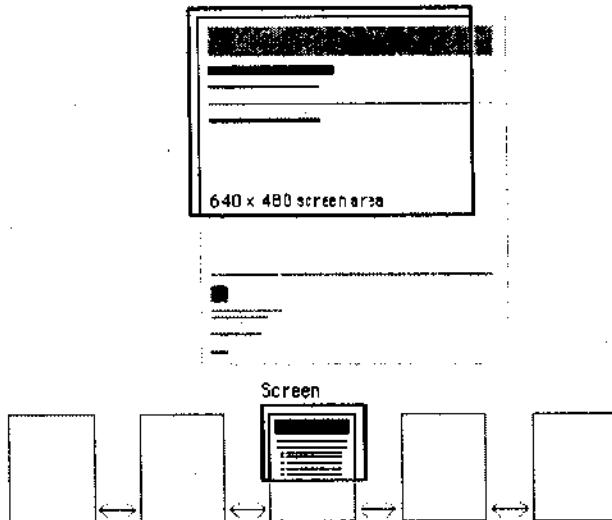
Việc cung cấp một tập hợp phong phú các điều khiển đồ họa và liên kết tương tác trong trang web sẽ thu hút sự chú ý của độc giả vào trang web, quên đi các liên kết đa năng của trình duyệt và lôi cuốn họ vào nội dung. Bằng cách sử dụng các phím án đồng nhất, có thể dự đoán được, chúng ta đã giúp độc giả có được cảm nhận của website của mình và làm cho logic, trật tự của website trở nên rõ ràng. Ví dụ sau đây từ trang chủ của PBS phong phú đồ họa, liên kết đã hấp dẫn ngay người đọc vào site này:



(Được thu nhỏ từ ảnh của [www.pbs.org](http://www.pbs.org))

### 3.5.9. Tạo ngữ cảnh hoặc mắt đọc giả

Độc giả cần cảm nhận ngữ cảnh, về vị trí của họ trong tổ chức thông tin. Trong các tài liệu trên giấy, cảm giác "ta đang ở đâu" là sự phối hợp các cách xử lý về biên tập, đồ họa có được từ thiết kế sách, cách tổ chức văn bản và cảm giác vật lý của cuốn sách. Các tài liệu điện tử không cung cấp một ám chỉ vật lý nào cho việc truy nhập thông tin. Khi thấy một liên kết web trên một trang, chúng ta có ít cảm nhận sẽ được dẫn đi đâu, có bao nhiêu thông tin ở đó, và chính xác thông tin đó quan hệ thế nào với trang hiện tại. Đại đa số các trang web không vừa khớp với màn hình 14-15 inch, do đó luôn có một phần của trang mà độc giả không thể nhìn thấy.



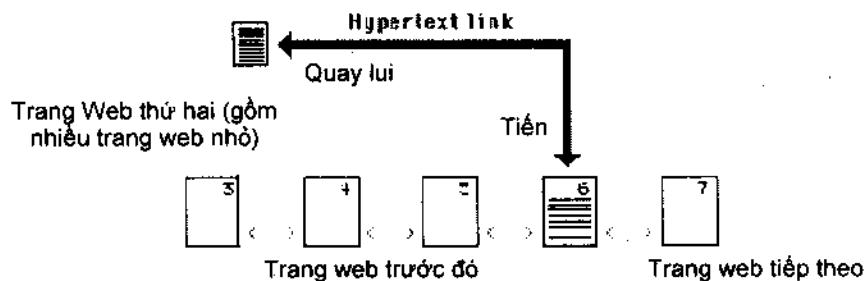
Các trang web cần cho độc giả cảm nhận rõ ràng ngữ cảnh và tổ chức thông tin vì chỉ có một phần nhỏ của Website (ít hơn một trang) được hiển thị vào một thời điểm.

Nếu là một người thiết kế web, chúng ta cần chuẩn bị để cung cấp cho độc giả các khả năng này.

### ***3.5.10. Liên kết và điều khiển "Quay lại" và quay về trang trước***

Tất cả các hệ thống siêu văn bản đều chia sẻ một vấn đề chung đó là: quay lại từ các liên kết mà độc giả đã đi qua sẽ không giống như lật các trang ngược lại trong các tài liệu in, khi mà các trang được đánh số. Khi độc giả nhấp vào một liên kết trong một tài liệu web, họ thường di chuyển từ một website này đến website khác, có khi từ một quốc gia này đến một quốc gia khác. Cũng do liên kết là hai chiều, độc giả có thể quay lại Website mà họ vừa rời khỏi bằng cách nhấp vào phím "Back" của trình duyệt hoặc nhấn phím "Forward" cho phép độc giả đi đến một website mới.

Trang Web đầu tiên

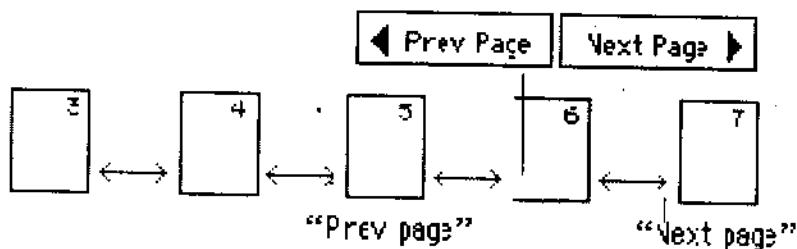


### **3.5.11. Tác dụng của thanh phím ấn**

Đối với các nhà thiết kế thông tin, các liên kết siêu văn bản là một giải pháp tuyệt vời nhưng tương đối phức tạp Cụ thể, sự thay đổi căn bản trong ngữ cảnh mà các liên kết tạo ra dễ gây lúng túng cho các độc giả, những người cần đến các gợi ý và các yếu tố tác động có tổ chức, khi họ di theo các liên kết siêu văn bản từ trang này sang trang khác.

Điều này đặc biệt đúng khi chúng ta đưa ra cho độc giả các tài liệu có nhiều phần đánh số. Trên hình trên, độc giả đã nhảy đến website thứ hai tại trang 6, và website đó gồm các trang được đánh số.

Bằng việc tăng thêm các phím chuẩn (Back và Forward) của trình duyệt với các phím "Trang trước" và "Trang tiếp" được đưa vào các trang, độc giả đã có thêm công cụ để định vị quan hệ thông tin của website như mong muốn. Thanh phím ấn cũng có thể hiển thị vị trí của thông tin, giống như tiêu đề chương của các cuốn sách in.



### **3.5.12. Liên kết cố định và tương đối**

Không như các phím Back hay Forward trong các trình duyệt Netscape và MS Internet Explorer, Mosaic chỉ có chức năng tương đối đến các trang mà độc giả vừa rời khỏi, phím Trang tiếp và Trang trước trong các trang web là các liên kết cố định do chúng ta tạo nên chỉ đến một văn bản nhất định. Bằng cách tạo các phím lật trang, phím chỉ đến mục lục, chúng ta đã cung cấp cho độc giả phương tiện để hiểu cách thức tổ chức thông tin trên website, ngay cả khi họ đến không phải từ trang chủ, hoặc từ mục lục nội dung. Phím ấn không cho phép độc giả đọc tin theo thứ tự họ chọn, nhưng cho phép đọc các trang liên tiếp.

## **4. THIẾT KẾ WEBSITE**

### **4.1. Khái quát**

Tồn tại những lý thuyết có hệ thống và cơ bản cho việc chia nhỏ bất cứ khối thông tin lớn nào, cho dù nó được đem đi in hay cho World Wide Web. Cơ sở của mọi hệ thống lý luận là giới hạn của con người trong việc lưu giữ và nhớ lại thông tin. Các nhà tâm lý học nhận thức đã biết từ hàng thập niên trước rằng đại đa số chúng ta chỉ có thể lưu giữ khoảng 4 đến 7 mẩu thông tin rời rạc trong trí nhớ ngắn hạn. Mục đích của các hệ thống tổ chức là giữ số lượng mẩu thông tin mà người đọc cần lưu nhớ đến tối thiểu, bằng việc sử dụng kết hợp giữa thiết kế đồ họa, quy ước lớp và biên tập thông tin thành các đơn vị riêng rẽ. Phương pháp độc giả tìm kiếm và sử dụng thông tin cũng thừa nhận, các tin ngắn hơn, riêng biệt sẽ chức năng hoá hơn và dễ định vị hơn khỏi thông tin dài.

Đại đa số website có thông tin tham khảo để độc giả tìm kiếm thường trong các bài viết ngắn. Độc giả rất ít khi đọc các tài liệu dài, liên tục trên màn hình, và những người đi tìm một mẩu tin, sẽ khó chịu khi phải rà soát một bài viết toàn chữ để tìm tài liệu họ cần. Các đoạn tin nhỏ của các thông tin có liên quan sẽ dễ tổ chức hơn thành các khối thông tin riêng để tạo nên hệ thống đồng nhất, hình thành nên cơ sở các liên kết hypertext. "Nhỏ" chỉ có thể được xác định trong ngữ cảnh của tài liệu chúng ta trình bày và cái ta mong muốn đem lại cho độc giả.

### **4.2. Các bước trong tổ chức thông tin**

Cuộc sống xã hội và công nghiệp ngày càng ít đòi hỏi tạo lập bản nội dung chi tiết về những điều chúng ta biết và những mối liên quan giữa chúng, nhưng không có một nền tảng hệ thống hoá logic và vững chắc, website của chúng ta sẽ không hoạt động tốt, ngay cả khi các nội dung cơ bản là chính xác và lôi cuốn. Bốn bước cơ bản trong việc tổ chức thông tin là chia nó thành các đơn vị logic, thiết lập hệ thống cấp bậc theo tầm quan trọng và tính tổng quát, sử dụng hệ thống này để tạo cấu trúc quan hệ giữa chúng, sau đó phân tích sự thành công về chức năng và thẩm mỹ của các hệ thống.

### **4.3. Cắt đoạn thông tin**

Đa số thông tin trên World Wide Web gồm có các bài giới thiệu ngắn không cần đọc nối tiếp. Điều này rất đúng đối với các website của các tổ chức, chính phủ, doanh nghiệp và giáo dục hay cung cấp các thông tin đã được in trên giấy trước đó. Những nhã viết tài liệu kỹ thuật đã phát hiện rất lâu trước khi web ra đời rằng độc giả đánh giá cao các đoạn thông tin ngắn do có thể nhanh chóng rà soát và định vị chúng. Các đoạn thông tin ngắn, tổ chức thống nhất đặc biệt thích hợp với trình bày của web, do có ít độc giả dành thời gian để đọc tài liệu dài trên màn hình.

Đa số họ sẽ lưu tài liệu dài vào đĩa, hoặc in chúng, chỉ đọc những gì bao quát trực tuyến. Các mẫu tin ngắn, riêng biệt thích hợp với liên kết web. Độc giả thường muốn tìm thấy một phần thông tin chủ định, mà không phải toàn bộ cả quyển sách để từ đó lọc ra. Nhưng nếu thông tin bị chia cắt quá nhỏ, ta sẽ làm độc giả thất vọng. Từ một đến ba trang (in) thông tin là đủ cho một đoạn thông tin trên web. Một liên kết mà chỉ dẫn đến một mẫu tin cụt lùn sẽ không hợp lý.

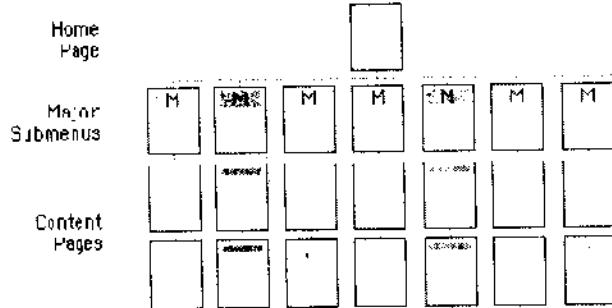
Hình thức đồng nhất của cách tổ chức và hình thức trình bày thông tin cho phép độc giả áp dụng kinh nghiệm của họ có từ website của chúng ta để tìm kiếm, khám phá, và cũng cho phép độc giả dự đoán được phần website mới, lạ sẽ được tổ chức như thế nào. Các đoạn thông tin ngắn gọn, súc tích sẽ thích hợp hơn với màn hình máy tính, có tầm nhìn hạn chế với các văn bản dài.

Việc áp dụng phân chia thông tin phải linh động, và nhất quán với ý thức chung, với hệ thống logic và với sự thuận tiện cho độc giả web. Cách tốt nhất để phân chia và tổ chức thông tin thực hiện theo bản chất của nội dung. Cũng có lúc cần tạo một tài liệu dài trên web như một bản tóm hợp của các đoạn thông tin. Điều này cần thiết khi chúng ta tạo các trang web để độc giả có thể lưu hay in chúng.

### **4.4. Hệ thống phân cấp**

Nếu chỉ xác định cấu trúc định vị cơ bản cho độc giả thì mọi tổ chức cần đến sự phân cấp theo tầm quan trọng. Mọi "đoạn" thông tin có thể và nên được sắp xếp theo mức quan trọng và được hệ thống theo mức độ quan hệ giữa các thành phần. Khi đã xác định được hợp lý các mức ưu tiên,

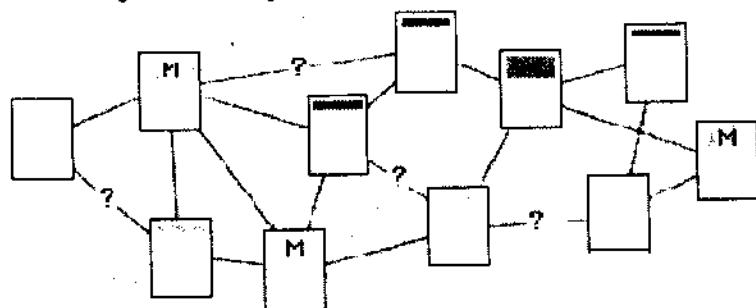
chúng ta có thể xây dựng một hệ thống phân cấp từ mức ưu tiên nhất hay mức tổng quát nhất, xuống đến mức cụ thể nhất hay mức chi tiết nhất. Hệ thống phân cấp thực sự là cần thiết đối với web, vì ý tưởng trang chủ - liên kết phụ thuộc vào sự phân cấp, di chuyển từ cái nhìn khái quát nhất của toàn website (trang chủ), qua các menu con xuống đến trang nội dung đã ngày càng trở nên đặc trưng.



#### 4.5. Các mối quan hệ

Khi đối diện với một hệ thống thông tin mới, phức tạp, độc giả bắt đầu xây dựng các mô hình lý trí, và sau đó sử dụng chúng để đánh giá các mối liên hệ giữa những chủ đề, và giả thiết về vị trí tìm thấy thông tin họ chưa thấy trước đó. Sự thành công của website chúng ta như một hệ thống thông tin sẽ chủ yếu được xác định bởi hệ thống đó cân xứng bao nhiêu với các mong muốn của độc giả. Hệ thống hợp lý cho phép độc giả dự đoán đúng vị trí họ tìm thấy thông tin mong muốn. Thông tin được sắp xếp bằng hình ảnh, dán nhãn, được nhóm và phân loại nhất quán cho phép độc giả mở rộng nhận biết từ các trang đã xem vào các trang mới lạ. Nếu cấu trúc không hợp lý hoặc phức tạp, độc giả sẽ gặp khó khăn khi tìm thông tin. Không nên xây dựng một website trong như sau:

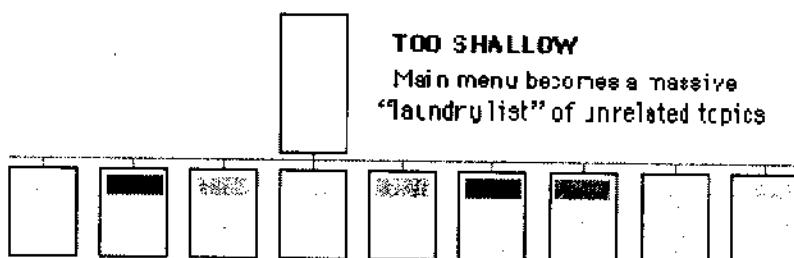
**Confusing “user image”**



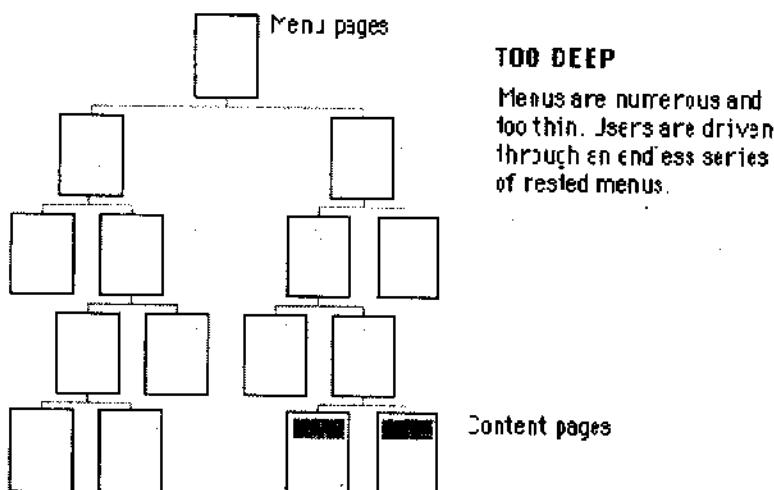
#### 4.6. Chức năng

Sau khi đã tạo nên website, chúng ta nên phân tích tính thẩm mỹ, tính thực tế cũng như tính hiệu quả của cả cơ cấu hệ thống. Việc chọn cấu trúc hệ thống nào cho website không quan trọng, quan trọng là thiết kế website thích hợp, cân bằng giữa cấu trúc và quan hệ của menu hay các trang "homepage" với các trang nội dung độc lập, các đồ họa liên kết, tài liệu. Mục đích là để xây dựng một hệ thống phân cấp của menu, trang web sao cho tự nhiên đối với độc giả, không gây trở ngại hoặc làm lúng túng khi đọc website.

Các website quá nông, chỉ có một mức liên kết, uỷ thác vào các trang menu nặng nề sau một thời gian sẽ tạo thành một mớ hỗn độn các thông tin không liên quan đến nhau, được liệt kê không theo một trật tự nào.



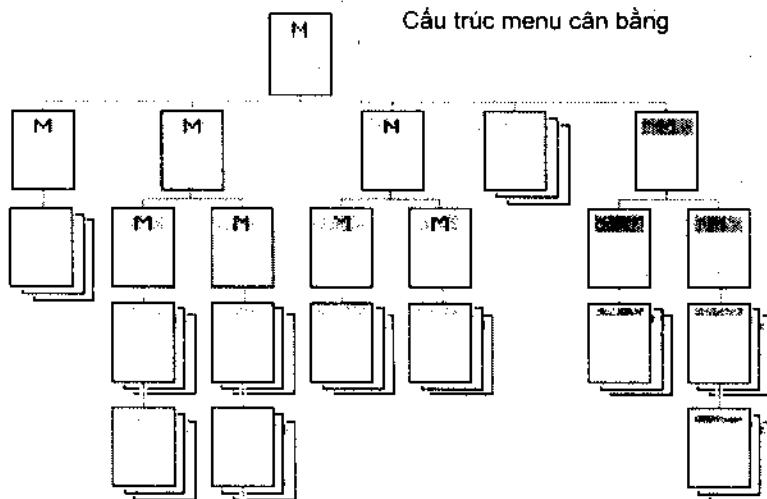
Hệ thống menu lại có thể quá sâu, cắt giấu thông tin dưới nhiều lớp menu:



Các Gopher site là ví dụ điển hình cho những bất tiện của các menu lồng nhau, nơi mà chúng ta đôi khi phải mở nhiều thư mục trước khi gấp các

tài liệu. Menu mất giá trị khi chúng không chuyển tải từ 4 đến 5 liên kết; các trang menu dựa trên danh sách, bằng chữ có thể tải rất nhiều liên kết mà không "dìm sâu" độc giả hay bắt họ cuộn màn hình qua một danh sách dài.

Sẽ không hợp lý khi bắt độc giả định vị qua nhiều mức menu lồng nhau trước khi đọc được thông tin cần thiết. Nếu website phát triển nhanh, sự cân bằng các menu và trang web là rất cần thiết. Các phản hồi của độc giả (và sự phân tích việc sử dụng website) có thể giúp chúng ta quyết định được cấu trúc website có còn thích hợp không, hay có phần nào thiết kế kém không. Các tài liệu phức tạp đòi hỏi sự phân bổ menu sâu, nhưng xu hướng độc giả không muốn truy cập vào các trang toàn menu nếu có khả năng truy nhập trực tiếp. Mục tiêu là tạo một cây phân lớp thích hợp có khả năng truy nhập thông tin nhanh và giúp độc giả hiểu được thông tin được tổ chức như thế nào.



#### 4.7. Cấu trúc site

Với những người quan tâm, chú ý đến World Wide Web, các khái niệm hypertext, hypermedia hoàn toàn không xa lạ. Ngày nay, chế biến máy tính dày ấp những ý tưởng về việc thông tin trên web có thể bằng cách nào đó liên kết mọi thứ với nhau. Với những gợi ý này, bạn có thể bò qua một trong những thách thức lớn nhất của việc trình bày thông tin - đặt thông tin vào trật tự logic như thế nào và tạo một website dễ hiểu, đáng quan tâm cho độc giả.

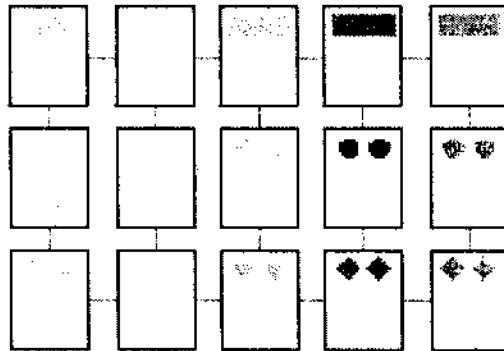
Nếu không có một cái nhìn tổng thể hoặc ý thức rõ ràng về tổ chức, liên kết trong website, độc giả của chúng ta sẽ sớm nhận ra và đại đa số sẽ bỏ đi để tìm thông tin khác tốt hơn.

*Sự nối tiếp* : Cách đơn giản nhất để hệ thống thông tin là theo dãy, với cách đó, chúng ta có thể hiển thị thông tin một cách tuần tự. Thông tin sẽ tiếp nối nhau như một bản tường thuật, theo thời gian, hoặc trong sự sắp xếp logic nó là ý tưởng cho sự luận bàn nối tiếp. Sắp xếp tuần tự có thể theo thứ tự thời gian, ví dụ như một chuỗi logic các chủ đề được phát triển từ tổng quát đến cụ thể, hoặc cũng có thể theo thứ tự abc, như các chỉ số, tự điển bách khoa, tự điển thuật ngữ. Tuy nhiên, cách tổ chức này chỉ thích hợp với các website nhỏ (hoặc các danh sách cấu trúc như chỉ số), các chuỗi càng dài càng trở nên phức tạp hơn, và khi đó càng cần có cấu trúc hơn để vẫn giữ được tính dễ hiểu.

Nhiều website lớn vẫn còn được tổ chức kiểu nối tiếp, nhưng mỗi trang trong chuỗi chính có thể có một hay nhiều trang nói ngoài đề, thông tin chen giữa, hoặc các liên kết đến các website khác.

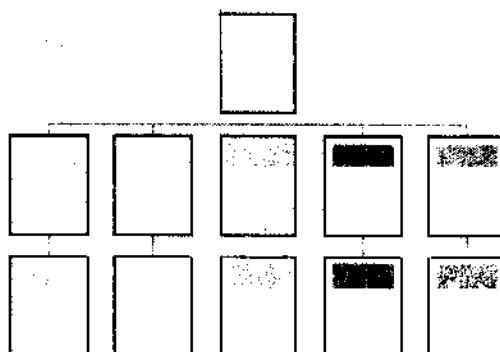


*Ô lưới* : Nhiều bản hướng dẫn, danh sách các khoá học của trường đại học hoặc các giải nghĩa cho các khái niệm kỹ thuật được tổ chức tốt nhất theo phương thức ô lưới. Nó là cách tốt để tương quan các biến số như sự kiện, công nghệ, văn hoá,... Để thành công, từng đơn vị riêng biệt trong lưới nhất định phải có cùng cấu trúc cho các chủ đề lớn và nhỏ. Các chủ đề thường không có sự phân cấp về mức độ quan trọng. Ví dụ, "TCP/IP" cũng quan trọng không hơn, không kém so với "IPX/SPX", do vậy cả hai mô tả nên có cùng cấu trúc. Như vậy độc giả có thể đi tiếp (đọc xuống lưới) để đọc diễn giải về "TCP/IP", hay đi ngang (đi ngang lưới) bằng cách đọc phần "packet" của cả hai chủ đề TCP/IP và IPX/SPX. Nhuộc điểm của tổ chức lưới có thể khó hiểu với những độc giả khi chưa xác định được mối liên quan giữa các loại thông tin, nhưng sẽ rất phù hợp với những độc giả có kinh nghiệm, những người đã có sẵn kiến thức về chủ đề và hệ thống của nó. Các sơ đồ tổng quát thường rất hữu ích đối với các site kiểu lưới.



#### 4.8. Phân cấp

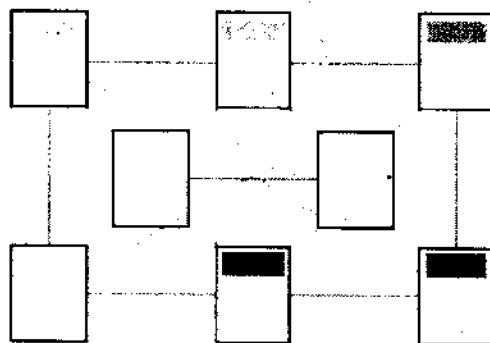
Sự phân cấp thông tin là một trong những cách tốt nhất để tổ chức các khối thông tin phức hợp. Sắp xếp có phân cấp đặc biệt thích hợp cho các website, vì các website luôn được thực hiện theo cách rẽ nhánh từ một trang chủ duy nhất. Đa số độc giả quen thuộc với các biểu đồ phân cấp, và các ví dụ sẽ trở nên dễ hiểu hơn. Một tổ chức phân cấp cũng tác động có ích đến bản thân việc phân tích nội dung website do sự phân cấp chỉ hoạt động hiệu quả khi đã tổ chức hoàn hảo nội dung của chúng ta. Do biểu đồ phân cấp rất giống với cấu trúc của các doanh nghiệp, viện nghiên cứu độc giả dễ dàng xây dựng mô hình lý trí của cả website:



#### 4.9. Web (Mạng)

Cấu trúc tổ chức giống mạng nhện yêu cầu ít hạn chế cho việc sử dụng mẫu thông tin. Mục đích thường cho ý tưởng liên kết giống nhau và tự do, nơi mà độc giả đi theo sự quan tâm của họ trong một mô hình tự khám phá, tự do tư tưởng đối với từng độc giả đến website. Mô hình website này gồm nhiều các liên kết đến các tài liệu ở trong website đó cũng như trên World

Wide Web. Mục đích là khai thác triệt để năng lực của web trong việc liên kết và kết hợp, tuy vậy các cấu trúc theo kiểu mạng nhện này rất dễ phát triển thành một mớ hỗn độn, lộn xộn các khối thông tin.

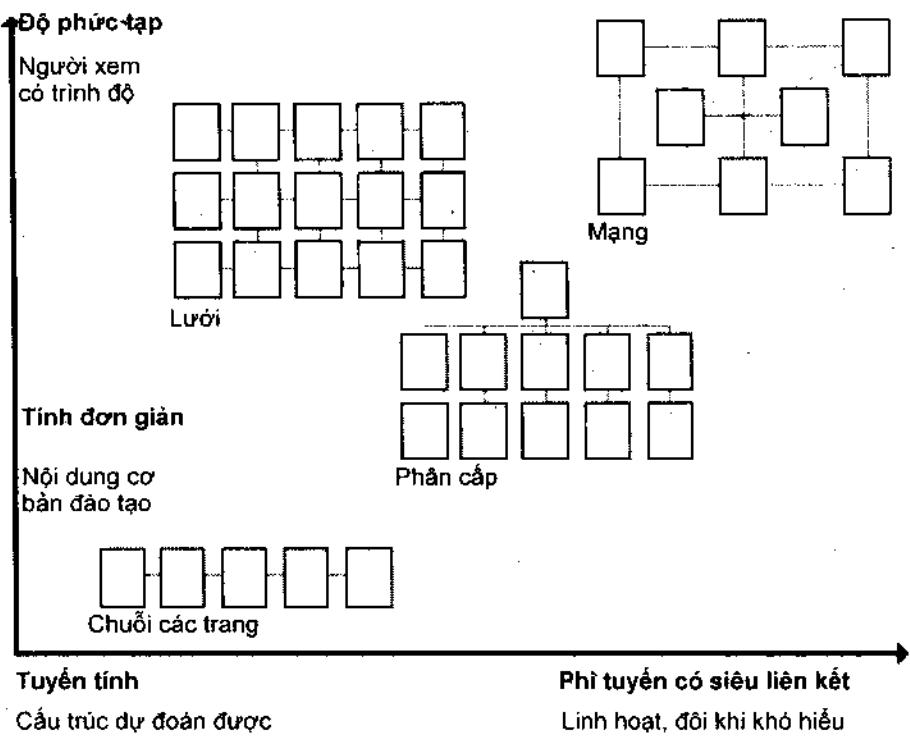


Tuy vậy, các web có tổ chức lại thường có cấu trúc phi thực tế nhất đối với các website, vì nó rất khó hiểu, dự đoán đối với độc giả. Các web làm việc tốt nhất cho các site nhỏ, có nhiều danh sách liên kết, nhắm vào các độc giả chuyên nghiệp hoặc trình độ cao, những độc giả tìm kiếm những kiến thức chuyên sâu.

#### 4.10. Kết luận

Đa số các website phức tạp đều sử dụng cả bốn kiểu cấu trúc thông tin trên. Ngoại trừ các site mà có yêu cầu khắt khe về việc phải hỗ trợ các trang hiển thị nối tiếp, độc giả thích sử dụng các website theo kiểu tự do như "mạng", như đa số các sách tra cứu hay kỹ thuật thường sử dụng. Tuy nhiên, điều này cũng không tháo gỡ cho người thiết kế web khỏi sự cần thiết phải hệ thống hóa suy nghĩ và trình bày nó một cách minh bạch, nhất quán, đó là những yếu tố cần thiết hỗ trợ cho các mục đích của website. Sau đây là biểu đồ thể hiện mối tương quan giữa bốn kiểu cấu trúc và mục đích, sự phức tạp của nội dung.

Các Website có thể vô cùng đa dạng về kiểu cách, nội dung, cách tổ chức, và mục đích, nhưng tất cả các website đều được thiết kế để thực hiện chức năng của chúng là các nguồn tài nguyên cùng chia sẻ những đặc điểm cơ bản.



## 5. TRANG CHỦ (Home Page)

Tất cả các website đều được thiết lập quanh "home page" giữ nhiệm vụ như một điểm xuất phát đến các trang web phức tạp khác trong website. Trong hệ thống phân cấp, trang chủ chiếm vị trí trên đỉnh của sơ đồ. URL WWW của trang chủ cũng là địa chỉ web được sử dụng để hướng dẫn độc giả đến với website của chúng ta. Địa chỉ của trang chủ quan trọng cũng như địa chỉ nhà riêng hay địa chỉ của một doanh nghiệp. Đầu trang chủ sẽ là nội dung đầu tiên độc giả nhìn thấy khi truy nhập đến website, do đó trang chủ được thiết kế thích hợp là điều kiện cơ bản để website thành công. Chiến lược thiết kế trang chủ rất biến hóa, dựa trên chức năng và yêu cầu của các độc giả đặc trưng của website, mục đích của website và cũng phụ thuộc vào tính chất, sự phức tạp của toàn bộ website.

### *Menu đồ họa hay văn bản*

Quyết định cách bố trí cơ bản nhất cho trang chủ liên quan đến việc sử dụng đồ họa như thế nào trên trang web. Đa số các trang chủ của các doanh

nghiệp, viện hay trường hiển thị ít nhất một thanh ảnh nhỏ (banner) trên đầu trang, và trên các trang thương mại, xu hướng lại là ảnh đồ họa toàn trang, phức tạp, được xây dựng từ các menu thay đổi liên tục tạo nên cảm giác như các trang multimedia của CD-ROM. Trang web nhiều ảnh đồ họa có thể có hiệu quả để gây sự chú ý của độc giả, nhưng lại tốn thời gian nạp xuống, đặc biệt khi độc giả kết nối với Internet qua modem hay mạng trực tốc độ thấp. Ngay cả khi độc giả truy nhập website với tốc độ Ethernet (10 Mbps), các menu đồ họa cũng chậm hơn danh sách các liên kết trên cơ sở text đến 10 lần.

### *Ai là độc giả của trang chủ ?*

Việc phân cách giữa các trang chủ trang trí đồ họa hấp dẫn, nạp xuống chậm với các trang text đơn thuần, nhưng lại nạp nhanh đã phản ánh sự cần thiết phân biệt sự khác biệt đối tượng độc giả cùng nhu cầu của họ. Mục đích của đa số website là truyền tải thông tin nội tại (cho sinh viên, nhân viên và các khách hàng) và giao thiệp với các khách hàng quan trọng, nhưng một maket được thiết kế cẩn thận không chỉ mở rộng khả năng của một màn hình văn phòng. Bằng cách giữ ảnh đồ họa vừa phải về kích thước, trang web được nạp xuống không chậm lăm mà vẫn có menu đồ họa.

Trang chủ của W3C khá đơn giản, chủ yếu là menu text cung cấp một tỷ lệ hiệu quả giữa số kết nối với kích thước trang và với một giao diện thuần khiết. Trang web nạp nhanh, được thiết kế thích hợp cho các độc giả web đặc biệt, nhưng lại không hấp dẫn đối với các độc giả bình thường.



(Được thu nhỏ từ trang chủ lấy từ web [www.kodak.com](http://www.kodak.com))

*Realizing the Full Potential of the Web...*

### Announcing First Public Release of Amaya

"The Amaya client, like the Jigsaw server, is a tool for experimenting to find out what is possible and demonstrators what can be done. Experiments, tests and demonstrations of developments in HTML, CSS, HTTP are examples of the way Amaya has already been used to great effect. Amaya provides focus for the community to come to consensus on implementable, practical standards."

-- Tim Berners-Lee, W3C Director

*- Trial of W3C's HTTP 1.1, CSRI and PNG Can Make the Web As Much As 2-8 Times Faster*

### User Interface

Help  
Site Map  
Getting Started  
Documentation  
FAQs  
About  
Search

### Technology and Society

Accessibility  
Data Interoperability  
Innovation  
Electronic Commerce  
Policy  
Universal Design

(Được thu nhỏ từ trang chủ tại [www.w3.org/pub/WWW/](http://www.w3.org/pub/WWW/))



*is the part of Thomas Jefferson, a version of the U.S. Congress Library.*

**▼ NEW**

105th Congress:  
House Members  
Senate Members

Interested in participating in a variety of Government web site?

**▼ GO TO**

About THOMAS

**CONGRESS THIS WEEK**

Today's Events House of Representatives  
**BILLS**

Major Legislation:  
105th Session - Executive Orders - Executive Law  
104th Session - Executive Orders - Executive Law  
Bill Summaries & Status: Congress: H (1997-98) · 104 (1995-96)  
Bill Text: Congress: H (1997-98) · S (1995-96) · H (1993-94)  
Public Law by Law Number: H (1997-98) · S (1995-96)

**CONGRESSIONAL RECORD**

Congressional Record Text: Congress

(Được làm từ ảnh tại Website: <http://thomas.loc.gov>)

## 5.1. Sử dụng cả hai cách tiếp cận

Cách tốt nhất đáp ứng đòi hỏi của cả độc giả ngẫu nhiên và độc giả thường xuyên, yêu cầu cao là cung cấp cả các trình bày thay thế của website. Một cách tiếp cận là cung cấp một trang chủ đầy ấn tượng cho các độc giả chung, nhưng cũng có một trang chủ thay thế, chủ yếu là text, chú trọng về tốc độ truy nhập thông tin qua các menu chi tiết. Cách tiếp cận khác là sử dụng banner đồ họa trên đầu trang, sau đó là các liên kết text. Website "Thomas" của Library of Congress phản ánh cả hai cách tiếp cận, với banner đồ họa vừa phải và các liên kết phong phú, tổ chức tốt.

## **5.2. Thời gian quản lý**

Nhiều website cần được cập nhật thường xuyên để thông tin không bị lạc hậu. Nhưng không phải tự nhiên mà độc giả cảm nhận được thông tin mới nếu chúng ta không bô công sức ra để làm cho độc giả biết đến nó. Nếu một chức năng nào đó được cập nhật, chúng ta nên đặt một ký hiệu "Mới" lên cạnh nó. Chúng ta cũng nên ghi thời gian lên từng trang web, cả thời gian cập nhật để độc giả được đảm bảo là nhận thông tin mới nhất. Tuy nhiên, nếu website phức tạp, nhiều mức thông tin, trải ra hàng chục (thậm chí hàng trăm, nghìn) trang web, tốt hơn hết là chúng ta nên làm một trang "Có gì mới - What's New" được thiết kế đặc biệt dành cho việc thông báo cho độc giả sự thay đổi trên website.

## **5.3. Trang chủ, menu và các submenu**

Khi website nhỏ, có thể chỉ cần một ít trang menu con để độc giả đi tiếp từ danh sách các chủ đề chung. Trong các website lớn, phức tạp với hàng chục chủ đề, thì không thực tế khi đưa cho độc giả một trang chủ với rất nhiều liên kết - một trang quá dài để nạp xuống, các trang này sẽ làm nản lòng độc giả.

Mỗi submenu lớn có thể trở thành một trang chủ nhỏ cho khu vực đó của website. Đối với các menu chi tiết, đặc biệt chúng ta có thể khuyến khích các độc giả thường xuyên kết nối trực tiếp vào các submenu trong website. Khi đó, các submenu này có thể trở thành các trang chủ dành riêng cho các nhóm độc giả nhất định.

Chỉ cần đảm bảo là trên từng submenu, chúng ta thêm các liên kết cơ bản đến các khu vực khác của website và quan trọng nhất là, thêm liên kết đến trang chủ hay menu chính trên mọi trang web.

## **5.4. Bản kê "các site liên quan khác"**

World Wide Web thay đổi nhanh chóng khiến cho ngay cả các dịch vụ tìm kiếm web không lồ như Yahoo cũng chỉ là một phần thông tin có thể truy nhập trên web. Thông thường, tập hợp đầu tiên của các liên kết khi xây dựng website là bộ sưu tầm các website ưa thích, có liên quan đến công việc, nghề nghiệp hay sở thích các nhân. Tuy nhiên, trong các website của các doanh nghiệp lớn, viện nghiên cứu thì trang tham khảo được biên tập tốt có thể sẽ là

trang có giá trị nhất, là nguồn tài nguyên được sử dụng nhiều nhất trong website.

### 5.5. Thư viện, phụ lục, chỉ số

Khái niệm tài liệu, văn bản trong môi trường điện tử như trang web thường linh động, và việc xuất bản điện tử về mặt logic, kinh tế đã tạo khả năng cung cấp nhiều thông tin hơn cho độc giả mà không phải trả giá như tài liệu giấy. Để làm một bản báo cáo nhân sự trên giấy, chúng ta cần in bản sao từng nhân sự một. Yêu cầu bản báo cáo phải ngắn gọn, súc tích, không có nhiều thông tin hỗ trợ hay phụ lục. Những hạn chế này không còn đối với các trang web, bên cạnh phần chính của bản báo cáo, chúng ta có thể thêm danh sách các nguồn thông tin cần thiết khác. Thư mục, phụ lục, từ điển thông tin có thể rất lớn để đi kèm với bản báo cáo, hoặc đặt lên web, tạo điều kiện cho các độc giả khác tìm hiểu mà không cần đọc qua bản báo cáo bao gồm những phần không cần thiết với họ.

### 5.6. Các vấn đề thường gặp - các trang FAQ

Web và môi trường trên cơ sở web đã mở ra một trang FAQ, đó là nơi các vấn đề, câu hỏi mà độc giả thường đặt ra được liệt kê cùng các câu trả lời. Các trang web FAQ lý tưởng cho việc thiết kế website hỗ trợ, cung cấp thông tin trong nhóm làm việc, cho nội bộ doanh nghiệp. Với một trang FAQ được thiết kế tốt, mà khi được tham chiếu đến đó, sẽ giúp cho độc giả hiểu thông tin và các dịch vụ chúng ta cung cấp qua website. Trang FAQ cũng giúp giảm bớt nhân sự vào việc hỗ trợ khách hàng, những người thường xuyên chỉ trả lời các câu hỏi lặp lại, thường ngày của khách hàng, độc giả.

## 6. THIẾT KẾ INTRANET

Đa số các website được thiết kế cho các độc giả trong học viện, công ty và không cung cấp ra ngoài thế giới World Wide Web. Các site intranet chia sẻ cùng công nghệ với các site được thiết kế cho độc giả web trên Internet, tuy vậy việc thiết kế và nội dung của intranet cần phản ánh được những tác động đa dạng của các độc giả intranet.



## **6.1. Các site bên ngoài**

Các site bên ngoài thường có mục đích thu hút độc giả. Mục đích chung nhất là nâng cao thời gian truy nhập, dẫn dắt độc giả vào sâu trong site và đem lại cho độc giả những thông tin hấp dẫn, giải trí. Đối với các website bên ngoài, độc giả ít có động lực để lưu tâm, do vậy để thu hút độc giả, cần sử dụng đồ họa đẹp, thôi thúc họ tìm hiểu, khám phá website.

## **6.2. Các site bên trong**

Các intranet site thành công là thu thập được nhiều thông tin hữu ích, được tổ chức trong các hệ thống logic và cung cấp nó theo một cách thức có hiệu quả. Chúng ta không thể mong muốn độc giả intranet nán ná trên các trình duyệt web hay lúng túng với việc không thể tìm thấy cái họ tìm, hoặc chỉ dạo chơi trên mạng cục bộ mà không giải quyết được việc gì. Hãy để độc giả (trong trường hợp này là các nhân viên, học viên,...) nhận được thông tin cần tìm nhanh chóng.

## **6.3. Các tiêu chuẩn thiết kế**

Các Internet và intranet site của công ty Sun Microsystem là những mô hình thiết kế web nhất quán, tiếp cận theo chiều sâu. Các câu hỏi điều tra cho thấy, các nhân viên của công ty Sun trung bình đọc 12 trang intranet mỗi ngày, và khoảng 2 sub-site intranet mới mỗi tuần. Jakob Nielsen, chuyên gia thiết kế giao diện người dùng của Sun, ước tính rằng việc thiết kế lại giao diện Intranet của Sun, có thể tiết kiệm cho mỗi nhân viên 5 phút/1 tuần thông qua các giao diện chuẩn và các ứng dụng thống nhất toàn công ty. Tổng số thời gian nhân viên của Sun tiết kiệm có thể lên đến 10 triệu USD mỗi năm, nhờ tránh được các việc vô ích, nâng cao hiệu quả do mỗi nhân viên sử dụng intranet của công ty.

## **6.4. Các tiêu chuẩn thiết kế**

Tất cả các đơn vị ứng dụng intranet đều có những động lực về xã hội và kinh tế để phát triển và truyền bá các kinh nghiệm về thiết kế các website nội bộ và nguồn thông tin của họ. Nhưng cũng có nhiều vấn đề vẫn đề này sinh khi các nhóm, cá nhân muốn có quyền được thiết kế, xuất bản theo ý muốn và thường là có kinh nghiệm, thành thạo sử dụng hơn là quản lý nó. Các nhóm sử dụng web nhiều năm cũng có sự đầu tư đáng kể vào những

thiết kế của chính họ và sẽ rất miễn cưỡng nếu phải thay đổi. Các quản trị viên hệ thống lại thường eo hẹp về tài chính để viết các bản hướng dẫn sử dụng, hướng dẫn thiết kế và thúc đẩy được các phòng, ban hành chính ban hành chúng. Không có một tiêu chuẩn chung mang tính quốc gia hay quốc tế về cái gì là phần tử làm nên một website tốt, điều này càng làm cho vấn đề trở nên phức tạp hơn.

### 6.5. Thiết kế coi trọng người đọc

Các vấn đề nêu trên rất quen thuộc với mọi webmaster và cho bất cứ ai đã từng ngồi trong ban chấp hành intranet, website. Không có sự thiết kế cẩn trọng, thống nhất, người sử dụng sẽ lạc vào mê cung, không có hiệu quả, cũng không có cơ hội để thu lợi từ công nghệ intranet. Nếu lựa chọn cách tiếp cận coi người sử dụng là trung tâm của mọi thiết kế, thiết kế nhất quán rõ ràng, sẽ thuyết phục được các sở thích đặc biệt của các bộ phận, cá nhân tham gia. Chúng ta nên luôn nhớ một điều là nếu một người sử dụng đặc trưng của intranet thấy nó hỗn độn hơn là thấy một nguồn thông tin hữu ích thì sẽ chẳng có một ai thu lợi được từ intranet đó cả.

Không có tiêu chuẩn thiết kế rõ ràng, intranet sẽ phát triển thành tập hợp các trang chắp vá - lúc thiết kế rất tốt, lúc lại rất lộn xộn và tất cả cấu thành một hệ thống rời rạc hay hoàn toàn tê liệt. Thiếu tiêu chuẩn thiết kế cũng giới hạn việc sử dụng intranet do phải tuân thủ hàng loạt quy tắc phức tạp, đối với người dùng mới muốn phát triển các site intranet lại phải đổi diện với các công việc như phát triển đồ họa, thiết kế giao diện thay vì đơn giản chỉ áp dụng những cái đã có.

# *Chương 6*

## **THƯƠNG MẠI ĐIỆN TỬ**

### **1. KHÁI NIỆM VÀ ĐẶC TRƯNG CỦA THƯƠNG MẠI ĐIỆN TỬ**

#### **1.1. Khái niệm thương mại điện tử**

Thương mại điện tử sẽ trở thành một nhân tố kinh tế có ý nghĩa toàn cầu. Cơ sở hạ tầng của thương mại điện tử là mạng máy tính khi nó đã đạt đến trình độ tiêu chuẩn cho hoạt động kinh doanh, đời sống gia đình và hoạt động của các chính phủ. Mạng máy tính liên kết các máy tính và các thiết bị điện tử khác thông qua mạng viễn thông, do đó người sử dụng có thể tiếp cận các thông tin được lưu trữ và giao tiếp giữa các máy tính độc lập nhưng đại đa số các máy tính cá nhân đã kết nối với mạng máy tính toàn cầu, đó là internet, hoặc kết nối trong nội bộ của một tổ chức, đó là intranet. Intranet là mạng công ty thực hiện chức năng công nghệ thông tin như trình duyệt hoặc sử dụng giao diện internet. Một môi trường máy tính khác là extranet, là mạng nối kết intranet của các đối tác kinh doanh với internet. Vậy, tại sao các doanh nghiệp lại quan tâm đến thương mại điện tử? Lý do là ở chỗ công nghệ thông tin nói chung và thương mại điện tử nói riêng đã trở thành nhân tố thúc đẩy cơ bản hoạt động kinh doanh. Thương mại điện tử đã trở thành yếu tố xúc tác làm thay đổi cơ cấu và quản lý của các tổ chức.

Thương mại điện tử (Electronic commerce – EC or E.Commerce) là một khái niệm được dùng để mô tả quá trình mua và bán hoặc trao đổi sản phẩm, dịch vụ và thông tin thông qua mạng máy tính, kể cả internet. Thuật ngữ “Thương mại” (Commerce) được nhiều người hiểu là một số giao dịch được thực hiện giữa các đối tác kinh doanh. Vì vậy, thương mại điện tử cũng thường được hiểu theo nghĩa hẹp là mua và bán trên mạng, hay mua bán thông qua các phương tiện điện tử. Họ đồng nghĩa E.Commerce với E.Trade.

Trên thực tế, có nhiều người sử dụng thuật ngữ “Kinh doanh điện tử (Electronic business – E.Business)” để chỉ một phạm vi rộng hơn của thương mại điện tử. Đó không chỉ là quá trình mua và bán mà còn là dịch vụ

khách hàng, kết nối với các đối tác kinh doanh, thực hiện các giao dịch điện tử trong phạm vi một tổ chức. Kinh doanh điện tử (E.Business) là chu kỳ kinh doanh, tốc độ kinh doanh, toàn cầu hoá, nâng cao năng suất, tiếp cận khách hàng mới và chia sẻ kiến thức giữa các tổ chức nhằm đạt được lợi thế cạnh tranh<sup>(1)</sup>. Thuật ngữ “Thương mại (Commerce)” trong cuốn sách này được hiểu theo nghĩa rộng, là các hoạt động kinh tế nhằm mục tiêu sinh lợi của các chủ thể kinh doanh trên thị trường<sup>(2)</sup>. Vì vậy, thương mại điện tử (E.Commerce) cũng được hiểu tương đương với kinh doanh điện tử (E.Business) và có thể xem xét dưới các góc độ sau đây:

– *Xem xét từ góc độ số hoá*: Thương mại điện tử có thể thực hiện dưới nhiều hình thức phụ thuộc vào mức độ số hoá của các sản phẩm/dịch vụ mua bán, quá trình mua bán và cơ quan vận chuyển và giao nhận hàng. Hình 6.1 cho thấy mức độ số hoá của thương mại điện tử. Một sản phẩm có thể là sản phẩm vật chất hoặc sản phẩm số hoá ào, một tổ chức có thể là một cơ quan cụ thể hoặc một cơ quan số hoá ào và một quá trình cũng có thể là quá trình vật chất hay số hoá ào. Điều này tạo nên tám hình khối, mỗi hình khối có ba khía cạnh. Trong thương mại truyền thống, tất cả các khía cạnh này đều là hữu hình (Hình khối phía dưới bên trái). Trong thương mại điện tử thuần túy, tất cả các khía cạnh này đều là số hoá (Hình khối phía trên bên phải). Tất cả các hình khối còn lại đều là hỗn hợp của thương mại truyền thống và thương mại điện tử thuần túy, tuỳ thuộc vào mức độ số hoá của các hoạt động trong các giao dịch mà có thể xem đó là giao dịch thương mại truyền thống nhưng nếu việc mua hàng, giao nhận hàng và thanh toán đều được số hoá thì đó là thương mại điện tử thuần túy.

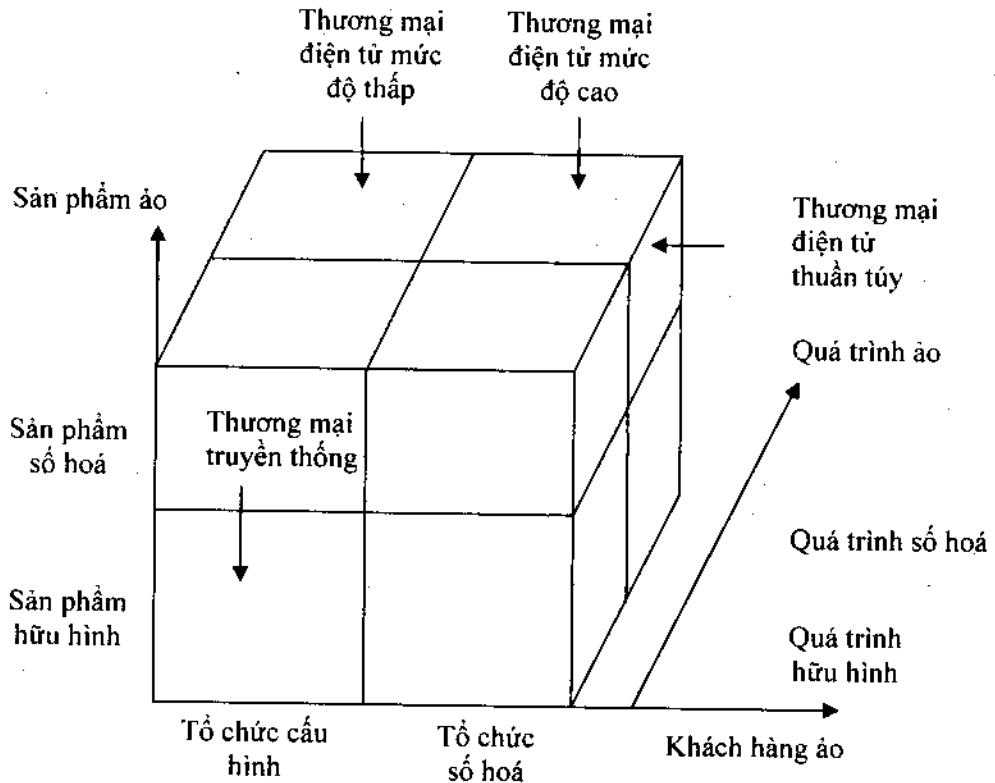
– *Xem xét từ góc độ các lĩnh vực kinh doanh*: Thương mại điện tử diễn ra ở hầu khắp các lĩnh vực kinh doanh, thúc đẩy các lĩnh vực đó phát triển đồng thời tạo nên bản sắc mới của hoạt động kinh doanh trên phạm vi nền kinh tế thế giới.

– *Từ góc độ kinh doanh viễn thông*: Thương mại điện tử là việc chuyên giao thông tin, sản phẩm hay dịch vụ hoặc thanh toán thông qua điện thoại, mạng máy tính, hoặc bất kỳ phương tiện điện tử nào khác.

<sup>(1)</sup>. Theo Lou Gerster, IBM's CEO.

<sup>(2)</sup>. Giáo trình Kinh tế Thương mại, GS.TS. Đặng Đình Đào – GS.TS. Hoàng Đức Thân (Chủ biên), NXB Thông kê, Trường Đại học Kinh tế Quốc dân, 2003.

- *Từ góc độ quá trình kinh doanh*: Thương mại điện tử là việc ứng dụng công nghệ để tự động hóa các giao dịch kinh doanh và các dòng chu chuyển sản phẩm.
- *Từ góc độ kinh doanh dịch vụ*: Thương mại điện tử là phương tiện để các doanh nghiệp, người tiêu dùng và nhà quản lý cắt giảm chi phí dịch vụ trong khi vẫn nâng cao chất lượng hàng hoá, tăng tốc độ chuyển giao dịch vụ.
- *Từ góc độ trợ giúp trực tuyến*: Thương mại điện tử cung cấp khả năng mua và bán sản phẩm và thông tin trên internet và dịch vụ trực tuyến khác.



**Hình 6.1.** Thương mại điện tử theo mức độ số hóa

Nguồn: Choi (edit): *Kinh tế học của thương mại điện tử*, NXB Kỹ thuật Macmillan, 1997

## 1.2. Đặc trưng của thương mại điện tử

Thương mại điện tử không thể hiện các văn bản giao dịch trên giấy (Paperless Transactions). Tất cả các văn bản đều có thể thể hiện bằng các dữ

liệu tin học, các băng ghi âm hay các phương tiện điện tử khác. Đặc trưng này làm thay đổi căn bản văn hoá giao dịch bởi lẽ độ tin cậy không còn phụ thuộc vào cam kết bằng giấy tờ mà bằng niềm tin lẫn nhau giữa các đối tác. Giao dịch không dùng giấy cũng làm giảm đáng kể chi phí và nhân lực để chu chuyển, lưu trữ và tìm kiếm các văn bản khi cần thiết. Người sử dụng thông tin có thể tìm kiếm ngay trong ngân hàng dữ liệu của mình mà không cần người khác tham gia nên bảo vệ được bí mật ý tưởng và cách thức thực hiện ý đồ kinh doanh. Giao dịch không dùng giấy đòi hỏi kỹ thuật bảo đảm an ninh và an toàn dữ liệu mới. Đó là an ninh và an toàn giao dịch thương mại điện tử.

– Thương mại điện tử phụ thuộc công nghệ và trình độ công nghệ thông tin của người sử dụng. Để phát triển thương mại điện tử cần phải xây dựng và không ngừng nâng cao trình độ công nghệ thông qua phát triển cơ sở hạ tầng kỹ thuật của thương mại điện tử như mạng máy tính và khả năng tiếp nối của mạng với các cơ sở dữ liệu thông tin toàn cầu. Cùng với cơ sở mạng, thương mại điện tử cần có đội ngũ nhân viên không chỉ thành thạo về công nghệ mà còn kiến thức và kỹ năng về quản trị kinh doanh nói chung, về thương mại nói riêng.

– Thương mại điện tử phụ thuộc mức độ số hoá (Thương mại số hoá). Tùy thuộc mức độ số hoá của nền kinh tế và khả năng hội nhập số hoá với nền kinh tế toàn cầu mà thương mại điện tử có thể đạt được các cấp độ từ thấp đến cao. Cấp độ thấp nhất là sử dụng thư điện tử, đến Internet để tìm kiếm thông tin, đến đặt hàng trực tuyến và dịch vụ trực tuyến, đến xây dựng các website cho hoạt động kinh doanh và cuối cùng là áp dụng các giải pháp toàn diện về thương mại điện tử (thương mại điện tử thuần túy).

– Thương mại điện tử có tốc độ nhanh. Nhờ áp dụng kỹ thuật số nên tất cả các bước của quá trình giao dịch đều được tiến hành thông qua mạng máy tính. Ngôn ngữ của công nghệ thông tin cũng cho phép rút ngắn độ dài của các “văn bản” giao dịch. Các dịch vụ phần mềm ngày càng hoàn hảo, tốc độ đường truyền nhanh cho phép rút ngắn thời gian soạn thảo, giao tiếp và ký kết các văn bản giao dịch điện tử. Tất cả những điều này đã làm cho thương mại điện tử đạt tốc độ nhanh nhất trong các phương thức giao dịch, tạo nên tính cách mạng trong giao dịch thương mại.

## 2. SỰ KHÁC BIỆT CỦA THƯƠNG MẠI ĐIỆN TỬ VÀ THƯƠNG MẠI TRUYỀN THÔNG

Thương mại điện tử, do những đặc trưng của nó, khác với thương mại truyền thống trên nhiều phương tiện như công nghệ thực hiện, đặc điểm thị trường, tiến trình mua hàng của khách hàng.

### 2.1. Khác biệt về công nghệ

Thương mại điện tử liên kết công nghệ truyền thông và công nghệ thông tin với tiến trình kinh doanh của doanh nghiệp. Mỗi quan hệ này được thể hiện ở bảng 6.1.

**Bảng 6.1. QUAN HỆ GIỮA CÔNG NGHỆ VÀ QUÁ TRÌNH KINH DOANH TRONG THƯƠNG MẠI ĐIỆN TỬ**

Lĩnh vực TMĐT	Công nghệ	Quá trình kinh doanh
Truy cập thông tin (khách hàng cho phép doanh nghiệp truy cập CSDL của mình)	<p><i>Khách hàng:</i> CSDL là thông tin đáng tin cậy Bức tường lửa để kiểm soát truy cập từ bên ngoài</p> <p><i>Doanh nghiệp:</i> Máy tính với khả năng truy cập mạng</p>	<p>Khách hàng bảo đảm tính kịp thời của CSDL</p> <p>Khách hàng bảo đảm thông báo cho doanh nghiệp những thay đổi</p> <p>Doanh nghiệp đồng ý sử dụng CSDL</p>
Dịch vụ viễn thông (khách hàng và doanh nghiệp hợp tác thiết kế sản phẩm)	<p>Hệ thống thiết kế bằng mạng máy tính có thể đọc được các files của nhau</p> <p>Các ứng dụng kiểm tra bản vẽ</p>	<p>Khách hàng và doanh nghiệp thoả thuận hợp tác trong thiết kế</p> <p>Chấp nhận một hệ thống thiết kế trên mạng máy tính phù hợp</p> <p>Đào tạo các nhóm thiết kế</p>
Mua bán hàng hoá qua trang Web	<p><i>Người bán:</i> Trang Web bán hàng trực tuyến</p> <p><i>Người mua:</i> Khả năng giữ được trang Web trong môi trường thay đổi</p>	<p><i>Người bán:</i> Khả năng giữ được trang Web trong môi trường thay đổi</p> <p><i>Người mua:</i> Có hệ thống đảm nhiệm hoạt động mua hàng trực tuyến</p>
Các mô hình kinh doanh ảo (Hệ thống cung ứng hợp nhất – Integrated supply)	<p>EDI</p> <p>E-mail</p> <p>Mẫu biểu điện tử</p>	<p>Xây dựng tiến trình nhập đơn đặt hàng và hệ thống mua bán</p> <p>Sử dụng nhân viên để giải quyết các công việc phát sinh</p>

## 2.2. Khác biệt về tiến trình mua bán

Khác với thương mại truyền thống, tiến trình mua bán trong thương mại điện tử thay đổi cả về hình thức và nội dung thực hiện (xem bảng 6.2).

**Bảng 6.2. TIẾN TRÌNH MUA BÁN TRONG THƯƠNG MẠI TRUYỀN THÔNG VÀ THƯƠNG MẠI ĐIỆN TỬ**

Tiến trình mua bán	Thương mại điện tử	Thương mại truyền thống
1. Thu nhận thông tin	Trang Web, Catalogue trực tuyến	Tạp chí, tờ rơi, Catalogue giấy v.v...
2. Mô tả hàng hóa	Các mẫu biểu điện tử, e.mail v.v...	Thư và các mẫu biểu in trên giấy
3. Kiểm tra khả năng cung ứng và thỏa thuận giá	E.mail, Web, EDI v.v...	Điện thoại, thư, fax v.v...
4. Tạo đơn hàng	Đơn hàng điện tử	Đơn hàng trên giấy, in sẵn
5. Trao đổi thông tin	E.mail, EDI	Thư, Fax
6. Kiểm hàng tại kho	Các mẫu biểu điện tử, EDI, E.mail v.v...	Các mẫu biểu in sẵn, Fax
7. Giao hàng	Chuyển hàng trực tuyến, phương tiện vận tải	Phương tiện vận tải
8. Thông báo	e.mail, EDI	Thư, Fax, điện thoại
9. Chứng từ	Chứng từ điện tử	Chứng từ in trên giấy
10. Thanh toán	EDI, tiền điện tử, giao dịch ngân hàng số hoá	Séc, hồi phiếu, tiền mặt, thanh toán qua ngân hàng

## 2.3. Khác biệt về thị trường

Thị trường điện tử là thị trường được phát triển trên cơ sở ứng dụng công nghệ Internet. Do môi trường ảo, với những đặc trưng riêng, thị trường điện tử có nhiều điểm khác biệt với thị trường trong thương mại truyền thống. Sự khác biệt đó được thể hiện qua sự so sánh giữa thị trường truyền thống và thị trường điện tử (bảng 6.3).

**Bảng 6.3. SỰ KHÁC BIỆT GIỮA THỊ TRƯỜNG ĐIỆN TỬ VÀ  
THỊ TRƯỜNG TRUYỀN THÔNG**

Thị trường truyền thông	Thị trường điện tử
Marketing và quảng cáo rộng rãi	Marketing và quảng cáo có mục tiêu, tương tác một – một
Sản xuất đại trà (Sản phẩm và dịch vụ tiêu chuẩn)	Khách hàng hoá quá trình sản xuất
Tư duy thiên về phía cung	Tư duy nghiêng về phía cầu
Khách hàng là mục tiêu	Khách hàng là đối tác
Thị trường phân tách	Thị trường công đồng
Sản phẩm và dịch vụ vật chất	Sản phẩm và dịch vụ số hoá
Nhãn hiệu trên hàng hoá	Giao tiếp, mô tả
Sử dụng trung gian	Không sử dụng trung gian hoặc trung gian kiểu mới
Danh mục hàng hoá trên giấy	Danh mục hàng hoá điện tử

### **3. LỢI ÍCH VÀ TÁC ĐỘNG CỦA THƯƠNG MẠI ĐIỆN TỬ**

#### **3.1. Lợi ích và những vấn đề đặt ra với thương mại điện tử**

*Lợi ích của thương mại điện tử:* Trong lịch sử nhân loại chỉ có một vài đồi mới mang lại những lợi ích tiềm năng như thương mại điện tử. Đó là tạo ra bản chất toàn cầu của công nghệ, cắt giảm chi phí, tạo cơ hội để tiếp cận hàng triệu người trong một thời gian ngắn, tạo ra sự tương tác lẫn nhau và kích thích tính năng động, tạo ra nhiều khả năng và khai thác tối đa các nguồn lực, tạo ra sự tăng trưởng nhanh của cơ sở hạ tầng trợ giúp làm tăng lợi ích tiềm năng của các tổ chức, cá nhân và toàn xã hội. Những lợi ích này mới chỉ được hiện thực hoá và sẽ còn tăng lên đáng kể khi thương mại điện tử được áp dụng rộng rãi hơn.

*Lợi ích đối với các tổ chức:* Lợi ích đối với các tổ chức của thương mại điện tử bao gồm:

- Thương mại điện tử mở rộng phạm vi giao dịch trên thị trường toàn cầu. Với một lượng vốn tối thiểu, các doanh nghiệp dễ dàng và nhanh chóng

tiếp cận được với nhiều khách hàng, lựa chọn được nhà cung ứng tốt nhất và xác định được đối tác kinh doanh phù hợp nhất.

– Thương mại điện tử làm giảm chi phí thu thập, xử lý, phân phối, lưu trữ và sử dụng thông tin. Chẳng hạn, áp dụng đấu thầu mua sắm điện tử, doanh nghiệp có thể cắt giảm chi phí quản trị mua sắm đến 85%. Trong thanh toán, nhờ sử dụng các phương tiện thanh toán điện tử, công ty có thể cắt giảm chi phí phát hành séc bằng giấy.

– Thương mại điện tử tạo ra khả năng chuyên môn hoá cao trong kinh doanh, đặc biệt là đối với các doanh nghiệp thương mại. Nhiều siêu thị điện tử quy mô nhỏ sẽ chuyên môn hoá vào bán một hoặc một số mặt hàng, ví dụ các siêu thị [www.dogtoys.com](http://www.dogtoys.com) hay [www.cattoys.com](http://www.cattoys.com).

– Thương mại điện tử góp phần giảm lượng tồn kho và đòi hỏi về cơ sở vật chất kỹ thuật thông qua việc áp dụng phương pháp quản lý dây chuyền cung ứng "kéo" ("pull" – type supply chain management). Quá trình này bắt đầu từ đặt hàng của khách hàng và sử dụng phương pháp sản xuất đúng thời hạn. Phương pháp kéo thúc đẩy sự tương thích sâu sắc giữa nhu cầu của khách hàng với sản phẩm và dịch vụ của doanh nghiệp tạo ra lợi thế cạnh tranh khi marketing trên thị trường.

– Thương mại điện tử làm giảm thời gian khi thanh toán tiền đến khi nhận được hàng hoá hoặc dịch vụ.

– Thương mại điện tử kích thích sự sáng tạo và tạo điều kiện để khởi động những dự án kinh doanh mới, tăng khả năng thành công của các phương án kinh doanh nhờ thay đổi quy trình cho hợp lý, tăng năng suất của người bán hàng, trang bị kiến thức cho người lao động, đặc biệt là lao động quản lý.

– Thương mại điện tử làm giảm chi phí viễn thông trong quá trình giao tiếp, đàm phán và ký kết các hợp đồng mua bán hàng hoá/dịch vụ. Chẳng hạn, sử dụng Internet rẻ hơn rất nhiều sử dụng VANs (value added network services - là các hãng dịch vụ không chỉ cung cấp các mạng vật lý để truyền dữ liệu mà còn có các tiện ích đi kèm như email, bảo mật, kiểm định...).

– Thương mại điện tử cũng góp phần cải thiện hình ảnh doanh nghiệp, nâng cao chất lượng dịch vụ khách hàng, tìm kiếm đối tác kinh doanh, đơn

giản hoá quá trình kinh doanh, rút ngắn chu kỳ và thời gian giao nhận hàng hoá, tăng năng suất, loại bỏ giấy tờ, xử lý thông tin nhanh hơn, giảm chi phí vận tải, tăng tính linh hoạt trong kinh doanh của doanh nghiệp.

Lợi ích đối với người tiêu dùng: Lợi ích với người tiêu dùng của thương mại điện tử bao gồm:

- Thương mại điện tử cho phép khách hàng mua sắm và thực hiện các giao dịch 24/24 giờ trong ngày, tất cả các ngày trong năm và không bị giới hạn bởi phạm vi địa lý.

- Thương mại điện tử cung cấp cho khách hàng nhiều sự lựa chọn hơn. Khách hàng có thể lựa chọn các cơ sở cung cấp khác nhau, từ máy bán hàng tự động cho đến siêu thị. Lựa chọn các loại sản phẩm khác nhau, từ hàng điệu từ lâu bền đến một món quà tặng.

- Thương mại điện tử làm giảm chi tiêu cho khách hàng về sản phẩm hàng hoá/dịch vụ họ nhận được thông qua việc chấp nhận mua bán không phụ thuộc vào vị trí địa lý của người cung ứng và có thể so sánh để lựa chọn người cung ứng nhanh nhất với giá cả phù hợp nhất.

- Trong một số trường hợp, đặc biệt là các sản phẩm số hoá, thương mại điện tử có khả năng giao hàng rất nhanh cho khách hàng.

- Thương mại điện tử tạo khả năng cho khách hàng tham gia các cuộc đấu giá trên mạng.

- Thương mại điện tử tạo điều kiện để các khách hàng tác động, hỗ trợ lẫn nhau trong cộng đồng kinh doanh thương mại điện tử nhằm trao đổi các ý tưởng và kinh nghiệm kinh doanh.

- Thương mại điện tử thúc đẩy cạnh tranh và từ đó dẫn đến sự giảm giá bền vững.

Lợi ích đối với xã hội. Lợi ích đối với xã hội của thương mại điện tử bao gồm:

- Thương mại điện tử cho phép nhiều người có thể làm việc tại nhà, giảm thiểu việc đi mua sắm do đó giảm phương tiện giao thông lưu thông trên đường, giảm thiểu tai nạn và ô nhiễm môi trường sống.

- Thương mại điện tử dẫn đến việc bán hàng với giá thấp hơn nên nhiều người có thể mua được khối lượng hàng hoá lớn hơn, tăng mức sống của dân cư.

– Thương mại điện tử tạo điều kiện để dân cư ở các nước đang phát triển và khu vực nông thôn hưởng thụ các sản phẩm và dịch vụ mà trong hoàn cảnh khác họ không có khả năng như cơ hội để nâng cao kỹ năng nghề nghiệp và nhận được bằng cấp cao hơn.

– Thương mại điện tử thúc đẩy việc cung cấp các dịch vụ công cộng như chăm sóc sức khoẻ, giáo dục và phân phối các dịch vụ xã hội của Chính phủ ở mức chi phí thấp hoặc cải thiện chất lượng của các dịch vụ đó.

Những vấn đề đặt ra của thương mại điện tử hay những hạn chế của thương mại điện tử có thể phân chia thành hai nhóm, gồm các vấn đề về kỹ thuật và phi kỹ thuật.

*Những vấn đề kỹ thuật của thương mại điện tử bao gồm :*

- Thiếu sự an toàn của cả hệ thống, độ tin cậy, các chuẩn mực và cơ sở kỹ thuật cho những giao tiếp cần thiết.
- Độ rộng dài cần còn nhiều.
- Các công cụ phát triển phần mềm thay đổi thường xuyên.
- Còn tồn tại rất nhiều khó khăn trong việc hợp nhất Internet, phần mềm thương mại điện tử với một số ứng dụng cơ sở dữ liệu hiện có.
- Thiết bị ngoại vi cần những máy chủ, trang Web đặc biệt và cơ sở hạ tầng kỹ thuật khác bổ sung cho máy chủ của mạng giao dịch.
- Một số phần mềm thương mại điện tử không khớp với phần cứng hoặc không tương thích với hệ thống vận hành hoặc các thiết bị khác.
- Tuy nhiên, những vấn đề kỹ thuật có thể dần dần được khắc phục cùng với sự tiến bộ của công nghệ và chuyên giao công nghệ từ những quốc gia phát triển sang các quốc gia đang phát triển. Đồng thời, các doanh nghiệp tham gia thương mại điện tử cũng cần có một kế hoạch thích hợp để giảm thiểu ảnh hưởng tiêu cực của những vấn đề kỹ thuật.

*Những vấn đề phi kỹ thuật của thương mại điện tử:* Trên thực tế còn tồn tại rất nhiều những vấn đề phi kỹ thuật đã làm giảm tốc độ phát triển và sự mở rộng nhanh chóng của thương mại điện tử, những vấn đề đó bao gồm:

- Chi phí và hạch toán các chi phí: Chi phí phát triển thương mại điện tử trong các hộ gia đình có thể rất cao và có thể gấp những trặc do thiếu

kinh nghiệm. Có rất nhiều cơ hội để áp dụng các phương pháp tổ chức sản xuất kinh doanh tiên bộ (quốc tế hoá sản xuất theo chiều rộng và chiều sâu, sử dụng phương pháp JIT...) nhưng ứng dụng ở đâu và như thế nào thì không đơn giản. Hơn nữa, việc so sánh giữa chi phí bỏ ra và lợi ích thu được từ thương mại điện tử của các doanh nghiệp và hộ gia đình đang phải giải quyết vấn đề do lường những lợi ích vô hình như thế nào, chẳng hạn như cải thiện dịch vụ khách hàng và giá trị của quảng cáo.

– Vấn đề an toàn và bảo mật: Mặc dù vấn đề an toàn và bảo mật đã được cải thiện nhiều trong quá trình phát triển không ngừng của thương mại điện tử nhưng khách hàng vẫn coi đây là vấn đề cốt tử trong giao dịch của họ và do đó thương mại điện tử phải đổi mới với một nhiệm vụ khó khăn và lâu dài, đó là thuyết phục khách hàng rằng, giao dịch trực tuyến là an toàn và vẫn đảm bảo bí mật riêng của các cá nhân giao dịch.

– Thiếu niềm tin đối với khách hàng: Khách hàng thường không tin tưởng vào những đối tác giao dịch không hiện diện, không thể hiện trên giấy và thanh toán bằng tiền điện tử.

#### *Những vấn đề kỹ thuật khác:*

– Không thể cảm quan trực tiếp được sản phẩm trong giao dịch thương mại điện tử, ví dụ, người mua không thể xem xét trực tiếp được mẫu mã khi họ muốn mua trang phục cho mình.

– Nhiều vấn đề pháp lý không thể giải quyết được trong thương mại điện tử, đặc biệt là những quy định của chính phủ và những tiêu chuẩn cần thiết rất khó thiết lập và điều chỉnh kịp thời.

– Thiếu các dịch vụ trợ giúp cần thiết như vấn đề bảo hộ bản quyền trong các giao dịch thương mại điện tử còn yếu, các chuyên gia về thuế trong thương mại điện tử còn ít v.v...

– Thiếu một lực lượng bán hàng có kỹ thuật trong thương mại điện tử để đảm bảo lợi nhuận cho doanh nghiệp.

– Thương mại điện tử cũng có thể dẫn đến phá vỡ các mối quan hệ cần thiết giữa những người tham gia giao dịch.

– Tiếp cận mạng Internet vẫn còn đắt và không thuận tiện với nhiều khách hàng tiềm năng.

### **3.2. Tác động của thương mại điện tử**

. Tác động của thương mại điện tử có thể được xem xét trên ba phương diện: Thúc đẩy marketing sản phẩm, thay đổi bản chất thị trường, thay đổi các tổ chức và cải thiện hoạt động sản xuất kinh doanh của các doanh nghiệp.

#### *Thương mại điện tử thúc đẩy marketing sản phẩm*

Marketing trực tiếp truyền thống được thực hiện qua các thư đặt hàng hoặc các cuộc giao dịch bằng điện thoại. Hiện nay, marketing trực tiếp thông qua mạng máy tính tăng lên nhanh chóng, khoảng 2 tỷ USD ở Mỹ năm 1998. Điều này cho thấy tác động của thương mại điện tử đến marketing trực tiếp trên các mặt sau đây:

- *Xúc tiến sản phẩm*: Thương mại điện tử xúc tiến sản phẩm hàng hoá/dịch vụ thông qua việc cung cấp thông tin đầy đủ, trực tiếp cho khách hàng và tiếp xúc trực tiếp với khách hàng.
- *Các kênh phân phối mới*: Thương mại điện tử tạo ra những kênh phân phối mới cho sản phẩm hiện đại thông qua giao tiếp trực tiếp với khách hàng hoặc các phương tiện giao tiếp khác.
- *Tiết kiệm chi phí*: Thương mại điện tử làm giảm đáng kể chi phí cho khách hàng nếu so với thương mại truyền thống, đặc biệt là trong việc chuyển giao các sản phẩm số hoá.
- *Rút ngắn chu kỳ kinh doanh*: Việc chuyển đưa các sản phẩm hàng hoá/dịch vụ số hoá có thể chỉ còn tính bằng giây, công việc quản lý liên quan đến chuyển giao các sản phẩm hữu hình trên phạm vi thị trường thế giới cũng giảm đáng kể, có những nghiệp vụ giao dịch thương mại quốc tế rút ngắn đến 99% chu kỳ kinh doanh, chẳng hạn hàng Tradenet đã giảm thời gian làm thủ tục giao dịch liên quan đến cảng từ tính bằng ngày xuống phút.
- *Tăng dịch vụ khách hàng*: Dịch vụ khách hàng được cải thiện đáng kể thông qua cung ứng các thông tin trực tuyến, các hằng tư vấn có thể trả lời khách hàng bằng thư điện tử trong một vài giây, các dịch vụ nhân sự có thể thực hiện bởi các phần mềm trợ giúp.
- *Định vị hình ảnh doanh nghiệp và sản phẩm*: Hình ảnh sản phẩm và doanh nghiệp có thể được định vị trong khách hàng mới rất nhanh thông qua các trang Web. Niềm tin của khách hàng và công chúng vào doanh nghiệp

được xây dựng qua những thông tin trên các mạng máy tính và là nhân tố để tăng khả năng bán hàng hoá/dịch vụ.

– *Các tác động marketing khác:* Thương mại điện tử còn góp phần làm cho sản phẩm hàng hoá/dịch vụ của doanh nghiệp tương thích với nhu cầu của khách hàng, quảng cáo sống động hơn, khuếch trương hơn và mang thông tin đến khách hàng đầy đủ hơn, nhanh hơn so với quảng cáo truyền thống. Hệ thống đặt hàng trực tuyến nhanh hơn, chính xác hơn, được xử lý khoa học hơn, chính xác hơn, do đó làm giảm thời gian và chi phí.

#### *Thương mại điện tử làm thay đổi bản chất thị trường*

Thị trường truyền thống (trong điều kiện không gian và thời gian cụ thể) không tồn tại trong thương mại điện tử và được thay bằng thị trường điện tử. Trong thị trường điện tử, hàng hoá được chuyển giao trực tiếp đến người mua khi hành vi mua được hoàn thành làm cho hiệu quả hơn. Những sản phẩm dựa trên nền tảng số hoá như phần mềm, âm nhạc, thông tin đã thay đổi nhanh chóng. Tuy rất nhỏ nhưng những sản phẩm phần mềm mạnh mẽ được chuyển giao qua Internet đã làm giảm nhu cầu về mô hình phân phối truyền thống. Những mô hình bán hàng mới như phần mềm tự nguyện, phần mềm miễn phí, phần mềm trả tiền khi sử dụng đang được sử dụng rộng rãi làm tăng thêm tiềm năng của Internet. Mặc dù các mô hình bán hàng mới chỉ phát triển ở một số khu vực thị trường như phần mềm và xuất bản nhưng chắc chắn sẽ lan sang các khu vực thị trường mới. Thương mại điện tử cũng làm thay đổi cách thức giao tiếp trên thị trường, giao tiếp không gặp mặt và có thể thực hiện đồng thời giữa nhiều người nói lên trên thị trường điện tử làm tăng khả năng giao tiếp trong kinh doanh.

Hơn nữa, thương mại điện tử góp phần tạo nên những sản phẩm mới và đổi mới các sản phẩm hiện tại cho phù hợp nhu cầu khách hàng. Điều đó làm thay đổi nhiệm vụ của tổ chức và cách thức hoạt động trên thị trường. Thương mại điện tử tạo điều kiện để doanh nghiệp thu nhận đầy đủ hơn các thông tin về khách hàng, xây dựng hồ sơ khách hàng, từ đó thiết kế sản phẩm hoặc hoàn thiện sản phẩm phù hợp với khách hàng. Chẳng hạn, nhiều công ty trên thế giới như Motorola, Dell Computer, JCPenny, Levi's đã nghiên cứu nhu cầu của khách hàng cụ thể, chế tạo sản phẩm phù hợp nhu cầu và gửi sản phẩm đến tận tay khách hàng ngay trong ngày. Sử dụng các trang Web, khách hàng cũng có thể tự thiết kế sản phẩm cho mình hoặc

phác họa ý tưởng về sản phẩm và đề nghị các công ty chế tạo và cung ứng cho họ. Những hình thức cung ứng sản phẩm cho khách hàng nhờ áp dụng các mô hình mới thông qua mạng máy tính vẫn có thể duy trì mức giá như thương mại truyền thống đã làm thay đổi bản chất của thị trường, đặc biệt là thị trường bán lẻ.

#### *Thương mại điện tử làm thay đổi tổ chức*

Những bước tiến nhanh chóng của thương mại điện tử đã buộc các doanh nghiệp phải bổ sung công nghệ mới và nghiên cứu những sản phẩm hàng hoá/dịch vụ mới. Việc bổ sung công nghệ mới sẽ dẫn đến những thay đổi về cơ cấu và chiến lược. Sự thay đổi đó lại là nhân tố thúc đẩy sự thay đổi cách thức kinh doanh. Trong xu hướng đó, cơ cấu của tổ chức sẽ nhanh chóng thích ứng với những yêu cầu của công nghệ. Đó là một quá trình do đòi hỏi khách quan trong sự thay đổi nhu cầu của khách hàng, do những phản hồi của khách hàng hiện tại và cả những cố gắng của doanh nghiệp. Hơn nữa, một cơ cấu tổ chức mới khi thực hiện thương mại điện tử lại đòi hỏi các nhà quản trị và marketing phải linh hoạt hơn để đáp ứng những đòi hỏi mới của tổ chức lại trở thành nhân tố tác động tích cực đến việc ứng dụng thương mại điện tử.

Song song với thay đổi cơ cấu tổ chức, thương mại điện tử cũng góp phần thay đổi bản chất của công việc theo hướng chuyên nhanh sang kỹ nguyên số hoá. Bị thúc đẩy bởi sự cạnh tranh ngày càng cao trên thị trường toàn cầu, các doanh nghiệp đang cố gắng giảm lao động và đưa những công việc cần sử dụng nhiều lao động sang các quốc gia có chi phí lao động thấp. Sự thay đổi này sẽ tạo ra những cơ hội mới và những rủi ro mới, cũng như cách tư duy mới về việc làm, nghề nghiệp và tiền lương. Những người công nhân của kỹ nguyên số hoá sẽ trở nên linh hoạt. Nhiều người lao động sẽ trở nên yên tâm với công việc, học tập và rèn luyện để tăng thêm kiến thức chuyên môn và kỹ năng nghề nghiệp. Nhiều người sẽ được chuyển từ chế độ làm việc tại văn phòng sang làm việc tại nhà do đó hành vi mua sắm của các hộ gia đình cũng thay đổi.

#### *Thương mại điện tử tác động đến chế tạo sản phẩm hàng hoá/dịch vụ*

Thương mại điện tử thay đổi hệ thống chế tạo hàng hoá/dịch vụ từ chỗ định hướng sản xuất đến định hướng vào nhu cầu, định hướng khách hàng và mô hình sản xuất đúng thời hạn. Hệ thống sản xuất từng bước hợp nhất

với hệ thống tài chính, phân phối và các hệ thống chức năng khác. Chẳng hạn, các đơn đặt hàng có thể từ khách hàng chuyên trực tiếp tới người thiết kế thông qua công cụ thiết kế trên mạng máy tính, các thiết kế được chuyển tới người sản xuất. Quá trình này có thể tính theo giấy và do đó có thể cắt giảm chu kỳ kinh doanh hơn 50%, đặc biệt là trong trường hợp quá trình thiết kế, chế tạo đã được quốc tế hóa như IBM, General Motor, General Electronic, Boeng v.v... Thương mại điện tử cũng làm cho hệ thống chế tạo hàng hóa/dịch vụ trở nên linh hoạt, dễ hoàn thiện và giảm được chi phí, đặc biệt là chi phí dự trữ các yếu tố đầu vào và chi phí tồn kho thành phẩm.

#### *Thương mại điện tử tác động đến tình hình tài chính và hạch toán kinh doanh của doanh nghiệp*

Thương mại điện tử đòi hỏi hệ thống tài chính và kế toán đặc biệt. Hầu hết các nghiệp vụ thanh toán trong thương mại điện tử không thể hiện trên giấy. Việc sử dụng các phương tiện thanh toán mới như tiền điện tử sẽ làm cho quá trình này phức tạp hơn do sự chi phối của luật pháp và các thông lệ quốc tế. Tuy nhiên, tiền điện tử đã làm cho quá trình thanh toán nhanh hơn và thay đổi cách thức thanh toán. Trong nhiều trường hợp, tiền điện tử lại được hoàn trả bằng tiền hoặc một loại tài sản khác sẽ làm thay đổi cách thức chi tiêu và thói quen tiêu dùng của khách hàng. Tiền điện tử và hệ thống thanh toán điện tử trong hoạt động mua bán sẽ làm thay đổi phương pháp kế toán truyền thống, phương thức chuyên tiền liên thẻ chế được áp dụng. Để xử lý các thư đặt hàng điện tử, phương pháp giao dịch và thanh toán trả sau được sử dụng rộng rãi. Ví dụ, khi thư đặt hàng điện tử đến người bán, bán xác nhận trực tuyến của người bán, việc thanh toán có thể được thực hiện bằng séc tín dụng của người mua, séc bảo đảm bằng sản phẩm hoặc các phương tiện thanh toán khác.

#### *Thương mại điện tử tác động đến quản trị và đào tạo nguồn nhân lực.*

Thương mại điện tử làm thay đổi cách thức tuyển dụng, đánh giá, thăng tiến và phát triển nguồn nhân lực. Tất cả những vấn đề quản trị nguồn nhân lực của doanh nghiệp được sự hỗ trợ đặc lực của mạng máy tính nội bộ và mức độ chính xác sẽ cao hơn. Đào tạo nguồn nhân lực cũng có sự thay đổi. Đào tạo từ xa trực tuyến cung cấp cơ hội cho mọi người. Các doanh nghiệp nhờ đó mà cắt giảm chi phí đào tạo nhân lực. Nhiều công ty xem đào tạo trực tuyến trên mạng máy tính như một cánh cùu vì nó làm thay đổi môi trường, thay đổi công nghệ, thay đổi cách thức đào tạo người lao động tại chỗ.

#### **4. CÁC ĐIỀU KIỆN PHÁT TRIỂN THƯƠNG MẠI ĐIỆN TỬ**

Thương mại điện tử không đơn thuần là dùng phương tiện điện tử để thực hiện hành vi thương mại truyền thống mà với ứng dụng thương mại điện tử, toàn bộ nền thương mại của một quốc gia sẽ thay đổi, phương pháp quản lý truyền thống sẽ kém hiệu quả và cần áp dụng phương pháp quản lý mới, phương tiện thanh toán mới, phải thay đổi cơ sở nhận thức và hệ thống giáo dục, tập quán làm việc. Những thay đổi này cho thấy, để phát triển thương mại điện tử cần điều kiện về công nghệ, nhận thức xã hội, luật pháp và an ninh thương mại.

*Hệ tầng cơ sở công nghệ:* Thương mại điện tử là hệ quả tất yếu của sự phát triển kỹ thuật số hoá và công nghệ thông tin. Vì vậy, hệ tầng cơ sở công nghệ của thương mại điện tử cùng sự phát triển của kỹ thuật tính toán điện tử và truyền thông điện tử là quan trọng. Để phát triển thương mại điện tử, cơ sở hạ tầng công nghệ phải bao đảm tính hiện hữu, nghĩa là phải có một hệ thống các chuẩn của doanh nghiệp, của quốc gia và các chuẩn này phải phù hợp với quốc tế. Các chuẩn này gắn với hệ thống các cơ sở kỹ thuật và thiết bị ứng dụng của quốc gia như một phân hệ của hệ thống mạng toàn cầu. Cùng với tính hiện hữu, hệ tầng cơ sở công nghệ của thương mại điện tử còn phải đảm bảo tính kinh tế, nghĩa là chi phí của hệ thống thiết bị kỹ thuật và chi phí cho dịch vụ truyền thông phải ở mức hợp lý để bao đảm giá cả của các hàng hoá và dịch vụ thực hiện qua thương mại điện tử không cao hơn so với thương mại truyền thống.

*Hệ tầng cơ sở pháp lý:* Cơ sở pháp lý của thương mại điện tử bao gồm rất nhiều các vấn đề như các đạo luật và chính sách về thương mại điện tử ; các quy định cụ thể về thương mại điện tử trong hệ thống các quy định pháp lý của quốc gia. Để thương mại điện tử phát triển, hệ thống pháp luật của các quốc gia phải từng bước hoàn chỉnh để bảo đảm tính pháp lý của các giao dịch thương mại điện tử, của hợp đồng và các chứng từ điện tử. Hệ tầng cơ sở pháp lý của thương mại điện tử cũng bao gồm các vấn đề thiết hại cho hoạt động thương mại điện tử ở phạm vi quốc gia và quốc tế.

*Hệ tầng cơ sở nhân lực:* Để phát triển thương mại điện tử cần phải có con người có trình độ tương ứng. Con người cấu thành cơ sở nhân lực của thương mại điện tử trước hết là đội ngũ các chuyên gia tin học, thường xuyên cập nhật những kiến thức của công nghệ thông tin và có khả năng đưa

vào ứng dụng trong môi trường và điều kiện kinh doanh cụ thể. Đồng ngũ các chuyên gia công nghệ thông tin và kỹ thuật máy tính sẽ tạo ra cơ sở vật chất kỹ thuật cho thương mại điện tử. Đồng thời với các chuyên gia công nghệ thông tin và kỹ thuật máy tính, các nhà kinh doanh, những người quản lý, các chuyên viên và các khách hàng tiêu thụ cũng phải có khả năng tham gia thương mại điện tử với trình độ nhất định về công nghệ thông tin, về ngoại ngữ và kỹ năng giao dịch trên mạng.

*Nhận thức xã hội:* Phát triển thương mại điện tử cần phải có sự nhận thức sâu sắc của Chính phủ, các nhà quản lý, các nhà hoạch định chiến lược và toàn xã hội về cơ hội phát triển và những lợi ích mà nó mang lại. Chính phủ phải nhận thức được cơ hội và lợi ích của thương mại điện tử để thiết lập môi trường kinh tế, xã hội và pháp lý cho thương mại điện tử. Các nhà quản lý, các nhà hoạch định chiến lược nhận thức được cơ hội và lợi ích của thương mại điện tử về vạch chiến lược phát triển và đề ra giải pháp thích hợp. Từ chiến lược và giải pháp đó mà có kế hoạch phát triển cơ sở hạ tầng công nghệ, có chính sách phát triển. Toàn xã hội nhận thức được cơ hội và lợi ích của thương mại điện tử để tham gia vào hoạt động thương mại điện tử với tư cách là những chủ thể của quá trình.

*Bảo mật an toàn:* Thương mại điện tử có thể bị thiệt hại bởi sự đột nhập từ bên ngoài bởi các hacker để ăn cắp dữ liệu, mạo quan hệ, phá hỏng hệ thống thanh toán, chiếm dụng tiền, v.v... Việc giả mạo địa chỉ Internet (IP Spoofing) để mua hàng bất hợp pháp, phong tỏa các dịch vụ (Denial of service - DOS) làm mất khả năng cung ứng và sử dụng dịch vụ có hiệu quả thường xảy ra trong thương mại điện tử. Do những vấn đề này, thương mại điện tử đặt ra đòi hỏi cao về bảo mật và an toàn. Để đáp ứng yêu cầu kỹ thuật bảo mật như mã hoá, đã ra đời các công nghệ bảo mật như SSL (Secure Sockets Layer), SET (Secure Electronic Transaction) và chữ ký điện tử (Electronic Signature), chữ ký số hoá (Digital signature).

*Sở hữu trí tuệ:* Thương mại điện tử không chỉ vận dụng cho mua bán các sản phẩm hữu hình mà còn áp dụng cho mua bán những sản phẩm vô hình, chuyên giao tri thức và kinh nghiệm, chia sẻ các dữ liệu, chuyển giao công nghệ, các sản phẩm phần mềm. Tất cả những sản phẩm này có chung một đặc điểm là dễ nhân bản và khó giữ bản quyền. Vì vậy, một vấn đề đặt

ra là bảo vệ sở hữu trí tuệ và bản quyền các thông tin trong thương mại điện tử để bảo vệ quyền sở hữu và sử dụng tài sản cho người mua và quyền được hưởng lợi của người bán. Nếu không giải quyết tốt vấn đề này, tình trạng những người chủ thực sự của các tài sản vô hình sẽ bị vi phạm quyền lợi, hàng hoá thông tin sẽ không được chia sẻ bằng con đường mua bán trên thị trường mà thông qua những cách thức gian lận.

*Bảo vệ người tiêu dùng:* Trong thương mại điện tử, trước khi mua, khách hàng không có điều kiện để đánh giá trực quan sản phẩm hàng hoá hay dịch vụ. Những thông tin về hàng hoá và dịch vụ mà khách hàng nhận được để đánh giá các phương án khi đi đến quyết định mua đều dựa trên dữ liệu các trang Web do người bán cung cấp. Điều này dẫn đến một vấn đề là phải có những quy định và tổ chức để bảo vệ lợi ích người tiêu dùng. Các tổ chức và quy định bảo vệ người tiêu dùng cần xây dựng trên các phương diện đảm bảo chất lượng hàng hoá và dịch vụ, tăng độ tin cậy của các dữ liệu, bảo đảm bí mật các thông tin.

## 5. CÁC MÔ HÌNH THƯƠNG MẠI ĐIỆN TỬ

Dựa vào bản chất của giao dịch, thương mại điện tử có thể thực hiện theo các mô hình sau đây:

Thương mại điện tử giữa doanh nghiệp với khách hàng: Mô hình giao dịch này được thực hiện thông qua các mẫu biểu điện tử, thư điện tử, điện thoại, trang Web với các hình thức chủ yếu:

- Sưu tầm thông tin về sản phẩm hàng hoá và dịch vụ (*trên trang Web*).
- Đặt hàng.
- Thanh toán các khoản chi tiêu mua sắm hàng hoá và dịch vụ.
- Cung cấp các hàng hoá và dịch vụ trực tuyến cho khách hàng bao gồm hàng hoá, dịch vụ du lịch, dịch vụ văn phòng, chăm sóc sức khỏe, tư vấn, giải trí.

*Thương mại điện tử giữa doanh nghiệp với doanh nghiệp:* Mô hình giao dịch này chiếm tỷ trọng chủ yếu trong thương mại điện tử. Sử dụng các trang Web, các phương tiện thông tin hiện đại, các tiêu chuẩn về mã vạch, trao đổi dữ liệu điện tử (EDI) để thực hiện hai hình thức giao dịch cơ bản:

– Giao dịch thông tin giữa các tổ chức (IOS), các giao dịch trên thị trường điện tử.

– Trao đổi dữ liệu về quản lý tài chính, nhân sự, marketing và hậu cần sản xuất giữa các doanh nghiệp.

*Thương mại điện tử giữa khách hàng với khách hàng:* Mô hình này được thực hiện thông qua các hình thức mua bán trực tiếp giữa các khách hàng thông qua các trang Web cá nhân, điện thoại, thư điện tử như:

- Bán các tài sản cá nhân trên mạng.
- Dịch vụ quảng cáo trên Internet.
- Thực hiện các dịch vụ tư vấn cá nhân, chuyển giao tri thức trên mạng.
- Các cuộc bán đấu giá do một hoặc một số cá nhân thực hiện trên mạng.
- Thương mại ngân hàng điện tử.
- Các cá nhân sử dụng trang Web riêng để quảng cáo hàng hóa và dịch vụ để bán.

*Thương mại điện tử giữa khách hàng với doanh nghiệp:* Mô hình này bao gồm hình thức các cá nhân bán sản phẩm hoặc dịch vụ cho doanh nghiệp hoặc một số cá nhân hợp tác với nhau để thực hiện các giao dịch kinh doanh với các doanh nghiệp.

*Thương mại điện tử giữa các tổ chức phi kinh doanh:* Càng ngày càng có nhiều tổ chức phi kinh doanh tham gia vào thương mại điện tử như các cơ quan khoa học, các tổ chức phi lợi nhuận, các tổ chức tôn giáo, tổ chức xã hội, các cơ quan của chính phủ. Các hình thức thương mại điện tử giữa các tổ chức phi kinh doanh bao gồm G2G (Government to government), G2B (Government to Business), G2C (Government to customer), A2A (Administration to administration).

*Thương mại điện tử trong nội bộ doanh nghiệp* (Intrabusiness electronic commerce – Intrabusiness EC): Mô hình thương mại điện tử này bao gồm tất cả các hoạt động nội bộ doanh nghiệp thường được thực hiện trên Intranet. Đó là các hoạt động trao đổi hàng hóa, dịch vụ hoặc thông tin bao gồm từ bán hàng hóa hoặc dịch vụ của công ty cho người lao động của doanh nghiệp đến đào tạo trực tuyến của công ty.

## **6. CÁC CÔNG ĐOẠN CỦA MỘT GIAO DỊCH MUA BÁN TRÊN MẠNG**

Các công đoạn của một giao dịch mua bán trên mạng có sáu công đoạn sau:

Khách hàng từ một máy tính tại một nơi nào đó, điền những thông tin thanh toán và địa chỉ liên hệ vào đơn đặt hàng (Order Form) của Website bán hàng (còn gọi là Website thương mại điện tử). Doanh nghiệp nhận được yêu cầu mua hàng hóa hay dịch vụ của khách hàng và phản hồi xác nhận tóm tắt lại những thông tin cần thiết như mặt hàng đã chọn, địa chỉ giao nhận và số phiếu đặt hàng...

Khách hàng kiểm tra lại các thông tin và kích (click) vào nút (button) "đặt hàng", từ bàn phím hay chuột (mouse) của máy tính để gửi thông tin trả về cho doanh nghiệp.

Doanh nghiệp nhận và lưu trữ thông tin đặt hàng đồng thời chuyển tiếp thông tin thanh toán (số thẻ tín dụng, ngày đáo hạn, chủ thẻ ...) đã được mã hoá đến máy chủ (Server, thiết bị xử lý dữ liệu) của Trung tâm cung cấp dịch vụ xử lý thẻ trên mạng Internet. Với quá trình mã hoá các thông tin thanh toán của khách hàng được bảo mật an toàn nhằm chống gian lận trong các giao dịch (doanh nghiệp sẽ không biết được thông tin về thẻ tín dụng của khách hàng).

Khi Trung tâm xử lý thẻ tín dụng nhận được thông tin thanh toán, trung tâm sẽ giải mã thông tin và xử lý giao dịch phía sau bức tường lửa (FireWall) và tách rời mạng Internet (off the Internet), nhằm mục đích bảo mật tuyệt đối cho các giao dịch thương mại, định dạng lại giao dịch và chuyển tiếp thông tin thanh toán đến ngân hàng của doanh nghiệp (Acquirer) theo một đường dây thuê bao riêng (một đường truyền số liệu riêng biệt).

Ngân hàng của doanh nghiệp gửi thông điệp điện tử yêu cầu thanh toán (authorization request) đến ngân hàng hoặc công ty cung cấp thẻ tín dụng của khách hàng (Issuer) và tổ chức tài chính này sẽ phản hồi là đồng ý hoặc từ chối thanh toán đến trung tâm xử lý thẻ tín dụng trên mạng Internet.

Trung tâm xử lý thẻ tín dụng trên Internet sẽ tiếp tục chuyển tiếp những thông tin phản hồi trên đến doanh nghiệp, tùy theo đó doanh nghiệp thông báo cho khách hàng được rõ là đơn đặt hàng sẽ được thực hiện hay không.

Toàn bộ thời gian thực hiện một giao dịch qua mạng từ bước 1 -> bước 6 được xử lý trong khoảng 15 - 20 giây.

*Authorization number là gì?* Đây là mã số xác nhận. Sau khi kiểm tra thẻ tín dụng đã hợp lệ hay chưa, ngân hàng người mua sẽ gửi mã số xác nhận đồng ý chi trả cho doanh nghiệp kèm theo thông số về đơn đặt hàng.

*PSP là gì?* PSP là viết tắt của các từ Processing Service Provider, tức là nhà cung cấp dịch vụ xử lý thanh toán qua mạng.

*Merchant Account là gì?* Merchant Account là tài khoản thanh toán của các doanh nghiệp khi tham gia thương mại điện tử, nó cho phép chuyên tiền vào tài khoản của doanh nghiệp hay hoàn trả lại tiền thu được cho khách hàng nếu giao dịch bị hủy bỏ vì không đáp ứng được những yêu cầu thỏa thuận nào đó giữa người bán và người mua (ví dụ như chất lượng sản phẩm) thông qua bán hàng hóa hoặc dịch vụ trên mạng Internet.

Merchant Account phải được đăng ký tại các ngân hàng/ tổ chức tín dụng cho phép doanh nghiệp nhận được các khoản thanh toán bằng thẻ tín dụng.

*Đầu tư cho Thương mại điện tử có tốn kém không?*

Tất nhiên đầu tư là phải tốn kém. Nhưng Thương mại điện tử mang lại những hiệu quả lớn hơn nhiều lần so với chi phí phải bỏ ra nếu bạn biết đầu tư đúng cách.

*Dùng Thương mại điện tử có phải là giải pháp tối ưu cho cạnh tranh bán hàng không?*

Cho đến hiện nay, việc áp dụng Thương mại Điện tử là giải pháp tối ưu trong việc cạnh tranh bán hàng. Bạn không cần mất nhiều thời gian cho việc giới thiệu sản phẩm, dịch vụ của bạn vì khách hàng có thể tìm hiểu trên website của bạn.

Bạn sẽ không cần phải mất thời gian, chi phí cho việc vận chuyển, thuyết phục khách hàng. Website tự động sẽ giúp bạn có một chính sách chăm sóc khách hàng chu đáo, đây là một trong những yếu tố quan trọng trong việc giữ khách lâu dài. Công việc duy nhất mà bạn phải làm đó là xây dựng một website có tính tự động hoá cao. Một website đủ mạnh và hiệu quả sẽ giúp bạn qua mặt các đối thủ cạnh tranh.

*Khi các hãng kinh doanh áp dụng Thương mại điện tử, nếu có cạnh tranh nhau thì sự cạnh tranh đó sẽ diễn ra như thế nào?*

Doanh nghiệp của bạn áp dụng Thương mại điện tử thì doanh nghiệp đối thủ cũng có thể áp dụng Thương mại điện tử. Cạnh tranh là điều tất yếu trong kinh doanh, không phân biệt bất cứ ở đâu, lúc nào và đối tượng ra sao. Song nếu bạn áp dụng một cách hiệu quả thì chắc chắn bạn sẽ vượt qua đối thủ cạnh tranh.

*Dùng Thương mại điện tử trong kinh doanh thì tính bảo mật có được đảm bảo không?*

Có nhiều cấp độ bảo mật khác nhau tăng giảm tùy theo mức chi phí mà bạn chấp nhận. Nếu doanh nghiệp của bạn tham gia một cách nghiêm túc vào thương mại điện tử thì nên áp dụng chế độ bảo mật tốt nhất.

## TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Thúc Hải, *Mạng máy tính và các hệ thống mở*, Nhà xuất bản Khoa học - Kỹ thuật, 2003.
- [2]. Robert W. Sebesta, *Programming the Word Wide Web*, Addison Wesley, 2003.
- [3] Huỳnh Quyết Thắng, Vũ Đức Vượng và tập thể, *Lập trình Web, Internet*, Nhà xuất bản Khoa học - Kỹ thuật, 2005
- [4] Trần Văn Hoè, *Giáo trình Thương mại điện tử*, Nhà xuất bản Thông kê, 2006
- [5] Aptech World Wide.

# MỤC LỤC

<b>Chương 1. KHÁI QUÁT VỀ CÔNG NGHỆ WWW .....</b>	5
1. KHÁI NIỆM SIÊU VĂN BẢN .....	5
2. MÁY CHỦ WEB .....	6
3. TRÌNH DUYỆT WEB.....	7
<b>Chương 2. CƠ BẢN VỀ NGÔN NGỮ HTML .....</b>	8
1. GIỚI THIỆU HTML .....	8
1.1. HTML Development .....	9
1.2. Cấu trúc của một tài liệu HTML .....	10
1.3. Sử dụng thẻ <META> .....	13
1.4. Sử dụng ký tự đặc biệt trong tài liệu HTML .....	14
2. SỬ DỤNG CÁC SIÊU LIÊN KẾT .....	15
2.1. Giới thiệu siêu liên kết và URL.....	15
2.2. Sử dụng siêu liên kết .....	18
3. ĐIỀU HƯỚNG QUANH WEBSITE .....	24
4. CÁC THẺ CƠ BẢN .....	26
4.1. Headings (Tiêu đề).....	26
4.2. Các thẻ mức đoạn .....	27
4.3. Thẻ khối <SPAN>, <DIV> .....	30
4.4. Các thẻ định dạng ký tự thường dùng .....	31
4.5. Danh sách .....	33
4.6. Thẻ kè đường ngang: <HR>.....	41
4.7. Sử dụng font .....	42
4.8. Sử dụng màu sắc .....	44
5. SỬ DỤNG HÌNH ẢNH TRONG TÀI LIỆU HTML .....	45
5.1. Chèn ảnh.....	46
6. SỬ DỤNG BẢNG VÀ LỚP .....	48
6.1. Cách tạo bảng .....	49
6.2. Lớp (Layer) .....	59
7. CHÈN MULTIMEDIA VÀO TÀI LIỆU HTML .....	62
7.1. Chèn ảnh động vào (.GIF) vào tài liệu HTML.....	62
7.1. Chèn âm thanh vào tài liệu HTML.....	63
7.3. Chèn Video vào tài liệu HTML.....	65
7.4. Chèn các Java Applets.....	66
8. SỬ DỤNG BIỂU MẪU VÀ KHUNG .....	68
8.1. Giới thiệu biểu mẫu .....	68

8.2. Khung (Frame) .....	84
<b>Chương 3. CÁC STYLE SHEET.....</b>	<b>93</b>
1. DHTML.....	93
1.1. Giới thiệu DHTML .....	94
1.2. Các đặc điểm của DHTML .....	95
2. STYLE SHEETS.....	96
2.1. Khái niệm, chức năng và các thuận lợi của style sheet .....	96
2.2. Quy tắc style sheet.....	99
2.3. Các Selector (selector ) trong style sheet (hay các già lớp).....	101
2.4. Kết hợp, liên kết và chèn một style sheet vào tài liệu HTML .....	108
2.5. Thiết lập thuộc tính trong style sheet .....	110
<b>Chương 4. NỀN TẢNG CƠ BẢN VÀ CÚ PHÁP JAVA SCRIPT.....</b>	<b>111</b>
1. GIỚI THIỆU VỀ JAVA SCRIPT.....	111
1.1. Java Script là gì? .....	111
1.2. Định nghĩa.....	112
1.3. Hiệu ứng và quy tắc Java Script.....	112
1.4. Các công cụ Java Script và IDE, và môi trường thực thi .....	113
1.5. Nhúng Java Script vào trong trang Web.....	114
1.6. Ví dụ đơn giản sử dụng các hộp thông báo và phương thức write .....	121
2. CÁC BIÊN (VARIABLE).....	121
2.1. Khai báo biến .....	122
2.2. Phạm vi của biến .....	122
2.3. Các kiểu dữ liệu.....	123
2.4. Các toán tử .....	128
3. KIẾU TOÁN TỬ .....	134
3.1. Các biểu thức.....	134
3.2. Mảng .....	140
3.3. Các câu lệnh điều kiện .....	145
3.4. Các lệnh vòng lặp .....	146
3.5. Hàm (function) .....	150
4. CÁC ĐỐI TƯỢNG CƠ BẢN TRONG Java Script.....	154
4.1. Các đối tượng Java Script.....	154
4.2. Cây phân cấp của đối tượng .....	155
4.3. Câu lệnh this.....	156
4.4. Câu lệnh for...in .....	158
4.5. Câu lệnh with .....	158
4.6. Câu lệnh new .....	159
4.7. Câu lệnh eval .....	161
4.8. Đối tượng String.....	162

4.9. Đối tượng Math .....	164
4.10. Đối tượng Date .....	166
<b>5. CÁC ĐỐI TƯỢNG CỦA TRÌNH DUYỆT TRONG Java Script.....</b>	<b>170</b>
5.1. Đối tượng Event - Khái niệm .....	170
5.2. Các sự kiện Java Script .....	171
5.3. Trình xử lý sự kiện .....	180
5.4. Đối tượng trình duyệt thông thường.....	182
5.5. Đối tượng History .....	186
5.6. Đối tượng Location .....	187
<b>Chương 5. THIẾT KẾ GIAO DIỆN.....</b>	<b>188</b>
1. MỤC ĐÍCH CỦA WEBSITE CẦN THIẾT KẾ.....	188
1.1. Xác định mục tiêu cơ bản của website .....	188
1.2. Xác định độc giả.....	189
2. CHIÉN LƯỢC THIẾT KẾ.....	190
2.1. Đạo chơi trên Web(browsing).....	190
2.2. Đào tạo .....	191
2.3. Dạy học .....	191
2.4. Giáo dục .....	191
2.5. Tham khảo.....	192
3. THIẾT KẾ GIAO DIỆN.....	192
3.1. Khái quát .....	192
3.2. Trang Web đối lập với thiết kế tài liệu thông thường.....	193
3.3. Các tiền lệ trong in ấn .....	193
3.4. Hãy làm cho các trang web của chúng ta độc đáo.....	194
3.5. Thiết kế giao diện cơ bản .....	196
4. THIẾT KẾ WEBSITE.....	205
4.1. Khái quát .....	205
4.2. Các bước trong tổ chức thông tin .....	205
4.3. Cắt đoạn thông tin .....	206
4.4. Hệ thống phân cấp.....	206
4.5. Các mối quan hệ .....	207
4.6. Chức năng .....	208
4.7. Cấu trúc site .....	209
4.8. Phân cấp .....	211
4.9. Web (mạng nhện).....	211
4.10. Kết luận .....	212
5. TRANG CHỦ (Home Page).....	213
5.1. Đừng lựa chọn - hãy sử dụng cả hai .....	215
5.2. Thời gian quản lý .....	216

5.3. Trang chủ, menu và các submenu .....	216
5.4. Bản kê "các site liên quan khác" .....	216
5.5. Thư viện, phụ lục, chỉ số .....	217
5.6. Các vấn đề thường gặp - các trang FAQ .....	217
<b>6. THIẾT KẾ INTRANET .....</b>	<b>217</b>
6.1. Các site bên ngoài .....	218
6.2. Các site bên trong .....	218
6.3. Các tiêu chuẩn thiết kế .....	218
6.4. Các tiêu chuẩn thiết kế .....	218
6.5. Thiết kế coi trọng người đọc .....	219
<b>Chương 6. THƯƠNG MẠI ĐIỆN TỬ .....</b>	<b>220</b>
1. KHÁI NIỆM VÀ ĐẶC TRUNG CỦA THƯƠNG MẠI ĐIỆN TỬ .....	220
1.1. Khái niệm thương mại điện tử .....	220
1.2. Đặc trưng của thương mại điện tử .....	222
2. SỰ KHÁC BIỆT CỦA THƯƠNG MẠI ĐIỆN TỬ VÀ THƯƠNG MẠI TRUYỀN THÔNG .....	224
2.1. Khác biệt về công nghệ .....	224
2.2. Khác biệt về tiến trình mua bán .....	225
2.3. Khác biệt về thị trường .....	225
3. LỢI ÍCH VÀ TÁC ĐỘNG CỦA THƯƠNG MẠI ĐIỆN TỬ .....	226
3.1. Lợi ích và những vấn đề đặt ra với thương mại điện tử .....	226
3.2. Tác động của thương mại điện tử .....	231
4. CÁC ĐIỀU KIỆN PHÁT TRIỂN THƯƠNG MẠI ĐIỆN TỬ .....	235
5. CÁC MÔ HÌNH THƯƠNG MẠI ĐIỆN TỬ .....	237
6. CÁC CÔNG ĐOẠN CỦA MỘT GIAO DỊCH MUA BÁN TRÊN MẠNG .....	239

*Chịu trách nhiệm xuất bản :*

Chủ tịch HĐQT kiêm Tổng Giám đốc NGÔ TRẦN ÁI

Phó Tổng Giám đốc kiêm Tổng biên tập NGUYỄN QUÝ THAO

*Tổ chức bản thảo và chịu trách nhiệm nội dung :*

Chủ tịch HĐQT kiêm Giám đốc Công ty CP Sách ĐH – DN TRẦN NHẬT TÂN

*Biên tập nội dung và sửa bản in:*

BÙI MINH HIẾN

*Trinh bày bìa :*

BÙI QUANG TUẤN

*Chép bản :*

ĐÀN NGỌC

---

## **GIÁO TRÌNH THIẾT KẾ WEB**

Mã số : 6E010M7 - DAI

- In 1.000 bản (QĐ 74), khổ 16 x 24 cm, tại Xí nghiệp In Hà Tây.

Địa chỉ : 15 Quang Trung, TP Hà Đông.



CÔNG TY CỔ PHẦN SÁCH ĐẠI HỌC - DẠY NGHỀ  
**HEVOBCO**  
 25 HÀN THUYỀN – HÀ NỘI  
 Website : [www.hevobco.com.vn](http://www.hevobco.com.vn)



VƯƠNG MIỆN KIM CƯƠNG  
 CHẤT LƯỢNG QUỐC TẾ

## TÌM ĐỌC SÁCH THAM KHẢO MÔN TIN HỌC CỦA NHÀ XUẤT BẢN GIÁO DỤC

1. Giáo trình Vẽ cơ khí với AutoCAD 2004 Chu Văn Vượng
2. Giáo trình Lập trình Pascal (TCCN) Thạc Bình Cường - Lê Quốc Trung
3. Giáo trình Thiết kế Web (TCCN) Thạc Bình Cường
4. Giáo trình Hướng dẫn sử dụng Photoshop (TCCN) Nguyễn Phú Quang
5. Giáo trình Cad/Cam TS. Phan Hữu Phúc

Bạn đọc có thể mua tại các Công ty Sách - Thiết bị trường học ở các địa phương hoặc các Cửa hàng của Nhà xuất bản Giáo dục :

Tại Hà Nội : 25 Hàn Thuyên; 187B Giảng Võ; 232 Tây Sơn; 23 Tràng Tiền.

Tại Đà Nẵng : Số 15 Nguyễn Chí Thanh; Số 62 Nguyễn Chí Thanh.

Tại Thành phố Hồ Chí Minh : 104 Mai Thị Lựu, Quận 1;

Cửa hàng 451B - 453, Hai Bà Trưng, Quận 3; 240 Trần Bình Trọng – Quận 5.

Tại Thành phố Cần Thơ : Số 5/5, đường 30/4.

Website : [www.nxbgd.com.vn](http://www.nxbgd.com.vn)



8 934980 763087



Giá : 27.000 đ