Đại Học Sư Phạm Tp. Hồ Chí Minh

LẬP TRÌNH WEB PHP

Chương 01: Giới thiệu tổng quan HDJ

Nội dung

- Phần 1: Một số khái niệm
- Phần 2: HTML
- Phần 3: CSS
- Phần 4: JavaScript
- Phần 5: DOM

Nội dung phần 1

1. Tổng quan về thiết kế & lập trình Web
2. Phân loại trang web
3. Các bước chính trong phát triển website
4. Công bố website trên internet

Phần 1. Tổng quan về thiết kế & lập trình Web

1.1. Mạng, giao thức

- Mạng máy tính (Computer Network) Hệ thống các máy tính được kết nối với nhau nhằm trao đổi dữ liệu.
- Giao thức (Protocol)
  - Tập hợp các quy tắc được thống nhất giữa các máy tính trong mạng nhằm thực hiện trao đổi dữ liệu được chính xác
  - Ví dụ: TCP/IP, HTTP, FTP,…

1.2. Địa chỉ IP: IP Address

- Xác định một máy tính trong mạng dựa trên giao thức TCP/IP. Hai máy tính trong mạng có 2 địa chỉ IP khác nhau
- Có dạng x.y.z.t (0 ≤ x, y, z, t ≤ 255)
- Ví dụ: 222.255.77.2
- Đặc biệt: địa chỉ: 127.0.0.1 (địa chỉ loopback) là địa chỉ của chính máy tính đang sử dụng dùng để thử mạng

1.3. Tên miền (Domain Name)

- Là tên được "gắn" với 1 địa chỉ IP.
- Máy chủ DNS thực hiện việc "gắn" (ánh xạ)
- Ở dạng văn bản nên thân thiện với con người
- Được chia thành nhiều cấp, phân biệt bởi dấu chấm (.). Đánh số cấp lần lượt từ phải sang trái bắt đầu từ 1.
- Cấp lớn hơn là con của cấp nhỏ hơn
- Ví dụ: math.hcmup.edu.vn gắn với 222.255.77.2 trong đó:
  - vn: Nước Việt Nam (Cấp 1)
  - edu: Tổ chức giáo dục (Cấp 2)
  - hcmup: Tên cơ quan (Cấp 3)
  - math: đơn vị nhỏ trong cơ quan (Cấp 4)
- Đặc biệt: Tên localhost được gắn với 127.0.0.1

1.4. Máy chủ-máy phục vụ: Server

- Là máy tính chuyên cung cấp tài nguyên, dịch vụ cho máy tính khác.
- Thường được cài các phần mềm chuyên dụng để có khả năng cung cấp
- Một máy chủ có thể dùng cho một hay nhiều mục đích. Tên máy chủ thường gắn với mục đích sử dụng. Ví dụ:
  - **File server**
  - **Application server**
  - **Mail server**
  - **Web server**
  - **...**
- Thực tế: các máy chủ có cấu hình cao, khả năng hoạt động ổn định

1.5. Máy khách: Client

- Máy khai thác dịch vụ của máy chủ
- Với mỗi dịch vụ, thường có các phần mềm chuyên biệt để khai thác
- Một máy tính có thể vừa là client vừa là server
- Một máy tính có thể khai thác dịch vụ của chính nó.

1.6. Cổng dịch vụ: Service Port

- Là số ∈ [0; 65535] xác định dịch vụ của máy chủ
- 2 dịch vụ khác nhau chiếm các cổng khác nhau
- Mỗi dịch vụ thường chiếm các cổng xác định, ví dụ:
  - Web: 80
  - FTP: 21

1.7. Địa chỉ tài nguyên: URL (Uniform Resource Locator)

- Tài nguyên: file trên mạng
- URL: Xác định vị trí và cách khai thác file
- giao_thức://địa_chỉ_máy:cổng/đường_dẫn/tên_file
- Ví dụ: http://fit.hcmup.edu.vn:8080/html/test.htm
- Trong trường hợp mặc định, nhiều thành phần có thể bỏ qua:
  - Giao thức, cổng: Được trình duyệt đặt mặc định
  - Tên file: được máy chủ đặt mặc định

1.8. Trang web, web site, World Wide Web

- Trang web (Web page):
  - Là một trang nội dung
  - Có thể được viết bằng nhiều ngôn ngữ khác nhau nhưng kết quả trả về client là HTML
- Web site: Tập hợp các trang web có nội dung thống nhất phục vụ cho một mục đích nào đó
- World Wide Web (WWW): Tập hợp các web site trên mạng internet.

1.9. Web server, Web browser

- **Web server**:
  - Máy phục vụ web
  - Một số phần mềm web server chuyên dụng:
    - **Apache**: mã nguồn mở
    - **Internet** Information Services (IIS): Sản phẩm của Microsoft
- **Web Browser**:
  - Phần mềm chạy trên client để khai thác dịch vụ web
  - Một số Web browser:
    - **Nescape**
    - **Mozilla Firefox**
    - **Internet Explorer** (IE): tích hợp sẵn trong windows
    - ...

2. Phân loại trang web

- **Web tĩnh**:
  - Dễ phát triển
  - Tương tác yếu
  - Sử dụng HTML
  - Người làm web tĩnh thường dùng các công cụ trực quan để tạo ra trang web

Dựa vào công nghệ phát triển, có 2 loại

- **Web động**:
  - Khó phát triển hơn
  - Tương tác mạnh
  - Sử dụng nhiều ngôn ngữ khác nhau
  - Thường phải viết nhiều mã lệnh

3. Các bước chính trong phát triển website

- Đặc tả yêu cầu
- Phân tích
- Thiết kế (dữ liệu, giao diện, xử lý)
- Lập trình
- Kiểm thử

3. Các bước chính trong phát triển website

- Đặc tả yêu cầu
  - Web để làm gì?
  - Ai dùng?
  - Trình độ người dùng?
  - Nội dung, hình ảnh?
- Phân tích
  - Mối liên quan giữa các nội dung?
  - Thứ tự các nội dung?

3. Các bước chính trong phát triển website

- Thiết kế
  - Sơ đồ cấu trúc website
  - Giao diện
  - Tĩnh hay động
  - CSDL
  - Nội dung từng trang
  - Liên kết giữa các trang
- Lập trình
  - Cấu trúc thư mục
  - Các module dùng chung
  - …

3. Các bước chính trong phát triển website

- Kiểm thử
  - Kiểm tra trên nhiều trình duyệt
  - Kiểm tra trên nhiều loại mạng
  - Kiểm tra tốc độ
  - Kiểm tra các liên kết
  - Thử các lỗi bảo mật
  - …



4. Công bố website trên Internet

Để công bố Website trên Internet, ta cần các điều kiện cần thiết sau:

4.1. Xây dựng website

4.2. Đăng ký tên miền

4.3. Hosting

4.4. Duy trì website

4.5. Phát triển website

4.6. Quảng bá website

4.1. Xây dựng website

- Loại thông tin
  - Web tĩnh, động
  - Portal
- Giá thành
  - Web tĩnh: Tính theo các kiểu trang
    - Trang đơn giản: 70 – 150.000đ/trang
    - Trang hiệu ứng hình ảnh tốt: 150 – 400.000đ/trang
  - Web động: Tính theo các mục, các khối chức năng
    - Thiết kế CSDL
    - Các chức năng phía user: đưa tin, phân loại, tìm kiếm…
    - Các chức năng phía Admin: Đăng nhập, xem/thêm/sửa/xóa tin bài, báo cáo, thống kê…
    - Từ 5 triệu trở lên. (Thông dụng: 10-30 triệu)

4.1. Xây dựng website

- Có nên đăng ký tên miền, thiết kế và duy trì website hay không?
- Nếu có, đăng ký tên miền với tên thế nào, thể loại gì, theo hệ thống của Việt Nam hay theo các hệ thống tên miền bên ngoài
- Có tự Host website của mình không?
- Để phục vụ website, có các phương án dành cho?
  - Nhân sự
  - Kinh phí
  - Cơ chế tổ chức, hoạt động
  - Quy trình làm việc

4.2. Đăng ký tên miền

- Xác định tên
  - Tên tiếng Việt
  - Tên giao dịch tiếng Anh
  - Tên viết tắt
- Xác định nơi đăng ký
- Đăng ký tên miền càng sớm càng tốt
  - Thủ tục đơn giản, nhanh chóng
  - Kinh phí rẻ
    - Việt Nam: 450.000 – 480.000/năm
    - Nước ngoài: 8 – 12USD

4.3. Hosting

- Xác định môi trường vận hành của website
  - Máy chủ Windows
    - Support ASP, PHP…, SQL Server, MySQL…
    - Đắt hơn máy chủ Linux
  - Máy chủ Linux
    - Support PHP, JSP…, MySQL…
    - Rẻ hơn máy chủ Windows
- Xác định dung lượng thực tế của website, khả năng sẽ mở rộng
- Xác định băng thông, các dịch vụ đảm bảo an toàn, an ninh, backup dữ liệu…

4.4. Duy trì website

- Cập nhật thông tin
  - Web tĩnh:
    - Upload Webpage thông qua Web Browser
    - Upload Webpage thông qua FTP program (Cute FTP, FTP Voyager,…)
  - Web động
    - Form cập nhật CSDL nếu Site có kết nối CSDL

4.5. Phát triển website

- Các chiến lược marketing
  - Sử dụng thư điện tử
  - Đầu tư quảng cáo 1 đợt trên các phương tiện truyền thông (Báo, đài, Tivi…)
- Liên kết với các site cùng loại
  - Trao đổi banner
  - Giới thiệu lẫn nhau.

4.6. Quảng bá website

- Quảng bá Website
  - Đăng ký Website vào các máy tìm kiếm trong nước và thế giới (search engine)
    - Vietnam Searchengine: Panvietnam, vinaseek…
    - Global Searchengine: google, altavista, hotboot…
  - Nâng cao vị trí của Website trong hệ thống xếp hạng Website thế giới.
    - Google rank (the important of website: 1-10)
    - Alexa rank: Traffic ranking of website.
- Nâng tầm phát triển Website
  - Tự động hoá dần các chức năng của Website.
  - Biến Website thành một môi trường kinh doanh thực sự hiệu quả 24/24 trên Internet.

Phần 2: HTML

Nội dung phần 2

1. Ví dụ đầu tiên về trang web
2. Giới thiệu chung về HTML
3. Đặc điểm của HTML
4. Trình bày văn bản trong HTML
5. Multimedia
6. Form trên trang web
7. Frame

1. Ví dụ đầu tiên về trang web – Cấu trúc

```
<html>
<head>
    <title>
        Lập trình web
    </title>
</head>
<body>
Chào mừng bạn đến với <b>HTML</b>!
</body>
</html>
```

1. Ví dụ đầu tiên về trang web – Cấu trúc

- **Thử nghiệm**:
  - Mở trình duyệt web (IE)
  - Vào File/Open, chọn file Welcome.HTML vừa ghi
  - Nhấn OK → Có kết quả như hình bên
- **Thay đổi**:
  - Quay lại Notepad, sửa lại nội dung trang web rồi ghi lại
  - Chuyển sang IE, nhấn nút Refresh (F5) → thấy kết quả mới

2. Giới thiệu chung về HTML

- **HTML = HyperText Markup Language ngôn ngữ đánh dấu siêu văn bản, là ngôn ngữ dùng để viết trang web.**

- **Do Tim Berner Lee phát minh và được W3C (World Wide Web Consortium) đưa thành chuẩn năm 1994.**

3. Đặc điểm của HTML

- **HTML** sử dụng các thẻ (tags) để **định dạng** dữ liệu
- **HTML** không phân biệt chữ hoa, chữ thường
- Các trình duyệt thường không báo lỗi cú pháp HTML.
- Khi viết sai cú pháp thì trình duyệt sẽ hiển thị không đúng với dự định.
- Có nhiều thẻ, mỗi thẻ có 1 tên và mang ý nghĩa khác nhau.

3. Đặc điểm của HTML

- **Một thẻ có thể có các thuộc tính nhằm bổ sung tác dụng cho thẻ**
- **Mỗi thuộc tính có tên thuộc tính (tên_TT)**
- **Viết thẻ có thuộc tính:**

  <tên_thẻ tên_TT1="giá_trị1" tên_TT2="giá_trị2"…>

- **Chú ý:**
  - Có thể thay đổi thứ tự, số lượng các thuộc tính mà không gây ra lỗi cú pháp
  - Sự hỗ trợ các thẻ, thuộc tính ở mỗi trình duyệt là khác nhau. Chỉ giống nhau ở các thẻ, thuộc tính cơ bản.
  - Thẻ đóng của thẻ có thuộc tính vẫn viết bình thường (**</tên_thẻ>**)

3. Đặc điểm của HTML

- Có 2 loại thẻ: thẻ đóng và thẻ mở
- Cách viết thẻ:
  - Thẻ mở: <tên_thẻ>

    Ví dụ: <u>, , <img>…
  - Thẻ đóng tương ứng: </tên_thẻ>

    Ví dụ: </u>,

Chú ý:

- Luôn có thẻ mở nhưng có thể không có thẻ đóng tương ứng. Ví dụ: <img> không có thẻ đóng
- Các thẻ có thể lồng nhau, nhưng không đan xen lẫn nhau.

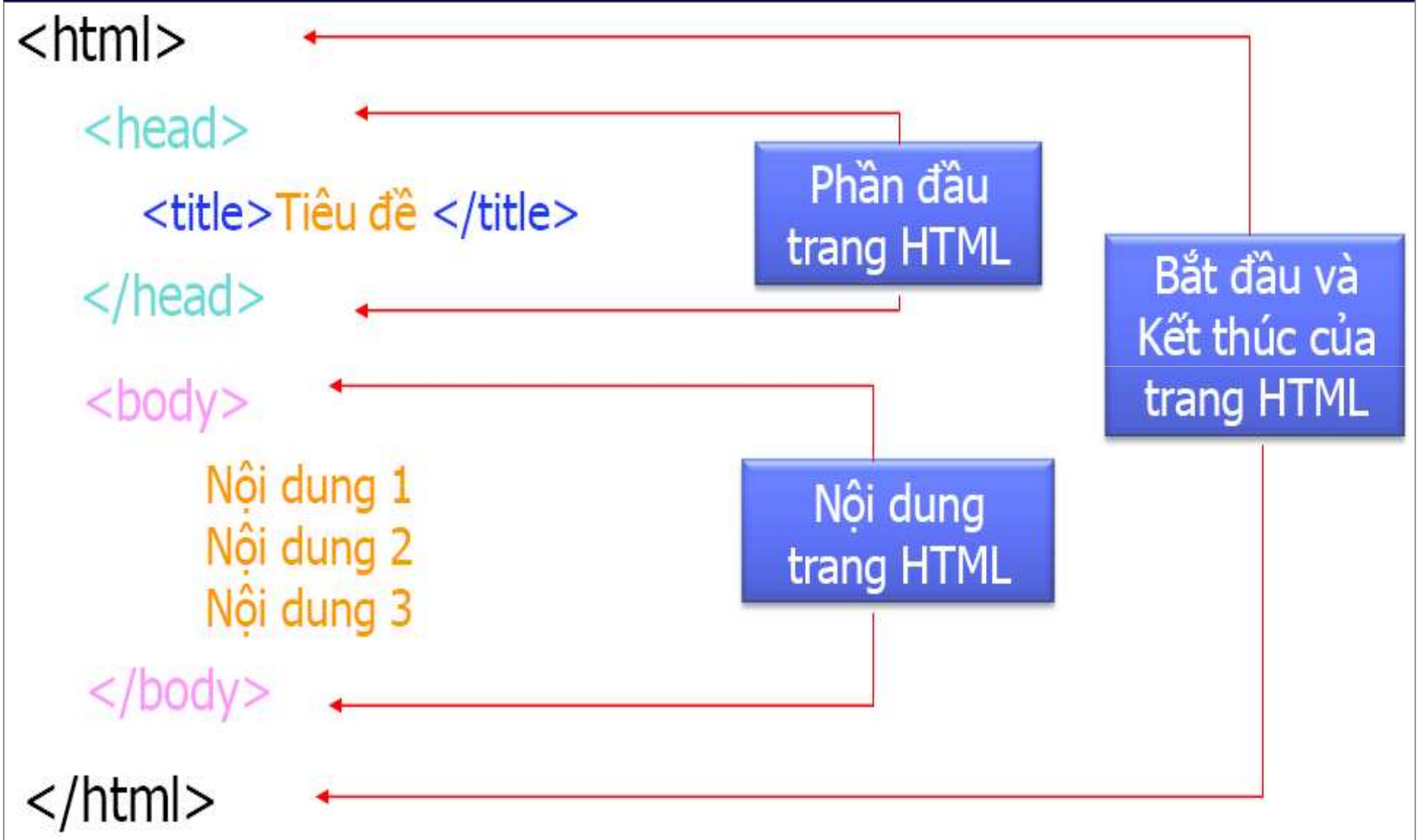
3. Đặc điểm của HTML

- Tập tin HTML có phần mở rộng (đuôi) là .HTM hoặc .HTML là tập tin thuần văn bản (plain text)
- Có thể soạn thảo file HTML bằng bất cứ trình soạn thảo "văn bản thuần" nào (Notepad, EditPlus, Notepad++, …)
- Trình hỗ trợ soạn thảo HTML (trực quan, code):
  - Microsoft FrontPage
  - Adobe Dreamweaver CS3/CS4
  - …

4. TRÌNH BÀY TÀI LIỆU TRONG HTML

- 4.1 Thẻ thể hiện cấu trúc tài liệu
- 4.2 Thẻ META
- 4.3 Thẻ định dạng khối
- 4.4 Thẻ định dạng danh sách
- 4.5 Thẻ định dạng ký tự
- 4.6 Liên kết
- 4.7 Bảng
- 4.8 Một số lưu ý

4.1. Thẻ thể hiện cấu trúc tài liệu



4.1. Thẻ thể hiện cấu trúc tài liệu

- <html></html>: Định nghĩa phạm vi của văn bản HTML
- <head></head>: Định nghĩa các mô tả về trang HTML. Thông tin trong tag này không được hiển thị trên trang web
- <title></title>: Mô tả tiêu đề trang web
- <body></body>: Xác định vùng thân của trang web, nơi chứa các thông tin

4.1. Thẻ thể hiện cấu trúc tài liệu

Thuộc tính của Body

BACKGROUND	Đặt một ảnh nào đó làm ảnh nền (background) cho văn bản. Giá trị của tham số này (phần sau dấu bằng) là URL của file ảnh. Nếu kích thước ảnh nhỏ hơn cửa sổ trình duyệt thì toàn bộ màn hình cửa sổ trình duyệt sẽ được lát kín bằng nhiều ảnh.
BGCOLOR	Đặt mầu nền cho trang khi hiển thị. Nếu cả hai tham số BACKGROUND và BGCOLOR cùng có giá trị thì trình duyệt sẽ hiển thị mầu nền trước, sau đó mới tải ảnh lên phía trên.
TEXT	Xác định màu chữ của văn bản, kể cả các đề mục.
ALINK VLINK LINK	Xác định màu sắc cho các siêu liên kết trong văn bản. Tương ứng, alink (active link) là liên kết đang được kích hoạt - tức là khi đã được kích chuột lên; vlink (visited link) chỉ liên kết đã từng được kích hoạt;

4.2. Thẻ META

- Đặt ở giữa <head>…</head>
- Thường dùng quy định thuộc tính cho trang web
- Có tác dụng lớn với Search Engine
- Có 2 cách viết thẻ <meta>:

```
<META NAME="name" CONTENT="content">

<META HTTP-EQUIV="name" CONTENT="content">
```

4.2. Thẻ META

Một số thẻ Meta thông dụng

- `<META NAME="description" content="">`
- `<META NAME="keywords" content="">`
- `<META NAME="author" CONTENT="author's name">`
- `<META HTTP-EQUIV="refresh" CONTENT="delay;url=new url">`
- `<META HTTP-EQUIV="expires" CONTENT="date">`
- `<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">`

4.2. Thẻ META

Tự động chuyển hướng trang web

- Tự động chuyển hướng trang web sang trang web khác (url) sau một khoảng thời gian t (tính theo giây)
- Cú pháp

```
<head>
        <META HTTP-EQUIV="refresh" CONTENT="t; URL=url">
</head>
```

4.3. Thẻ định dạng khối tài liệu

DIV	Thẻ <DIV> được sử dụng để đóng khối văn bản.
P	Thẻ <P> được sử dụng để định dạng một đoạn văn bản.
H1,H2,H3,…,H6	Xác định 1 trong 6 cấp của tựa đề (heading)
BR	Ngắt dòng
PRE	Văn bản nằm trong thẻ này sẽ được trình bày như nguyên thủy của nó khi gõ vào.

HTML – thẻ

- **Chức năng**
  - Xác định khối văn bản
  - Chia tài liệu thành những khối riêng biệt.
  - Hỗ trợ canh lề, định dạng style
- **Thuộc tính**

  `align, class, dir, id, lang, nowrap, onClick, onDblClick, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, title`

4.4. Thẻ định dạng danh sách

- **OL**
  - Danh sách được sắp xếp thứ tự
  - Hỗ trợ thuộc tính start cho phép chọn giá trị khởi đầu

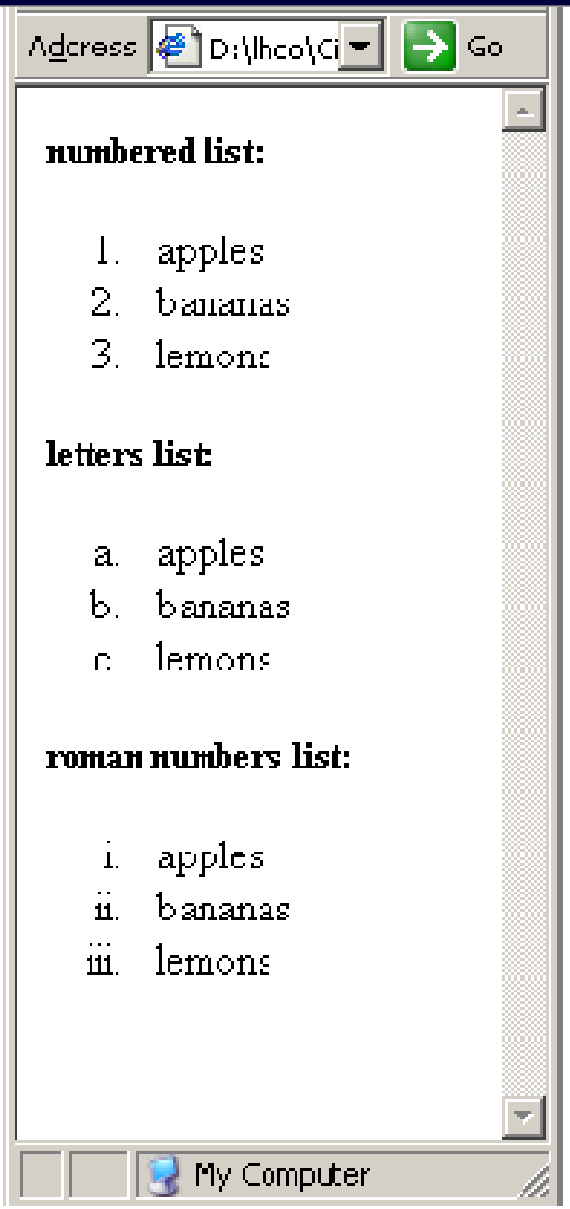
```
<OL TYPE=1/a/A/i/I>
    <LI>Mục 1
    <LI>Mục 2
    <LI>Mục 3
</OL>
```

- **UL**
  - Danh sách không sắp xếp
  - Thẻ **<LI>** có thuộc tính **TYPE** có các giá trị
    - **disc** (chấm tròn đậm);
    - **circle** (vòng tròn);
    - **square** (hình vuông)

```
<UL>
    <LI>Mục 1
    <LI>Mục 2
    <LI>Mục 3
</UL>
```

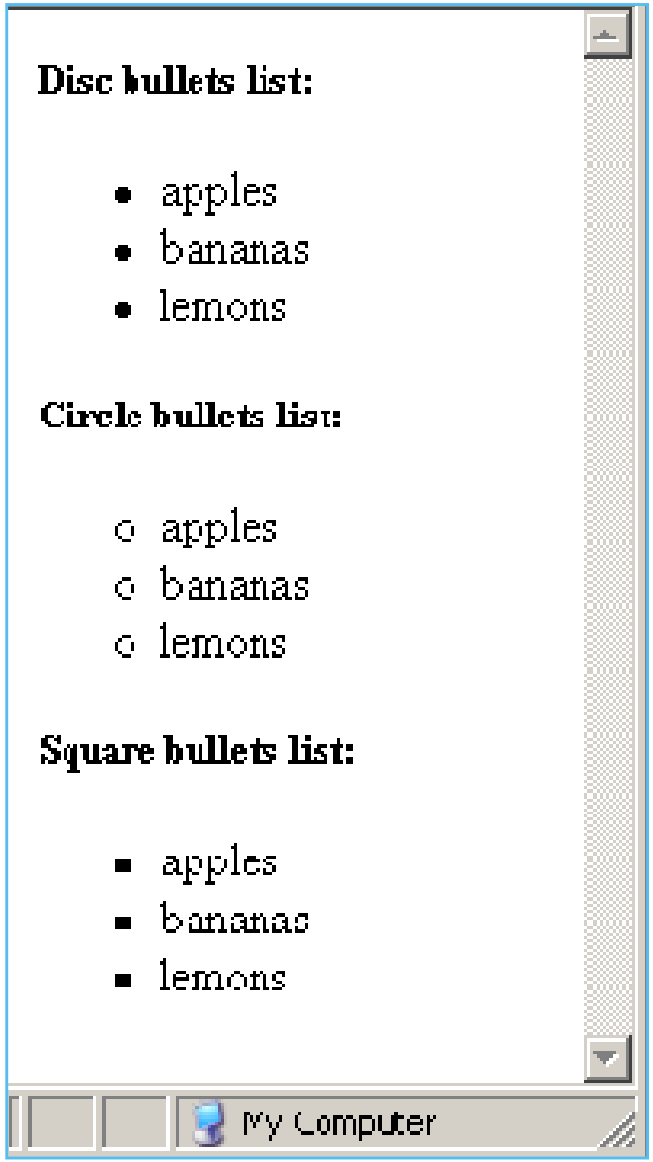
4.4. Thẻ định dạng danh sách

```
<html>
<head>
<title> danh sach co thu tu </title></head>
<body>
<h5>numbered list:</h5>
<ol>
    <li>apples</li>
    <li>bananas</li>
    <li>lemons</li>
</ol>
<h5>letters list:</h5>
<ol type="a">
    <li>apples</li>
    <li>bananas</li>
    <li>lemons</li>
</ol>
<h5>roman numbers list:</h5>
<ol type="i">
    <li>apples</li>
    <li>bananas</li>
    <li>lemons</li>
</ol>
</body>
</html>
```



4.4. Thẻ định dạng danh sách

```
1  <html>
2  <head>
3  <title> danh sach khong thu tu </title></head>
4  <body>
5  <h5>Disc bullets list:</h5>
6  <ul type="disc">
7      <li>apples</li>
8      <li>bananas</li>
9      <li>lemons</li>
10 </ul>
11 <h5>Circle bullets list:</h5>
12 <ul type="circle">
13     <li>apples</li>
14     <li>bananas</li>
15     <li>lemons</li>
16 </ul>
17 <h5>Square bullets list:</h5>
18 <ul type="square">
19     <li>apples</li>
20     <li>bananas</li>
21     <li>lemons</li>
22 </ul>
23 </body>
24 </html>
```



4.5. Thẻ định dạng ký tự

<B></B><STRONG></STRONG>	In chữ đậm
<I></I><EM></EM>	In chữ nghiêng
<U></U>	In chữ gạch chân
<S></S><STRIKE></STRIKE>	In chữ bị gạch ngang.
<BIG> ... </BIG>	In chữ lớn hơn kích thước font hiện thời lên một. Các thẻ <BIG> lồng nhau tạo ra hiệu ứng chữ tăng dần. Đối với mỗi trình duyệt có giới hạn về kích thước đối với mỗi font chữ, vượt quá giới hạn này, các thẻ <BIG> sẽ không có ý nghĩa.
<SMALL> ... </SMALL>	In chữ nhỏ hơn bình thường bằng cách giảm kích thước font hiện thời đi một. Tương tự như thẻ BIG
^{...}	Định dạng chỉ số trên (SuperScript)
_{...}	Định dạng chỉ số dưới (SubScript)
<FONT> ... </FONT>	Chọn font chữ, size hoặc color, kích thước có thể là tuyệt đối (nhận giá trị từ 1 đến 7) hoặc tương đối (+2,-4...) so với font chữ hiện tại.

4.6. Liên kết

```
<a href="URL" target='.....'> Linked content </a>
```

- **Thuộc tính:**
  - **href=“đích liên kết”**: Nếu trong cùng web nên sử dụng đường dẫn tương đối.
  - **target=“tên cửa sổ đích”,** tên cửa sổ phân biệt chữ hoa/thường
    - **name**: tải trang web vào frame có tên **name**
    - **_blank**: tải trang web vào cửa sổ mới
    - **_parent**: tải trang web vào cửa sổ cha của nó
    - **_self**: tải trang web vào chính cửa sổ hiện hành
    - **_top**: tải trang web vào cửa số cao nhất

4.6. Liên kết

- **Lưu ý:**
  - Liên kết với địa chỉ e-mail thì đặt

    href="mailto:địa_chỉ_e-mail"
  - Thực hiện lệnh JavaScript khi kích chuột vào thì đặt

    href="javascript:lệnh"
- **Liên kết đến trang khác**
  - **Thuộc tính href="url của trang khác"**
  - **Khi click vào liên kết sẽ chuyển đến trang khác**
- **Liên kết trong cùng một trang**
  - **Thuộc tính href="#id của thẻ trong trang"**
  - **Khi click và liên kết sẽ chuyển đến thẻ có "id" được ghi trong thuộc tính id của thẻ.**

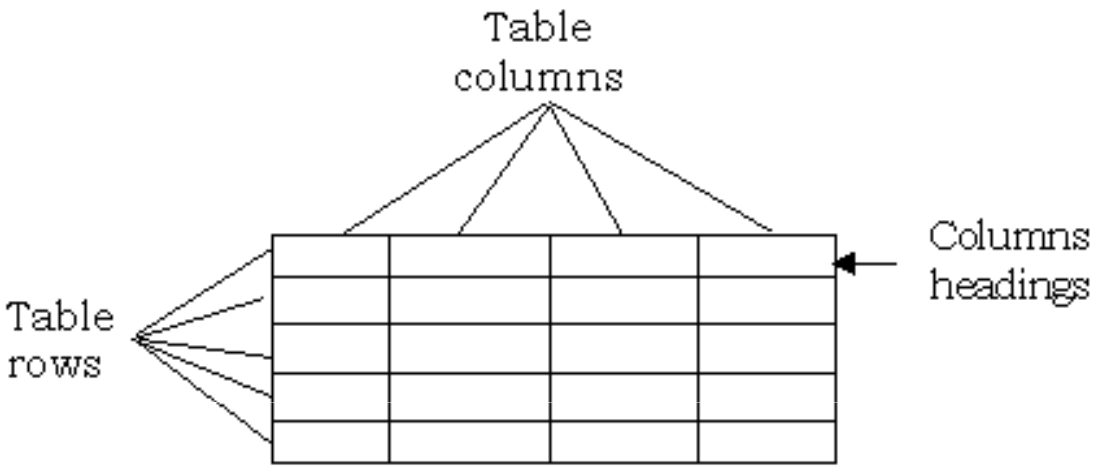
4.6. Liên kết

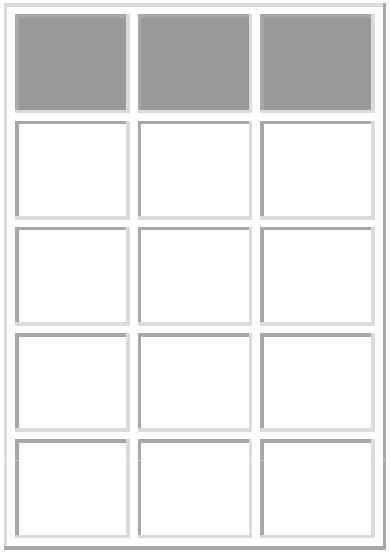
```
<a href="URL" target='.....'> Linked content </a>
```

- Địa chỉ URL phân làm 2 loại :
  - **Địa chỉ tuyệt đối**: Là vị trí tuyệt đối so với Mạng Internet
  - **Địa chỉ tương đối**: Là vị trí tương đối so với trang web hiện hành đang chứ liên kết.

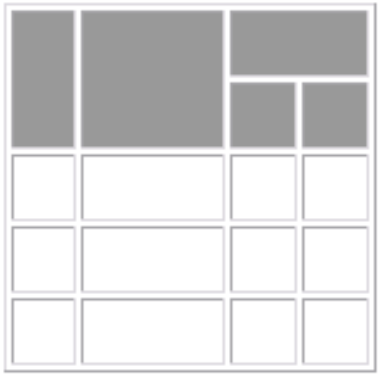
Ký hiệu	Ý nghĩa
/	Trở về thư mục gốc của website
./	Thư mục hiện tại của trang web sử dụng link (mặc định)
../	Quay ra thư mục cha / đi ngược lại 1 cấp thư mục

4.7 Bảng - Cấu trúc của 1 bảng





Bảng gồm nhiều dòng, một ***dòng gồm nhiều ô***, và ***chỉ có ô mới chứa dữ liệu*** của bảng.



4.7 Bảng

STT	Tên thẻ	Mô tả - Ví dụ
1	<table>…</table>	Khởi tạo 1 bảng
2	<tr>…</tr>	Tạo 1 dòng (thẻ <tr> phải nằm trong thẻ <table>)
3	<th>…<th>	Tạo 1 ô tiêu đề (thẻ <th> phải nằm trong thẻ <tr>)
4	<td>…</td>	Tạo 1 ô dữ liệu (thẻ <td> phải nằm trong thẻ <tr>)

- **Tổng số thẻ <td> và <th> bằng số ô của bảng. Dòng có bao nhiêu ô thì có bấy nhiêu thẻ <td> và/hoặc <th> nằm trong cặp thẻ <tr>…</tr> tương ứng.**
- **Để có được một ô trống trong bảng (ô không có dữ liệu) thì cần đặt nội dung ô là: **

4.7. Bảng

- Thuộc tính của thẻ <table>:
  - border="số": kích thước đường viền. Đặt bằng 0 (mặc định): không có đường viền.
  - width="rộng", height="cao": độ rộng và độ cao của bảng. Có thể đặt theo 2 cách:
    - n: (n là số) Quy định độ rộng, cao là n pixels
    - n%: Quy định độ rộng, cao là n% độ rộng, cao của đối tượng chứa bảng.
  - cellspacing="số": Khoảng cách giữa 2 ô liên tiếp
  - cellpadding="số": Khoảng cách từ góc ô đến nội dung ô
  - bgcolor="màu": màu nền của bảng
  - background="địa_chỉ_ảnh": Địa chỉ của file ảnh làm nền cho bảng. Nên sử dụng đường dẫn tương đối nếu có thể.

4.7. Bảng

- Thuộc tính của thẻ <td>,<th>:
  - bgcolor="màu": màu nền của ô
  - background="địa_chỉ_ảnh": Địa chỉ của file ảnh làm nền cho ô. Nên sử dụng đường dẫn tương đối nếu có thể.
  - width="rộng", height="cao": độ rộng và độ cao của ô. Có thể đặt theo 2 cách:
    - n: (n là số) Quy định độ rộng, cao là n pixels
    - n%: Quy định độ rộng, cao là n% độ rộng, cao của bảng.
  - align="căn_lề": cách căn chỉnh dữ liệu trong ô theo chiều ngang: left, right, center, justify.
  - valign="căn_lề_đứng": cách căn chỉnh dữ liệu trong ô theo chiều đứng: top, middle, bottom.
  - colspan="số": số cột mà ô này chiếm (mặc định là 1)
  - rowspan="số": số dòng mà ô này chiếm (mặc định là 1)
  - nowrap: nếu có sẽ làm cho dữ liệu trong ô không tự xuống dòng

4.8. Một số lưu ý

- **Lưu ý:**
  - Mọi khoảng trống, dấu xuống dòng trong HTML được thể hiện trên trang web là 1 khoảng trống duy nhất
  - Để gõ một số ký tự đặc biệt ta phải sử dụng mã:
    - Khoảng trống (trong trường hợp muốn có nhiều hơn 1 ký tự trống): ** **
    - Dấu nhỏ hơn (<) và lớn hơn (>): **<** **>**
    - Dấu ngoặc kép (“): **"**
    - Ký hiệu ©: **©**
    - …
- **Ghi chú trong HTML:**

```
<!-- Đây là ghi chú trong HTML-->
```

5. Multimedia

5.1 Hình ảnh

5.2 Âm thanh

5.3 Video

5.4 Flash

5.5 Applet

5.1. Hình ảnh

- <img> : Không có thẻ đóng
- Các thuộc tính của tag <img>:
  - align: left, right, center
  - src : Đường dẫn đến file hình ảnh
  - alt : Chú thích cho hình ảnh
  - position: Top, Bottom, Middle
  - border : Độ dày nét viền quanh ảnh (default=0)
  - width: độ rộng
  - height: độ cao
- Đặt ảnh nền cho trang web
  - <body background='Image Path'>

5.2. Âm thanh

- <bgsound> : Không có tag đóng
- Thuộc tính của tag <bgsound>
  - src : Đường dẫn đến file âm thanh
  - loop : Số lần lặp (bằng -1 : Lặp vô hạn)
- <bgsound> thường đặt trong tag <head> của trang web.
- Ví dụ: <BGSOUND src='vui.mid' LOOP='1'>

5.3. Video

- Thẻ <embed> không có thẻ đóng
- Thuộc tính
  - width: rộng
  - height: cao
  - src: địa chỉ của tập tin video
- <embed width=400 height=300 src="../video/clock.avi">

5.4. Flash

- Thẻ <embed> không có thẻ đóng
- Thuộc tính
  - width: rộng
  - height: cao
  - src: địa chỉ của tập tin flash
- <embed width=400 height=300 src="../flash/adam.swf">

5.5. Applet

```
<applet
  code="ten_fle.class"
  width="Số"
  height="Số">
</applet>
```

6. FORM TRONG TRANG WEB

1. Giới thiệu về Form(GT, tag, Vd)
2. Các thành phần của Form
3. Một số thuộc tính của form và input
4. Gởi dữ liệu bằng phương thức POST/GET
5. Thẻ <marquee>

6.1. Giới thiệu về Form

- Được dùng để nhận dữ liệu từ phía người dùng
- Giúp gởi yêu cầu của người dùng đến trang xử lý trong ứng dụng web
- Tag <form> dùng để chứa các thành phần khác của form
- Những thành phần nhập liệu được gọi là Form Field
  - text field
  - password field
  - multiple-line text field
  - ……

6.1. Giới thiệu về Form

```
<FORM NAME="..." ACTION="..." METHOD="...">
        <!-- các thành phần của Form -->
</FORM>
```

- Là container chứa các thành phần nhập liệu khác.
- Các thuộc tính của **</FORM>**
  - **NAME**: tên FORM
  - **ACTION**: chỉ định trang web nhận xử lý dữ liệu từ FORM này khi có sự kiện click của button SUBMIT.
  - **METHOD**: Xác định phương thức chuyển dữ liệu (**POST,GET**)

6.1. Giới thiệu về Form

Dangnhap.html

```
<html>
<body>
   <form   Name="Dangnhap"
           Action="./admin/xlDangnhap.php"
           Method="POST" >
           ……..……
   </form>
</body>
</html>
```

6.2. Các thành phần của Form – Nội dung

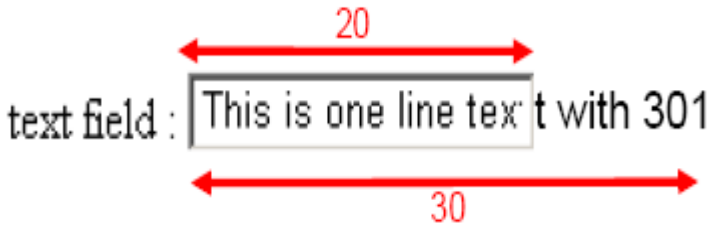
- Text field
- Password field
- Multiple-line text field
- Hidden Text field
- Pull-down menu (Combo box, List box)
- Check box
- Radio button
- File Form Control
- Submit Button, Reset Button, Generalized Button
- Label
- Field Set
- Tiện ích form

6.2. Các thành phần của Form – Text field

- Dùng để nhập một dòng văn bản
- Ví dụ:

```
<INPUT
        TYPE            = "TEXT"
        NAME            = string
        READONLY
        SIZE            = variant
        MAXLENGTH       = long
        TABINDEX        = integer
        VALUE           = string
        .....
>
```

```
<input type="text" name="txtName"
    value="This is one line text with 301"
    size="20" maxlength="30">
```

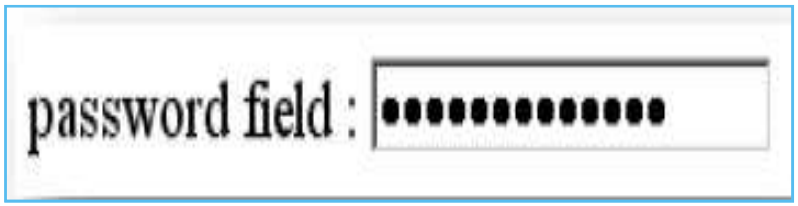


6.2. Các thành phần của Form – Password field

- Dùng để nhập mật khẩu
- Ví dụ:

```
<INPUT
        TYPE            = "PASSWORD"
        NAME            = string
        READONLY
        SIZE            = variant
        MAXLENGTH       = long
        TABINDEX        = integer
        VALUE           = string
        ..........
>
```

```
<input type="password" name="txtPass"
    value="123456asdfgh"
    size="20" maxlength="30">
```

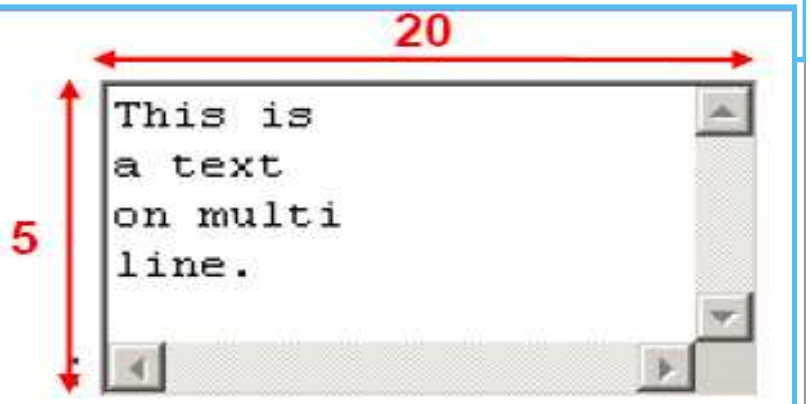


6.2. Các thành phần của Form – Multiline text

- Dùng để nhập văn bản nhiều dòng
- Ví dụ

```
<TEXTAREA
      COLS          = long
      ROWS          = long
      DISABLED
      NAME          = string
      READONLY
      TABINDEX      = integer
      WRAP          = OFF | PHYSICAL | VIRTUAL>
    ..............
</TEXTAREA>
```

```
<textarea cols="20" rows="5" wrap="off">
        This is a text on multiline.
</textarea>
```



6.2. Các thành phần của Form – Hidden text

- Dùng để truyền 1 giá trị của thuộc tính value khi form được submit
- Không hiển thị ra trên màn hình

```
<INPUT
        TYPE            = "HIDDEN"
        NAME            = string
        READONLY
        SIZE            = variant
        MAXLENGTH       = long
        TABINDEX        = integer
        VALUE           = string
        ..........
>
```

6.2. Các thành phần của Form – Pull-down

```
<Select name="…">
    <optgroup label="…">
        <option [selected] value="…" >……</option>
        ……
    </optgroup>
    <option [selected] value="…" >……</option>
    ……
</select>
```

Xem code ở slide tiếp

6.2. Các thành phần của Form – Combo box

List box:
<select multiple>

```
<html>
<body>
combo box:
<select name="DSSoftware">
    <optgroup label="Multimedia">
        <option value="WM10">Window Media 10</option>
        <option value="JA9">Jet Audio 9</option>
    </optgroup>
    <optgroup label="Operation System">
        <option value="WXP">Windows XP</option>
        <option value="WXPSP2">Windows XP SP2</option>
        <option value="WVT">Windows Vista</option>
    </optgroup>
    <option selected value="Office07">Office 2007</option>
</select>
</body>
</html>
```

6.2. Các thành phần của Form – Checkbox

```
<input
        TYPE    = "checkbox"
        NAME    = "text"
        VALUE   = "text"
        [checked]
>
```


```
<html>
<body>
Check box group: <br>
Anh văn: <input type="checkbox" name="Languages[ ]" value="En"><br>
Hoa: <input type="checkbox" name="Languages[ ]" value="Chz" checked><br>
Nhật: <input type="checkbox" name="Languages[ ]" value="Jp"><br>
</body>
</html>
```

6.2. Các thành phần của Form – Radio

```
<input
        TYPE    = "radio"
        NAME    = "text"
        VALUE   = "text"
         [checked]
>
```

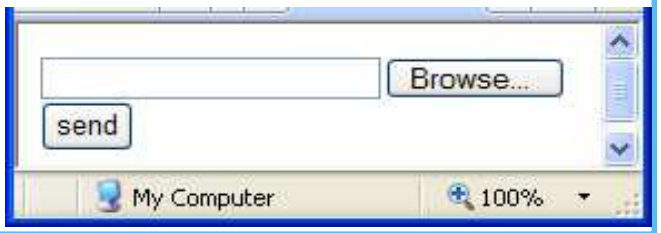
Lưu ý: **trường hợp hai radio không cùng tên**.

```
<html>
<body>
Radio Button Group : <br>
Nam: <input type="radio" name="sex" value="nam" checked><br>
Nu: <input type="radio" name="sex" value="nu" checked ><br>
</body>
</html>
```

6.2. Các thành phần của Form – File

- **Sử dụng để upload file lên server**

```
<form action="…" method="post" enctype="multipart/form-data"
name="...">
        <input TYPE="FILE" NAME="…">
</form>
```




```
<html>
<body>
<form name="frmMain" method="POST" enctype="multipart/form-data"
        action="xuly.php">
          <input type="file" name="fileUpload"><br>
          <input type="submit" value="send">
</form>
</body>
</html>
```

6.2. Các thành phần của Form – Submit

- Nút phát lệnh và gởi dữ liệu của form đến trang xử lý.
- Mỗi form chỉ có một nút submit và nút này được viền đậm

```
<input TYPE="submit" name="..." value="...">
```

```
<html>
<body>
<form name="frmMain" method="POST" enctype="multipart/form-data"
        action="xuly.php">
          <input type="file" name="fileUpload"><br>
          <input type="submit" value="send">
</form>
</body>
</html>
```

6.2. Các thành phần của Form – Reset

- Dùng để trả lại giá trị mặc định cho các control khác trong form

```
<input TYPE="reset" name="…" value="…">
```

```
<html>
<body>
<form name="frmMain" method="POST" enctype="multipart/form-data"
        action="xuly.php">
          <input type="file" name="fileUpload"><br>
          <input type="submit" value="send"><br>
          <input type="reset" value="reset">
</form>
</body>
</html>
```

6.2. Các thành phần của Form – Button

```
<input type="button" name="…" value="…" onclick="script">
```


```
<html>
<body>
   <input type="button"
      value="Hello!"
      onclick="alert('Hello from JavaScript');"
   >
</body>
</html>
```

```
<input type="button" name="btnNormal" value="Press Me!"
      onclick="alert('Hello from JavaScript');" >
```

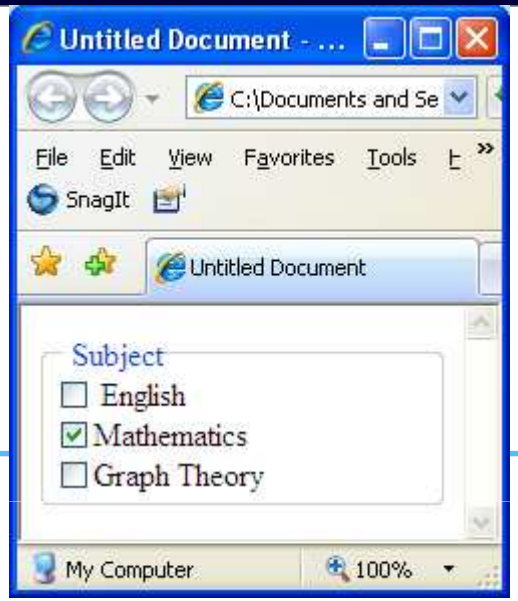
6.2. Các thành phần của Form – LABEL

```
<LABEL
        FOR = IDString
        CLASS=string
        STYLE=string
>
```


```
<label for="Languages">Anh văn: </label>
<input type="checkbox" name="Languages" id="Languages" value="Eng">
```

6.2. Các thành phần của Form – Fieldset

```
<fieldset>
        <legend>GroupBox </legend>
        <input ……>
        …
</fieldset>
```



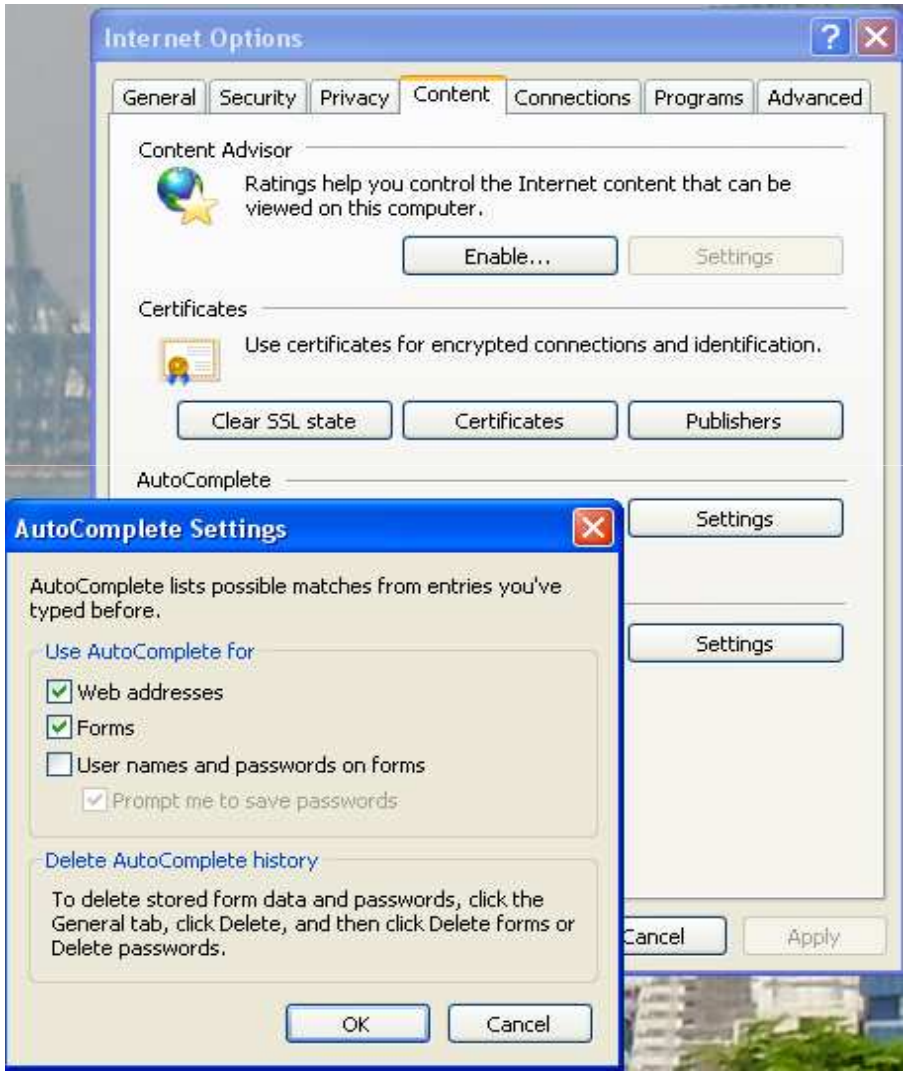

```
<html>
<body>
<fieldset>
    <legend> Subject </legend>
    <input type="checkbox" name="Subject[ ]" value="Eng"> English<br>
    <input type="checkbox" name="Subject[ ]" value="Math" checked>Mathematics<br>
    <input type="checkbox" name="Subject[ ]" value="GraphTheory">Graph Theory<br
</fieldset>
</body>
</html>
```

6.3. Các tiện ích của form và input

- Accesskey=*char*
  - Tạo phím nóng cho form fields.
  - Áp dụng cho tất cả form fields.
  - Cách nhấn Alt + *char*
  - Tránh các phím tắt của browser.
- Title = *string*
  - Tạo tooltip cho form fields.
  - Áp dụng cho tất cả form fields.
- Autocomplete = *ON/OFF*
  - Gợi ý tự động khi nhập liệu.
  - Áp dụng cho tất cả tag form, input.

6.3. Các tiện ích của form và input

- Để sử dụng tính năng AutoComplete thì phải mở nó lên trong trình duyệt web.
- Cách mở:
  - Tool → Internet Options
  - Chọn tab Content
  - Chọn settings trong ô AutoComplete



6.4. Gởi dữ liệu bằng phương thức POST/GET

- Các đối số của Form được ghi kèm theo vào đường dẫn URL của thuộc tính Action trong tag <Form>
- Khối lượng dữ liệu đối số được truyền đi của Form bị giới hạn bởi chiều dài tối đa của một URL trên Address bar.
- Chiều dài tối đa của một URL là 2048 bytes

6.4. Gởi dữ liệu bằng phương thức POST/GET

```
<html>
<body>
<form method="GET" action="xuly.php">
Anh văn: <input type="checkbox" name="L1" value="En"><br>
Hoa: <input type="checkbox" name="L2" value="Chz" checked><br>
Nhật: <input type="checkbox" name="L3" value="Jp"><br>
<input type="submit"  value="send">
</form>
</body>
</html>
```

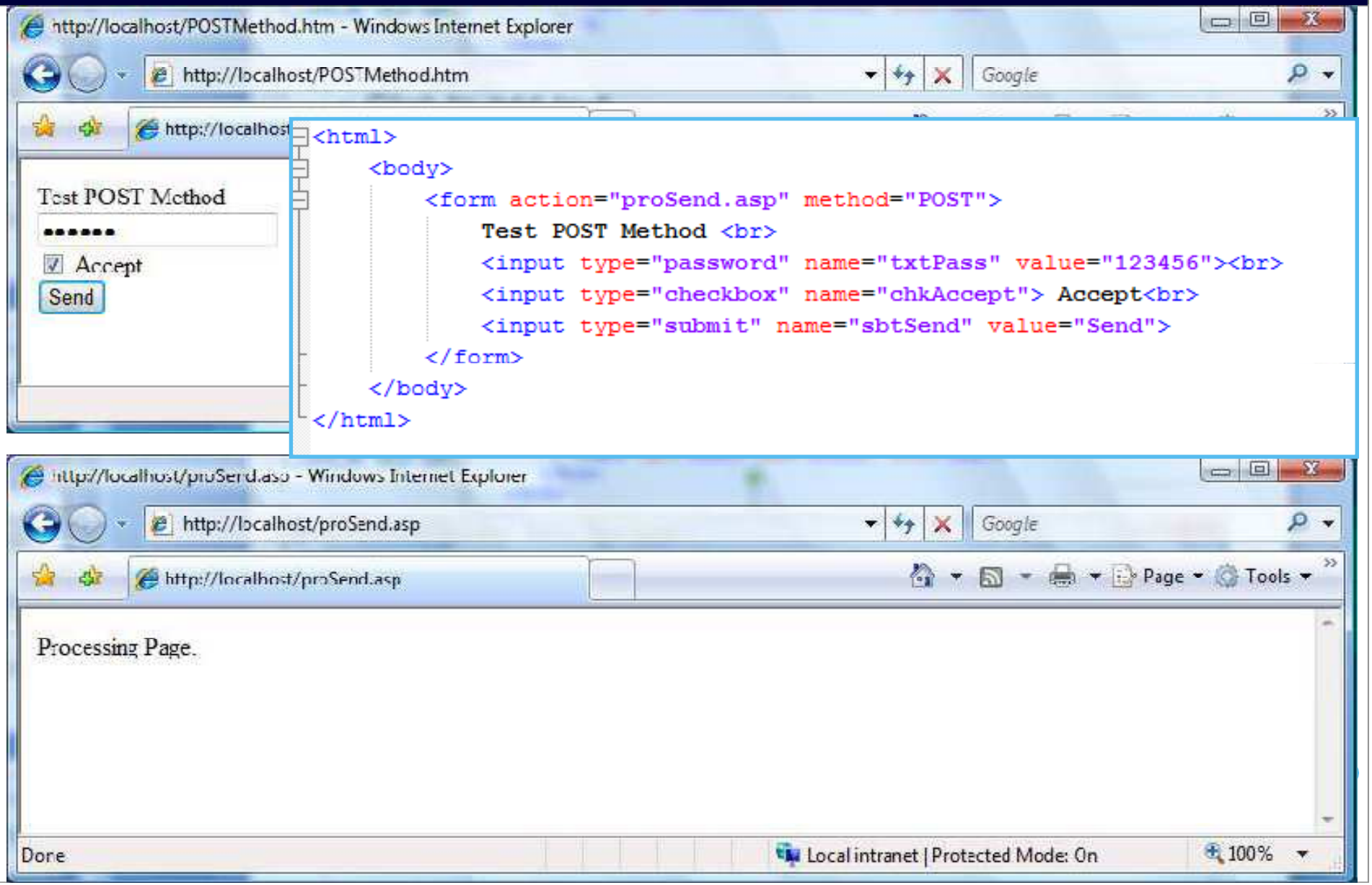
GET

6.4. Gởi dữ liệu bằng phương thức POST/GET

- Các đối số của Form được truyền "ngầm" bên dưới
- Khối lượng dữ liệu đối số được truyền đi của Form không phụ thuộc vào URL → Không bị giới hạn
- Chỉ sử dụng được phương thức truyền POST khi Action chỉ định đến trang web thuộc dạng xử lý trên Server

POST

6.4. Gởi dữ liệu bằng phương thức POST/GET



6.5. Thẻ <marquee>

- Dùng để tạo hiệu ứng chữ chạy trên màn hình trình duyệt

```
<MARQUEE
      BEHAVIOR = ALTERNATE | SCROLL | SLIDE
      DIRECTION = DOWN | LEFT | RIGHT | UP
      LOOP = string
      SCROLLAMOUNT=long
      SCROLLDELAY=long> Text Text Text
</MARQUEE>
```

7. Khung (Frame)

- Cho phép chia một trang web làm nhiều phần, mỗi phần chứa nội dung của 1 trang web khác
- Trình duyệt có thể không hỗ trợ khung

7. Khung (Frame)

- **Tạo trang web chứa các khung:**
  - **Thay thẻ <body>…</body> bằng:**

```
<frameset>
        các khung
</frameset>
<noframes>
  nội dung trong trường hợp trình duyệt
  không hỗ trợ khung
</noframes>
```

7. Khung (Frame)

- Một số thuộc tính của <frameset>
  - `rows = “n1, n2, ... nk”` hoặc `cols = “n1, n2, ... nk”`: Quy định có k dòng (hoặc cột), độ rộng dòng (cột) thứ i là n_i. n_i là số, có thể thay bằng *: phần còn lại
  - `frameborder = yes` hoặc `no`
  - `framespacing = “n”`: Khoảng cách giữa 2 khung

7. Khung (Frame)

- **Tạo 1 khung có nội dung là 1 trang web nào đó: `<frame>`**
  - **Thuộc tính:**
    - `src="Địa chỉ của trang chứa nội dung"`
    - `name="tên khung"`
    - `noresize`: **Không được thay đổi kích thước**
- **Thẻ <base> mặc định**
  - **Thuộc tính**
    - `target="Cửa sổ mặc định"`
    - `href="Địa chỉ gốc mặc định"`

Phần 3: CSS

Cascading Style Sheet

Nội dung

1. Giới thiệu
2. Định nghĩa style
3. Phân loại và sử dụng
4. Selector trong CSS

1. Giới thiệu

- CSS = Cascading Style Sheet
- Dùng để mô tả cách hiển thị các thành phần trên trang WEB
- Sử dụng tương tự như dạng TEMPLATE
- Có thể sử dụng lại cho các trang web khác
- Có thể thay đổi thuộc tính từng trang hoặc cả site nhanh chóng (cascading)

2. Định nghĩa Style

```
<tag style=
        "property1 :value1;
         property2 :value2;
         ……
         propertyN :valueN;"
>
…
</tag>
```

```
.SelectorName {
        property1 :value1;
        property2 :value2;
        ……
        propertyN :valueN;
}
<tag class="SelectorName">
…
</tag>
```

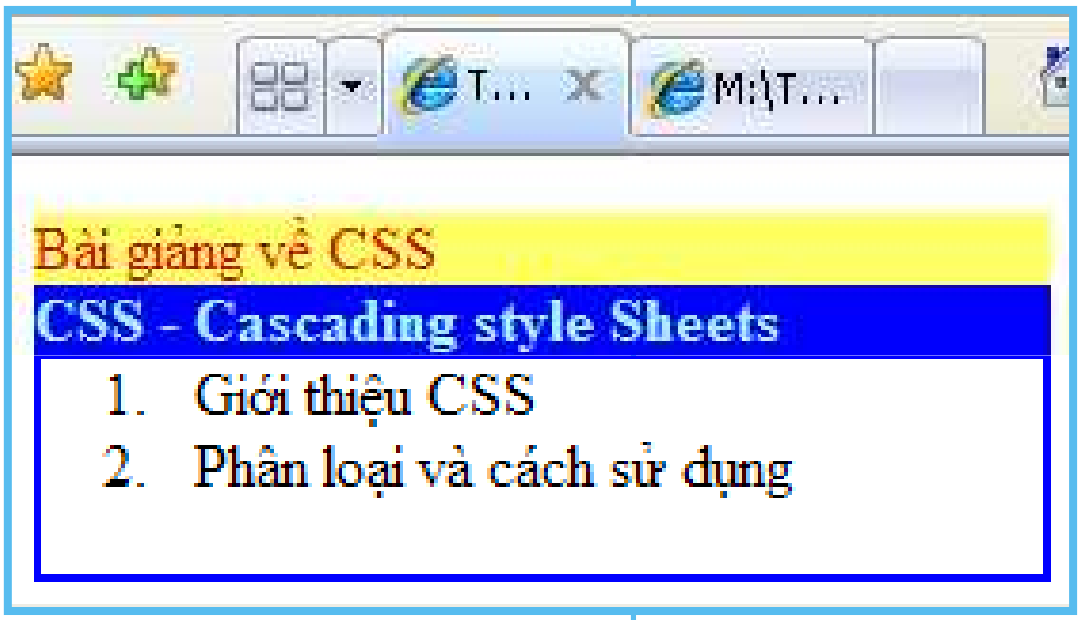
Cú pháp ghi chú: /* … */

```
/* Đây là ghi chú */
```

```
SelectorName{
property1: value1;/*Ghichu1*/
property2: value2;/*Ghichu2*/
}
```

2. Định nghĩa Style

```
1  <html>
2  <head>
3      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4      <title>Thiết kế web với HTML</title>
5      <style>
6          .TieuDe{
7              width:200pt;
8              background:#0000FF;
9              color:#99FFFF;
10             font-weight:bold;
11             size:20pt;
12         }
13         .NoiDung{
14             width:200pt;
15             border:thin;
16             border-style:solid;
17             border-color:#0000FF;
18         }
19     </style>
20 </head>
21 <body>
22     <div style="background-color:#FFFF66;color:#EE0000;width:200pt">
23         Bài giảng về CSS
24     </div>
25     <div class="TieuDe">CSS - Cascading style Sheets</div>
26     <div class="NoiDung">
27         <ol>
28             <li>Giới thiệu CSS</li>
29             <li>Phân loại và cách sử dụng</li>
30         </ol>
31     </div>
32 </body>
33 </html>
```



Ví dụ

3. Phân loại và sử dụng

1. Inline Style Sheet

   Style định nghĩa trong tag

2. Embedding Style Sheet

   Style định nghĩa trong tag <style>

3. External Style Sheet

   Style định nghĩa trong file *.css

3. Phân loại và sử dụng

```html
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Thiết kế web với HTML</title>
    <style type="text/css">
    <!-- @import url("vidu_css_02.css"); -->
    </style>
</head>
<body>
    <div style="background-color:#FFFF66;color:#EE0000;width:200pt">
        Bài giảng về CSS
    </div>
    <div class="TieuDe">CSS - Cascading style Sheets</div>
    <div class="NoiDung">
        <ol>
            <li>Giới thiệu CSS</li>
            <li>Phân loại và cách sử dụng</li>
        </ol>
    </div>
</body>
</html>
```

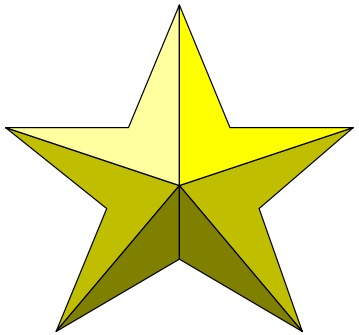
External Style Sheet

```css
@charset "utf-8";
/* CSS Document */
.TieuDe{
    width:200pt;
    background:#0000FF;
    color:#99FFFF;
    font-weight:bold;
    size:20pt;
}
.NoiDung{
    width:200pt;
    border:thin;
    border-style:solid;
    border-color:#0000FF;
}
```

3. Phân loại và sử dụng

```
<head>
    <link rel="stylesheet" href="URL" type="text/css">
</head>
```

External Style Sheet



```
<head>
    <style type="text/css"  media="all | print | screen" >
        @import url(URL);
    </style>
</head>
```

3. Phân loại và cách sử dụng

	Inline Style Sheet	Embedding Style Sheet	External Style Sheet
Khai báo	**Kiểu 1**	**Kiểu 2**	**Kiểu 3**
Cú pháp	 Test 	<style type="text/css"> .TieuDe1{color: red;} </style> Test 	<link rel="stylesheet " href="main.css" /> Test
Ưu điểm	• Dễ dàng quản lý Style theo từng tag của tài liệu web. • Có độ ưu tiên cao nhất	• Dễ dàng quản lý Style theo từng tài liệu web. • Không cần tải thêm các trang thông tin khác cho style	• Có thể thiết lập Style cho nhiều tài liệu web. • Thông tin các Style được trình duyệt cache lại
Khuyết điểm	• Cần phải Khai báo lại thông tin style trong từng tài liệu Web và các tài liệu khác một cách thủ công. • Khó cập nhật style	• Cần phải khai báo lại thông tin style cho các tài liệu khác trong mỗi lần sử dụng	• Tốn thời gian download file *.css và làm chậm quá trình biên dịch web ở trình duyệt trong lần đầu sử dụng

3. Phân loại và cách sử dụng

1. Inline Style Sheet
2. Embedding Style Sheet
3. External Style Sheet
4. Browser Default

Độ Ưu Tiên Giảm Dần

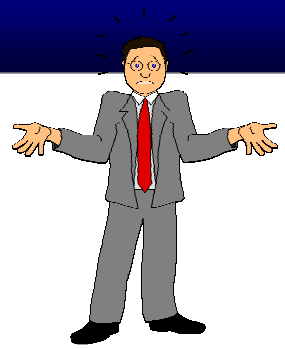
4. Selector trong CSS

Các loại Selector

Loại	Mô tả phạm vi ảnh hưởng	Ví dụ
element	**Định dạng áp dụng cho ND tất cả các tag** Element **trong tài liệu Web**	h1 {color: red;} /* ND của thẻ <h1> bị định dạng màu chữ=đỏ */
#id	**Định dạng áp dụng cho ND tất cả các tab có thuộc tính** id **trong tà liệu Web**	#test {color: green;} /* ND của bất kỳ tag có thuộc tính id=test đều bị định dạng màu chữ=xanh lá */
.class	**Định dạng áp dụng cho ND tất cả các tab có thuộc tính** class **trong tà liệu Web**	.note {color: yellow;} /* ND của bất kỳ tag có thuộc tính class=note đều bị định dạng màu chữ=vàng*/
element . class	**Định dạng áp dụng cho ND các tag** Element **có thuộc tính** class **tương ứng**	h1.note {text-decoration: underline;} /* ND của các thẻ <h1> có thuộc tính class=note đều bị định dạng gạch chân */

4. Selector trong CSS

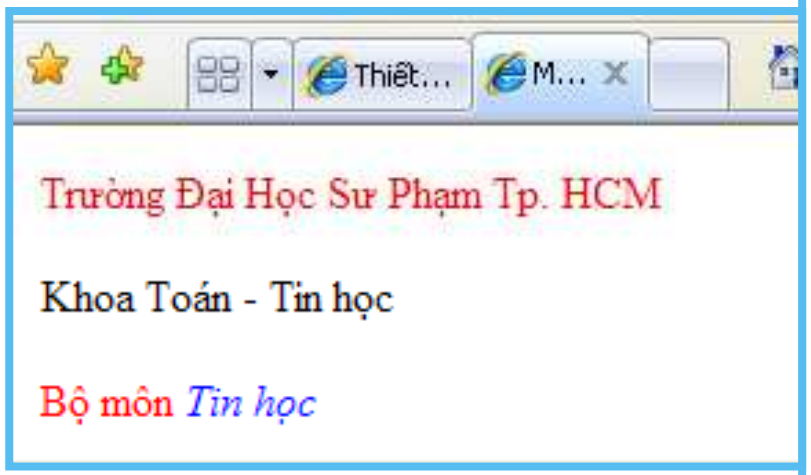
Các loại Selector



Loại	Mô tả phạm vi ảnh hưởng	Ví dụ
Grouping	Định dạng áp dụng cho ND một nhóm các tag trong tài liệu.	h1,h2,h3 {background-color: orange;} /* ND của các thẻ <h1> <h2> <h3> đều bị định dạng màu nền = màu cam */
Contextual	Định dạng áp dụng cho ND các thẻ được lồng trong một thẻ cha nào đó	p strong {color: purple;} /* ND của các thẻ nằm trong thẻ <p> đều bị định dạng màu chữ=màu tía */
Pseudo Class Pseudo element	Định dạng được áp dụng dựa vào trạng thái của các Element. (Không xuất hiện trong mã lệnh HTML)	a:active { color: green; }

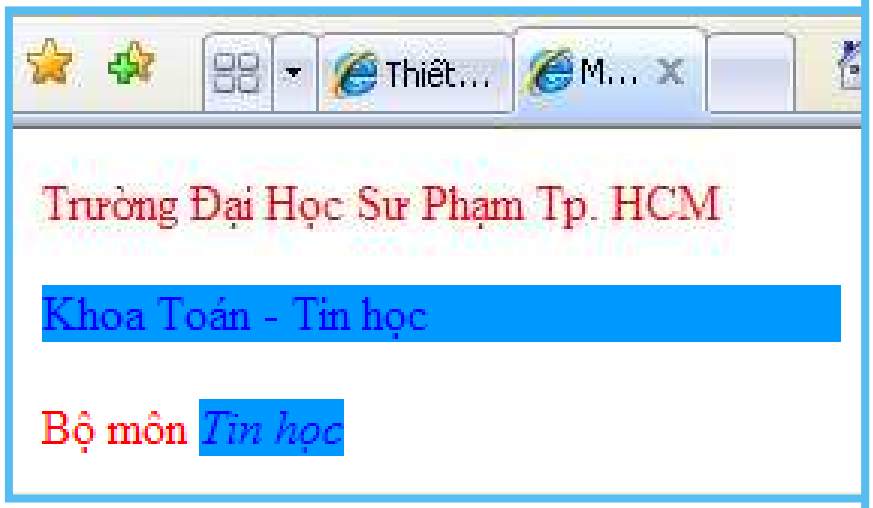
4.1. CSS - ELEMENT

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <style>
5      p{
6          color:red;
7      }
8      em{
9          color:blue;
10     }
11 </style>
12 </head>
13 <body>
14     <p>Trường Đại Học Sư Phạm Tp. HCM</p>
15     <div>Khoa Toán - Tin học</div>
16     <p>Bộ môn <em>Tin học</em></p>
17 </body>
18 </html>
```



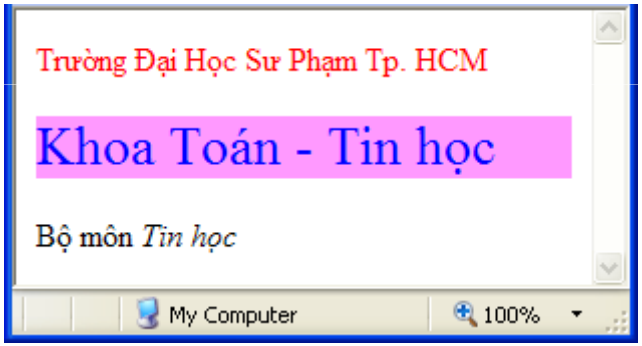
4.1. CSS - ID

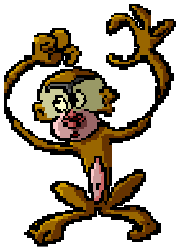
```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <style>
5      #id01{
6          color:red;
7      }
8      #id02{
9          color:blue;
10         background-color:#0099FF;
11     }
12 </style>
13 </head>
14 <body>
15     <p id="id01">Trường Đại Học Sư Phạm Tp. HCM</p>
16     <div id="id02">Khoa Toán - Tin học</div>
17     <p id="id01">Bộ môn <em id="id02">Tin học</em></p>
18 </body>
19 </html>
```



4.1. CSS - CLASS

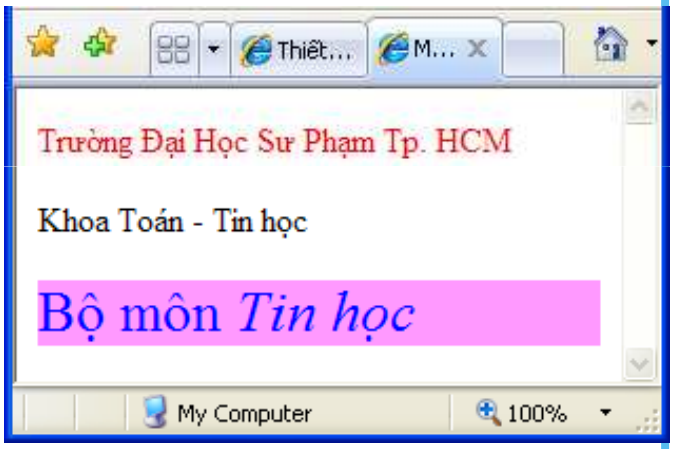
```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <style>
5      .truong{
6          color:red;
7      }
8      .khoa{
9          color:blue;
10         background-color:#FF99FF;
11         font-size:20pt;
12     }
13 </style>
14 </head>
15 <body>
16     <p class="truong">Trường Đại Học Sư Phạm Tp. HCM</p>
17     <div class="khoa">Khoa Toán - Tin học</div>
18     <p>Bộ môn <em>Tin học</em></p>
19 </body>
20 </html>
```

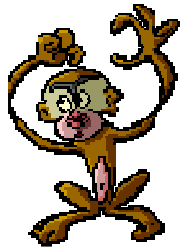




4.1. CSS – ELEMENT_CLASS

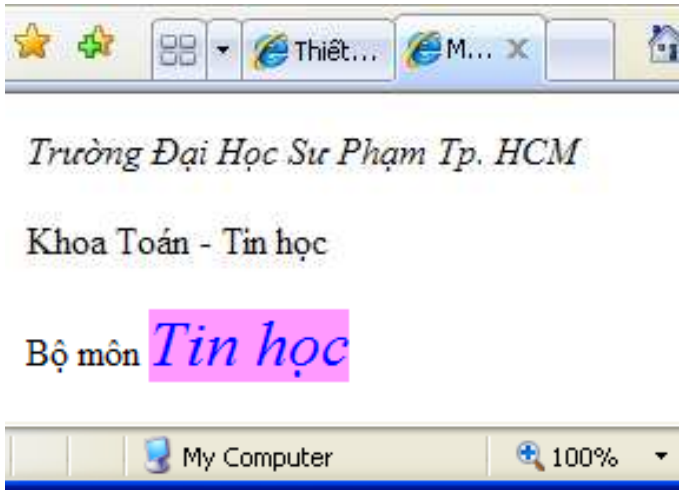
```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <style>
5      .truong{
6          color:red;
7      }
8      p.khoa{
9          color:blue;
10         background-color:#FF99FF;
11         font-size:20pt;
12     }
13 </style>
14 </head>
15 <body>
16     <p class="truong">Trường Đại Học Sư Phạm Tp. HCM</p>
17     <div class="khoa">Khoa Toán - Tin học</div>
18     <p class="khoa">Bộ môn <em>Tin học</em></p>
19 </body>
20 </html>
```

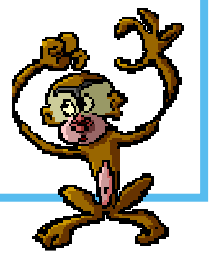




4.1. CSS – CONTEXTUAL

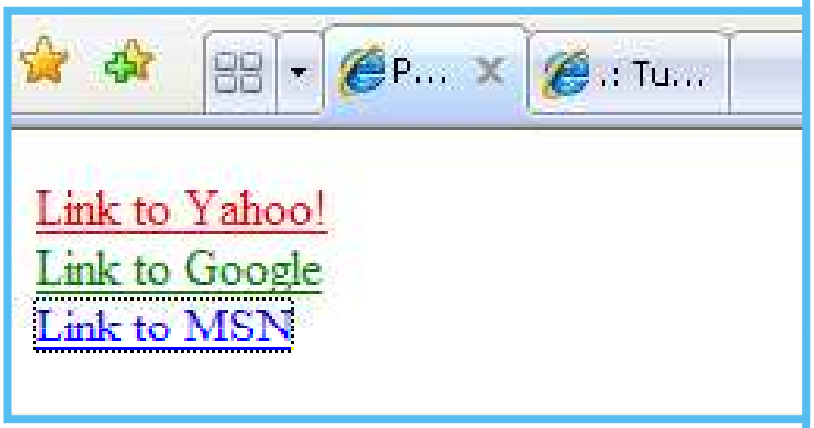
```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <style>
5      p em{
6          color:blue;
7          background-color:#FF99FF;
8          font-size:20pt;
9      }
10 </style>
11 </head>
12 <body>
13     <em><p class="truong">Trường Đại Học Sư Phạm Tp. HCM</p></em>
14     <div class="khoa">Khoa Toán - Tin học</div>
15     <p class="khoa">Bộ môn <em>Tin học</em></p>
16 </body>
17 </html>
```



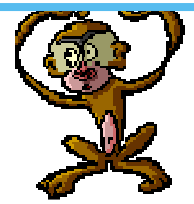


4.1. CSS – Pseudo Class

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>Pseudo Class</title>
5  <style type="text/css">
6      a:link{color:red}
7      a:visited{color:green}
8      a:active{color:blue}
9  </style>
10 </head>
11 <body>
12     <a href="http://yahoo.com">Link to Yahoo!</a><br>
13     <a href="http://google.com">Link to Google</a><br>
14     <a href="http://msn.com">Link to MSN</a>
15 </body>
16 </html>
```

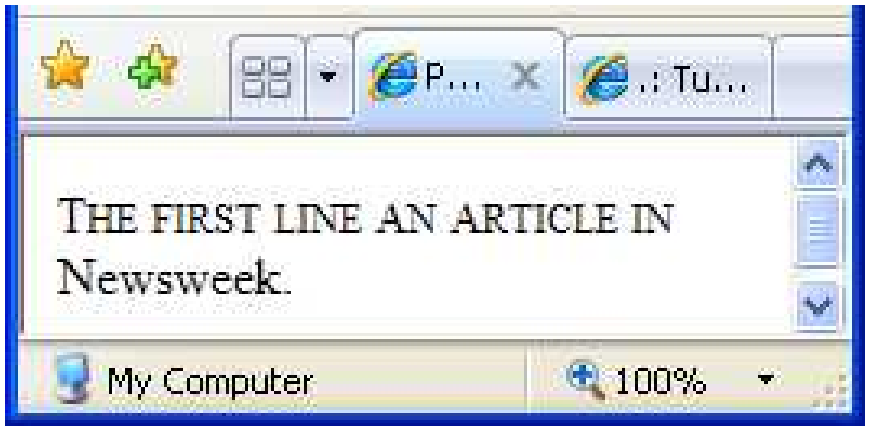


Dùng phím tab để active link



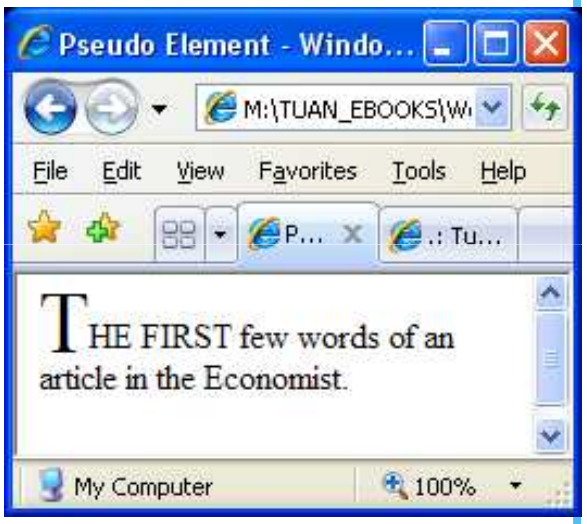
4.1. CSS – Pseudo Element

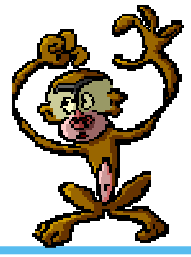
```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>Pseudo Element</title>
5  <style type="text/css">
6      div:first-line{font-variant:small-caps}
7  </style>
8  </head>
9  <body>
10     <div>The first line an article in Newsweek.</div>
11 </body>
12 </html>
```



4.1. CSS – Pseudo Element

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>Pseudo Element</title>
5  <style type="text/css">
6      P{font-size:20px;line-height:12pt}
7      P:first-letter{font-size:200%;float:left}
8      span{text-transform:uppercase}
9  </style>
10 </head>
11 <body>
12     <p>
13         <span>
14             <p>T</p>he first
15         </span>
16         few words of an article in the Economist.
17     </p>
18 </body>
19 </html>
```





Phần 4: JavaScript

Nội dung

1. Tổng quan
2. Một ví dụ
3. Ngôn ngữ JavaScript
4. Một số hàm khác

1. Tổng quan

Giới thiệu DHTML

- DHTML= Dynamic HyperTextMarkup Language
- DHTML = HTML + CSS + ClientScript + HTML DOM
- Tích hợp các tính năng của các trình duyệt thế hệ thứ 4 (IEv5, Netscape4, Firefox2.0+, Opera 7.0, …)

1. Tổng quan

Ngôn ngữ Script

- Là ngôn ngữ dạng thông dịch
- Giúp trang web có tính tương tác tốt
- Các ngôn ngữ script thông dụng
  - Javascript (Netscape)
  - Jscript (Microsoft)
  - VBScript (Microsoft)

1. Tổng quan

- Ứng dụng **Client-Side**:
  - Thực hiện tại Browser (Nescape Navigator, IE, Firefox, Safari, ...)
  - Script tại Client-Side: thực hiện các tương tác với người dùng, thay đổi cấu trúc trang web, kiểm tra dữ liệu được nhập vào của người dùng, …
- Ứng dụng **Server-Side**:
  - Thực hiện tại WebServer (IIS, Apache, Netscape Enterprise Server, ….)
  - Script tại Server-Side: kết nối CSDL, chia sẽ thông tin giữa các người duyệt web, truy cập hệ thống file trên server, …)

1. Tổng quan

Quá trình thực hiện ứng dụng Server-Side

- Tạo trang Web có chứa cả Srcipt Client-Side và Script Server-Side
- Khi Client browser yêu cầu thực hiện, server (run-time engine) sẽ thực hiện các lệnh Server-side Scipts và trả **trang Web HTML** về browser

1. Tổng quan

Vị trí của các đoạn Script

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

2. Một ví dụ

```
<html>
<head>
    <title>
        Sử dụng JavaScript
    </title>
</head>
<body>
    document.write(“Hello world!”);
    <hr>
    <script language=“javascript”>
        document.write(“Hello world!”);
    </script>
</body>
</html>
```

3. NGÔN NGỮ JAVASCRIPT

3.1. Giới thiệu

- **JavaScript** và **Java** là hai ngôn ngữ hoàn toàn khác nhau
  - **Java** là một ngôn ngữ lập trình "đầy đủ", trong đó các ứng dụng cần được biên dịch trước khi thực thi. Java là ngôn ngữ mạnh mẽ và phứ tạp hơn rất nhiều. Java được sáng tạo bởi công ty Sun Micro System
  - **JavaScript** không cần phải được biên dịch trước, cấu trúc lệnh đơn giản và là một ngôn ngữ kịch bản. JavaScript là sản phẩm của Netscape Communications Corporation

3.1. Giới thiệu

- **JavaScript** là một ngôn ngữ lập trình hướng đối tượng dạng kịch bản:
  - Không cần được biên dịch trước khi chạy, toàn bộ quá trình thông dịch sẽ diễn ra ngay trong quá trình đoạn kịch bản (script) được gọi
  - Thuận lợi
    - dễ dàng triển khai một cách nhanh chóng
    - hoạt động ở máy trạm, giảm tải cho máy chủ
  - Hạn chế
    - khó kiểm tra & xử lý lỗi
    - phụ thuộc vào trình duyệt web ở phía client
    - tốc độ không cao

3.1. Giới thiệu

- **JavaScript thường dùng**
  - tạo hiệu ứng cho các ảnh trong trang web
  - trò chơi (game)
  - trả lời các sự kiện: nhấn chuột, di chuyển chuột,…
  - đọc và ghi các thẻ HTML
  - kiểm tra tính xác thực của dữ liệu
  - phát hiện trình duyệt được sử dụng để duyệt web
  - tạo cookie
  - ...

3.2. Cú pháp và quy ước

- Javascript phân biệt chữ hoa – chữ thường
- Các câu lệnh javascript cách nhau bởi dấu “;”
- Không phân biệt khoảng trắng, Tab, xuống dòng trong câu lệnh.
- Chuỗi và dấu nháy
  - Chuỗi trong javascript được đặt trong cặp nháy đơn (‘’) hoặc nháy kép (“”)
  - Ví dụ:

```
<input value = ‘He said “Javascriptis good” ’>
<input type=“button” value=“Click Me!”
  onclick=“alert(‘Hello’);”>
```

3.2. Cú pháp và quy ước

- **Ghi chú: theo cú pháp của C++**
  - **Ghi chú dòng: //**
  - **Ghi chú đoạn: /* … */**

{ }	Đánh dấu khối lệnh
[ ]	Sử dụng trong cấu trúc Mảng
( )	Sử dụng trong hàm, thuộc tính đối tượng

\b : Backspace	\’ : Dấu nháy đơn	\t : tab
\f : Form feed	\” : Dấu nháy kép	
\n : New line	\r : carriage return	

3.2. Cú pháp và quy ước

TỪ KHÓA

break	do	if	switch	typeof
case	else	in	this	var
catch	false	instanceof	throw	void
continue	finally	new	true	while

3.2. Cú pháp và quy ước

TỪ KHÓA

abstract	double	goto	native	static
boolean	enum	implements	package	super
byte	export	import	private	synchronized
char	extends	int	protected	throws
class	final	interface	public	transient
const	float	long	short	volatile
debugger				

3.2. Cú pháp và quy ước

TỪ KHÓA

arguments	encodeURI	Infinity	Object	String
Array	Error	isFinite	parseFloat	SyntaxError
Boolean	escape	isNaN	parseInt	TypeError
Date	eval	Math	RangeError	undefined
decodeURI	EvalError	NaN	ReferenceError	unescape
decodeURIComponent	Function	Number	RegExp	URIError

3.2. Cú pháp và quy ước

Đặt tên

- Bắt đầu bởi chữ cái hay dấu gạch dưới (_) hay dấu dollar ($)
  - Dấu dollar là không hợp lệ trong các phiên bản trước JavaScript 1.1, được tích hợp vào để hỗ trợ các phần mềm sinh mã tự động.
  - Tránh sử dụng dấu này
- Tiếp theo bởi chữ cái, số hay dấu gạch dưới, dấu dollar
- Không đặt tên trùng với từ khóa

3.3. Kiểu dữ liệu

Kiểu dữ liệu	Ví dụ	Mô tả
Object	var listBooks = new Array(10) ;	Trước khi sử dụng, phải cấp phát bằng từ khóa new
String	“The cow jumped over the moon.” “40”	Chứa được chuỗi unicode Chuỗi rỗng “”
Number	0.066218 12	Theo chuẩn IEEE 754
boolean	true / false	
undefined	var myVariable ;	myVariable = undefined
null	connection.Close();	connection = null
function	var add = new function("x", "y", "return(x+y)"); add(2, 3);	*functionName* **= new function(** **[***argname1*, **[...** *argnameN*,**]]** *body* **);**

Biến trong JavaScript lưu bất kỳ giá trị của kiểu dữ liệu nào

3.4. Khai báo biến, phạm vi biến

- Sử dụng từ khóa **var** để khai báo biến
- Ví dụ: **var** i;
- Gán giá trị cho biến:
  - i = 10;
  - i = “Mười”;
- Một biến chưa được gán giá trị thì sẽ có giá trị là “undefined”

3.4. Khai báo biến, phạm vi biến

- Phạm vi của biến gắn liền với vùng chương trình nó được khai báo
- Biến toàn cục có phạm vi hoạt động trên toàn bộ tài liệu (khai báo ngoài hàm)
- Biến khai báo trong hàm chỉ có tác dụng bên trong hàm
- Một biến toàn cục được tồn tại từ khi nó được khai báo cho đến trang web đã đóng
- JavaScript không có khái niệm phạm vi theo khối

3.5. Toán tử

Loại	Toán tử
Toán học	+ - * / % ++ --
So sánh	< > <= >= != ==
Luận lý	&& || ?: ,
Xử lý bit	~ << >> >>> & (and) | (or) ^ (xor)
Gán	= += -= *= /= %= >>= <<= &= |= ^=

3.6. Một số đối tượng dữ liệu

- String Object
- Number Object
- Date Object
- Math Object
- Array Object
- ActiveX Object

3.6. Một số đối tượng dữ liệu - String

Thuộc tính

- length: chiều dài của chuỗi
- constructor: Dùng để kiểm tra kiểu của biến
- prototype: Bổ sung prototype hàm cho một đối tượng
- Nối kết các chuỗi bằng toán tử +

3.6. Một số đối tượng dữ liệu - String

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>JavaScript - String()</title>
5  </head>
6  <body>
7  <script language="javascript">
8      var strTen=new String();
9      strTen=2008;
10     if (strTen.constructor==String)
11         document.write("Đây là chuỗi có giá trị: "+strTen);
12     else
13         document.write("Đây là số có giá trị: "+strTen);
14 </script>
15 </body>
16 </html>
```

Thuộc tính

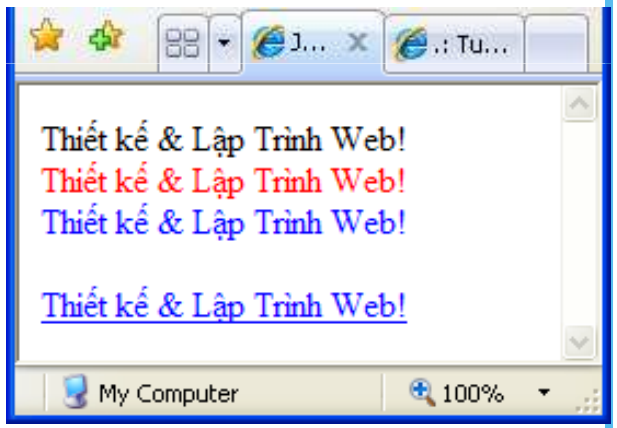
3.6. Một số đối tượng dữ liệu - String

- anchor
- big
- blink
- bold
- charAt
- charCodeAt
- concat
- fixed
- fontcolor
- fontsize
- fromCharCode
- indexOf
- italics
- lastIndexOf
- link
- localeCompare
- match
- replace
- search
- slice
- small
- split
- strike
- sub
- substr
- substring
- sup
- toLocaleLowerCase
- toLocaleUpperCase
- toLowerCase
- toUpperCase
- toString
- valueOf

Phương thức

3.6. Một số đối tượng dữ liệu - String

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript - String()</title>
</head>
<script language="javascript">
    function ChangeColor(color)
    {
        return this.fontcolor(color);
    }
    String.prototype.ChangeColor=ChangeColor;
</script>
<body>
<script language="javascript">
    var x = new String();
    x="Thiết kế & Lập Trình Web!";
    document.write(x+"<br>");
    document.write(x.ChangeColor('red')+"<br>");
    document.write(x.ChangeColor('blue')+"<br><br>");
    document.write(x.link("http://google.com.vn"));
</script>
</body>
</html>
```



Phương thức

3.6. Một số đối tượng dữ liệu - String

Phương thức	Ví dụ

strObj.charAt(index)	var str = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
	var s1 = str.charAt(1); // s1 = "B"
	var s2 = str.charAt(26); // s1 = null

strObj.charCodeAt(index)	var n = str.charCodeAt(1); //Get the Unicode char at pos 1.
strObj.concat([string2[, string3[,... [, stringn]]]])	var str1 = "ABCDEFGHIJKLM" var str2 = "NOPQRSTUVWXYZ"; var s = str1.concat(str2); // s = "ABC...NOP...XYZ"
strObj.fromCharCode([code1[, code2[,.... [, codeN]]]])	Tạo chuỗi từ mã ngoài Unicode. var test = String.fromCharCode(112, 108, 97, 105, 110);
strObj.indexOf(subString[, startIndex])	var str1 = "BABEBIBOBUBABEBIBOBU" var s = str1.indexOf("BEB", 0); // s = 2
strObj.lastIndexOf(substring[, startindex])	var str1 = "BABEBIBOBUBABEBIBOBU" var s = str1.lastIndexOf ("BEB", 0); // s = 12

Phương thức

3.6. Một số đối tượng dữ liệu - String

Phương thức	Ví dụ

strObj.match(rgExp)
rgExp = /pattern/[flags]
flags, may be combined:
g (global search for all occurrences of pattern)
i (ignore case)
m (multiline search)
	var s = "The rain in Spain falls mainly in the plain";
	var re = /The/gi; //Create regular expression pattern.
	var r = s.match(re); // r = {"The", "the"}.

strObj.replace(rgExp, replaceText)	Tìm và thay thế.
stringObj.search(rgExp)	Có xuất hiện, cho ra vị trí. Ngược lại -1.
stringObj.slice(start, [end])	Lấy chuỗi con.
stringObj.split([separator[, limit]]) separator: string or Regular Expression	var s = "The rain in Spain falls mainly in the plain."; var ss = s.split(" ", 3); // ss = {"The"; "rain"; "in"}

string.anchor(anchorString)	var string = "Information about Fish";
	var str_anchor = string.anchor("fish_infor");
	window.document writeln(str_anchor);
	// < A name="fish_infor">Information about fish

Phương thức

3.6. Một số đối tượng dữ liệu - Number

- Kiểu dữ liệu số nguyên, số thực
- Bắt đầu bằng ký số "0": Số nguyên hệ bát phân
- Bắt đầu bằng "0x": Số nguyên hệ thập lục phân
- VD: Cho biết giá trị thập phân tương ứng của các number sau:
  - 125 = ?
  - 010 = ? 014 = ? 028 = ?
  - 0xFF = ? 0x3.12 = ?

3.6. Một số đối tượng dữ liệu - Number

Thuộc tính

- constructor
- prototype
- MAX_VALUE (khoảng = 1.79E+308)
- MIN_VALUE (khoảng = 5.00E-324)
- NaN
- NEGATIVE_INFINITY (= -MAX_VALUE)
- POSITIVE_INFINITY (= -MIN_VALUE)

3.6. Một số đối tượng dữ liệu - Number

Phương thức

- toExponential

  numObj.toExponential([fractionDigits])

- toFixed

  numObj.toFixed([fractionDigits])

- toPrecision

  numObj.toPrecision([precision])

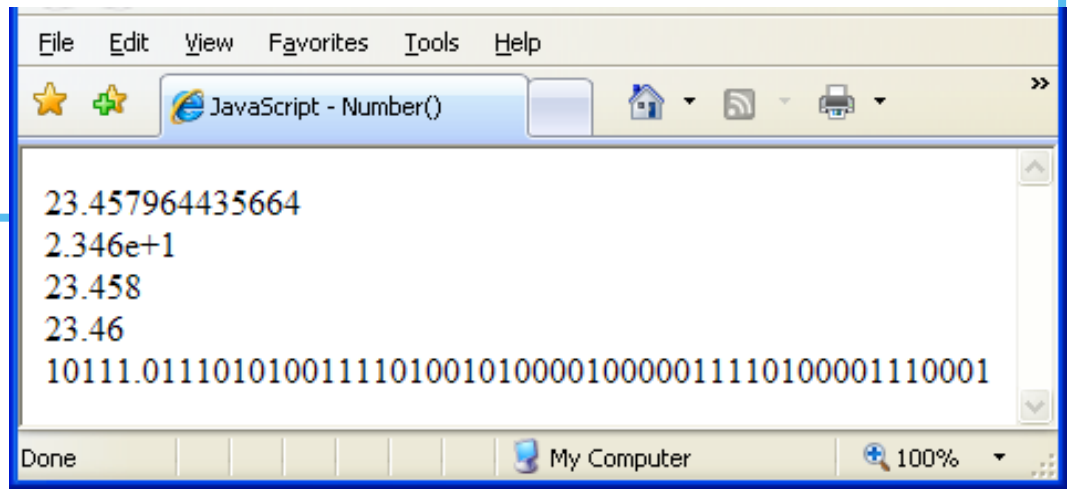
- toString

  objectname.toString([radix])

  radix : [2; 8; 10; 16] (các hệ cơ số)

3.6. Một số đối tượng dữ liệu - Number

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>JavaScript - Number()</title>
5  </head>
6  <script language="javascript">
7      var num = new Number();
8      num=23.457964435664;
9      document.write(num+"<br>");
10     document.write(num.toExponential(3)+"<br>");
11     document.write(num.toFixed(3)+"<br>");
12     document.write(num.toPrecision(4)+"<br>");
13     document.write(num.toString(2));
14 </script>
15 <body>
16 </body>
17 </html>
```



Phương thức

3.6. Một số đối tượng dữ liệu - Number

Phương thức

- **Chuyển chuỗi thành số**

```
parseInt();                    parseFloat();
parseInt("42")                 // result = 42
parseInt("42.33")              // result = 42
parseFloat("42.33")            // result = 42.33
3 + 3 + parseInt("3")          // result = 9
```

- **Chuyển số thành chuỗi** (auto)

```
3 + "3"                        // result = "33"
3 + 3 + "3"                    // result = "63"
(""+ 2500)                     // result = "2500"
(""+ 2500).length              // result = 4
```

- isNaN(number) → true nếu number khác NaN
- isFinite(number) → true nếu number khác NaN, NEGATIVE_INFINITY, POSITIVE…

3.6. Một số đối tượng dữ liệu - Math

- abs
- acos
- asin
- atan
- atan2
- ceil
- cos
- exp
- floor
- log
- max
- min
- pow
- random
- round
- sin
- sqrt
- tan

- E
- LN2
- LN10
- LOG2E
- LOG10E
- PI
- SQRT1_2
- SQRT2

3.6. Một số đối tượng dữ liệu - Date

Argument	Description	Example
None	Creates object with the current date and time.	var rightNow = new Date();
"month,dd, yyyy hh:mm:ss"	Creates object with the date represented by the specified month, day (*dd*), year (*yyyy*), hour (*hh*), minute (*mm*), and second (*ss*). Any omitted values are set to zero.	var birthDay = new Date("March 24, 1970");
Milliseconds	Creates object with date represented as the integer number of milliseconds after the epoch.	var someDate = new Date(795600003020);
yyyy, mm, dd	Creates object with the date specified by the integer values year (yyyy), month (mm), and day (dd).	var birthDay = new Date(1970, 2, 24);
yyyy, mm, dd, hh, mm, ss	Creates object with the date specified by the integer values for the year, month, day, hours, minutes, and seconds.	var birthDay = new Date(1970, 2, 24, 15, 0, 0);
yyyy, mm, dd, hh, mm, ss, ms	Creates object with the date specified by the integer values for the year, month, day, hours, seconds, and milliseconds.	var birthDay = new Date(1970, 2, 24, 15, 0, 250);

3.6. Một số đối tượng dữ liệu - Date

- getDate
- getDay
- getFullYear
- getHours
- getMilliseconds
- getMinutes
- getMonth
- getSeconds
- getTime
- getTimezoneOffset
- getVarDate
- getYear
- setDate
- setFullYear
- setHours
- setMilliseconds
- setMinutes
- setMonth
- setSeconds
- setTime
- setYear
- parse
- toDateString
- toGMTString
- toLocaleDateString
- toLocaleString
- toLocaleTimeString
- toString
- toTimeString
- toUTCString

3.6. Một số đối tượng dữ liệu - Date

```
var today = new Date();
document.write("The current date : "+today+"<<br />>");
document.write("Date.getDate() : "+today.getDate()+"<<br />>");
document.write("Date.getDay() : "+today.getDay()+"<<br />>");
document.write("Date.getFullYear() : "+today.getFullYear()+"<<br />>");
document.write("Date.getHours() : "+today.getHours()+"<<br />>");
document.write("Date.getMilliseconds() : "+today.getMilliseconds()+"<<br />>");
document.write("Date.getMinutes() : "+today.getMinutes()+"<<br />>");
document.write("Date.getMonth() : "+today.getMonth()+"<<br />>");
document.write("Date.getSeconds() : "+today.getSeconds()+"<<br />>");
document.write("Date.getTime() : "+today.getTime()+"<<br />>");
document.write("Date.getTimezoneOffset() : "+today.getTimezoneOffset()+"<<br />>");
document.write("Date.getYear() : "+today.getYear()+"<<br />>");
```

The current date : Mon Apr 30 23:00:15 PDT 2001
Date.getDate() : 30
Date.getDay() : 1
Date.getFullYear() : 2001
Date.getHours() : 23
Date.getMilliseconds() : 340
Date.getMinutes() : 0
Date.getMonth() : 3
Date.getSeconds() : 15
Date.getTime() : 988696815340
Date.getTimezoneOffset() : 420
Date.getYear() : 2001

3.6. Một số đối tượng dữ liệu - Array

- var arrayObj = new Array();
- var arrayObj = new Array([size]);
- var arrayObj = new Array([element0[, element1[, ...[, elementN]]]]);
- var arrayObj = new Array(Array);
- Chỉ số của mảng bắt đầu từ 0

3.6. Một số đối tượng dữ liệu - Array

Thuộc tính và Phương thức

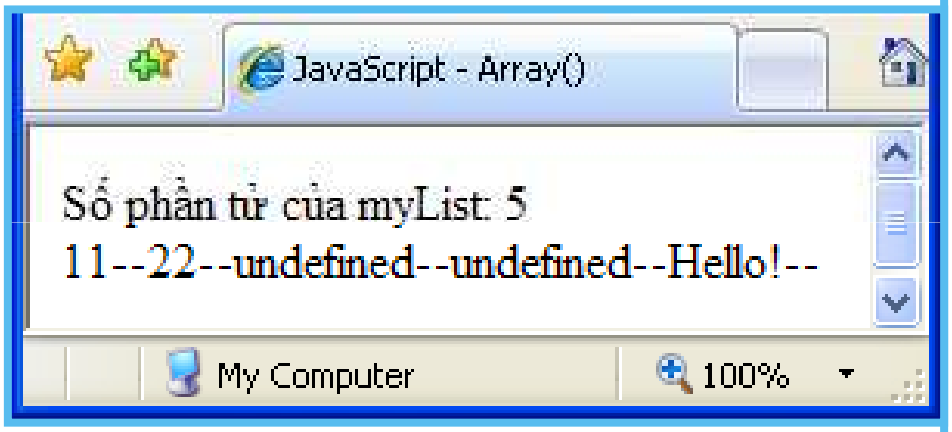
- length
- constructor
- prototype

- concat
- join
- pop
- push
- reverse
- shift

- slice
- sort
- splice
- toString
- unshift

3.6. Một số đối tượng dữ liệu - Array

```
1  <html>
2  <head>
3  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4  <title>JavaScript - Array()</title>
5  </head>
6  <script language="javascript">
7      var myList=new Array();
8      myList[0] = 11;
9      myList[1] = 22;
10     myList[4] = "Hello!";
11     delete myList[2];
12     delete myList[3];
13     document.write("Số phần tử của myList: " + myList.length + "<br>");
14     for (i=0; i<myList.length; i++)
15     {
16         document.write(myList[i] + "--");
17     }
18 </script>
19 <body>
20 </body>
21 </html>
```



3.6. Một số đối tượng dữ liệu - Array

Phương thức	Ví dụ

array1.concat([item1[, item2[,...,...,... [, itemN]]]])	var a, b, c, d;
	a = new Array(1,2,3);
	b = "JScript";
	c = new Array(42, "VBScript");
	d = a.concat(b, c);
	//Returns the array [1, 2, 3, "JScript", 42, "VBScript"]

arrayObj.join(separator)	var a, b;
	a = new Array(0,1,2,3,4);
	b = a.join("-"); // b = "0-1-2-3-4"

arrayObj.pop( )	var n = a.pop(); // n = 4
arrayObj.push([item1 [item2 [. . . [itemN]]]])	a.push(5); // a = [0,1,2,3,5]
arrayObj.reverse( )	a.reverse(); // a = [5,3,2,1,0]
arrayobj.sort(sortFunction)	function numerical_sorter(n1,n2) { return (n1 – n2); } var l = a.sort(); // l = [0,1,2,3,5].

3.6. Một số đối tượng dữ liệu - Array

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript - Array()</title>
</head>
<script language="javascript">
    var i,j;
    var MyArray= new Array(2);
    for(i=0; i<MyArray.length; i++)
        MyArray[i] = new Array(3);
    MyArray[0][0] = "Ryan Dias";    MyArray[0][1] = 1;  MyArray[0][2] = "Photoshop";
    MyArray[1][0] = "Mike Donne";   MyArray[1][1] = 2;  MyArray[1][2]= ".NET";
    for(i=1; i>=0; i--)
    {
        for(j=2; j>=0; j--)
            document.write(", "+ MyArray[i][j]);
        document.write("<br>");
    }
    document.write("<hr>");
    document.write(MyArray);
</script>
<body>
</body>
</html>
```

Mảng nhiều chiều

, .NET, 2, Mike Donne
, Photoshop, 1, Ryan Dias

Ryan Dias,1,Photoshop,Mike Donne,2,.NET

3.6. Một số đối tượng dữ liệu - Active

- Cú pháp :

```
newObj = new ActiveXObject(servername.typename[, location])
Obj = GetObject([pathname] [, class])
```

- ActiveX - Dictionary:

```
var y = new ActiveXObject("Scripting.Dictionary");
y.add ("a", "test");
if (y.Exists("a"))
   document.write("true");
```

- ActiveX - FileSystemObject :

```
var fso = new ActiveXObject("Scripting.FileSystemObject");
var a = fso.CreateTextFile("c:\\testfile.txt", true);
a.WriteLine("This is a test.");
a.Close();
```

3.6. Một số đối tượng dữ liệu - Active

Một ví dụ

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>JavaScript - Array()</title>
</head>
<script language="javascript">
    var fso= new ActiveXObject("Scripting.FileSystemObject");
    var a = fso.CreateTextFile("d:\\testfile.txt", true);
    a.WriteLine("This is a test.");
    a.Close();
//---------------------------------------------------------------------------
    var y = new ActiveXObject("Scripting.Dictionary");
    y.add("computer", "Máy vi tính, Máy điện toán, ..., Công cụ dùng để lưu trữ và xử lý thông tin!");
    document.write("<hr>");
    if (y.Exists("computer"))
    {
        document.write("<br> computer ==> "+y("computer"));
    }
</script>
<body>
</body>
</html>
```

3.7. Cấu trúc điều khiển

- Điều kiện if
- Điều khiển switch
- Vòng lặp for
- Vòng lặp while
- Vòng lặp do … while
- Vòng lặp for … in
- Từ khóa break, continue

3.7. Cấu trúc điều khiển

- Hầu hết cấu trúc điều khiển trong JavaScript giống với C++
- Trong phần này ta chỉ đề cập cấu trúc điều khiển của JavaScript không có trong C++

```
for (variable in [object | array])
{
        statements
}
```

3.8. Hàm

- Dạng thức khai báo chung:

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

- Hàm có giá trị trả về:

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

3.8. Hàm

- Ví dụ:

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

- Gọi hàm:

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

3.9. Lớp - Đối tượng

- Khai báo lớp:

```
function Tên_lớp()
{
    //Khai báo biến thành viên
    this.bien1 = value;
    this.bien2 = value;
}
```

- Định nghĩa hàm thành viên của lớp:

```
// Khai báo phương thức của lớp
Tên_lớp.prototype.Ten_phuongthuc = function(…)
{
    ……
    //return value;
}
```

3.9. Lớp - Đối tượng

```
// Khai báo lớp Diem
function Diem()
{
      this.x = 0; this.y = 0;
}
// Khai báo lớp DoanThang
function DoanThang()
{
      this.diem1 = new Diem();
      this.diem2 = new Diem();
}
DoanThang.prototype.Tinh_Dodai = function()
{
      var chieudai = new Number();
      chieudai = … …
      return chieudai;
}
// Sử dụng lớp
var canhhuyen = new DoanThang();
……
var x = canhhuyen.Tinh_Dodai();
```

4. Một số hàm khác

Table 7-2: Globally Available Methods

Method	Description	Example

>escape()	>Takes a string and returns a string where all non-alphanumeric characters such as spaces, tabs, and special characters have been replaced with their hexadecimal equivalents in the form %xx.	`>var aString="O'Neill & Sons";`
		`// aString = "O'Neill & Sons"`
		`aString = escape(aString);`
		`// aString="O%27Neill%20%26%20Sons"`

>eval()	>Takes a string and executes it as JavaScript code.	`>var x;`
		`var aString = "5+9";`
		`x = aString;`
		`// x contains the string "5+9"`
		`x = eval(aString);`
		`// x will contain the number 14`

>isFinite()	>Returns a Boolean indicating whether its number argument is finite.	`>var x;`
		`x = isFinite('56');`
		`// x is true`
		`x = isFinite(Infinity)`
		`// x is false`

>isNaN()	>Returns a Boolean indicating whether its number argument is **NaN**.	`>var x;`
		`x = isNaN('56');`
		`// x is False`
		`x = isNaN(0/0)`
		`// x is true`
		`x = isNaN(NaN);`
		`// x is true`

4. Một số hàm khác

Table 7-2: Globally Available Methods

>**parseFloat()**	>Converts the string argument to a floating-point number and returns the value. If the string cannot be converted, it returns **NaN**. The method should handle strings starting with numbers and peel off what it needs, but other mixed strings will not be converted.	>var x;
		x = parseFloat("33.01568");
		// x is 33.01568
		x = parseFloat("47.6k-red-dog");
		// x is 47.6
		x = parseFloat("a567.34");
		// x is NaN
		x = parseFloat("won't work");
		// x is NaN

>**parseInt()**	>Converts the string argument to an integer and returns the value. If the string cannot be converted, it returns **NaN**. Like **parseFloat()**, this method should handle strings starting with numbers and peel off what it needs, but other mixed strings will not be converted.	>var x;
		x = parseInt("-53");
		// x is -53
		
		x = parseInt("33.01568");
		// x is 33
		
		x = parseInt("47.6k-red-dog");
		// x is 47
		
		x = parseInt("a567.34");
		// x is NaN
		
		x = parseInt("won't work");
		// x is NaN

>**unescape()**	>Takes a hexadecimal string value containing some characters of the form %xx and returns the ISO-Latin-1 ASCII equivalent of the passed values.	>Var aString="O%27Neill%20%26%20Sons";
		aString = unescape(aString);
		// aString = "O'Neill & Sons"
		aString = unescape("%64%56%26%23");
		// aString = "dV&#"

4. Một số hàm khác

alert(string);

b_answer = confirm("Do you want to do this?");

str_result = window.prompt(*prompt string*, *default value string*);

Phần 5: DOM

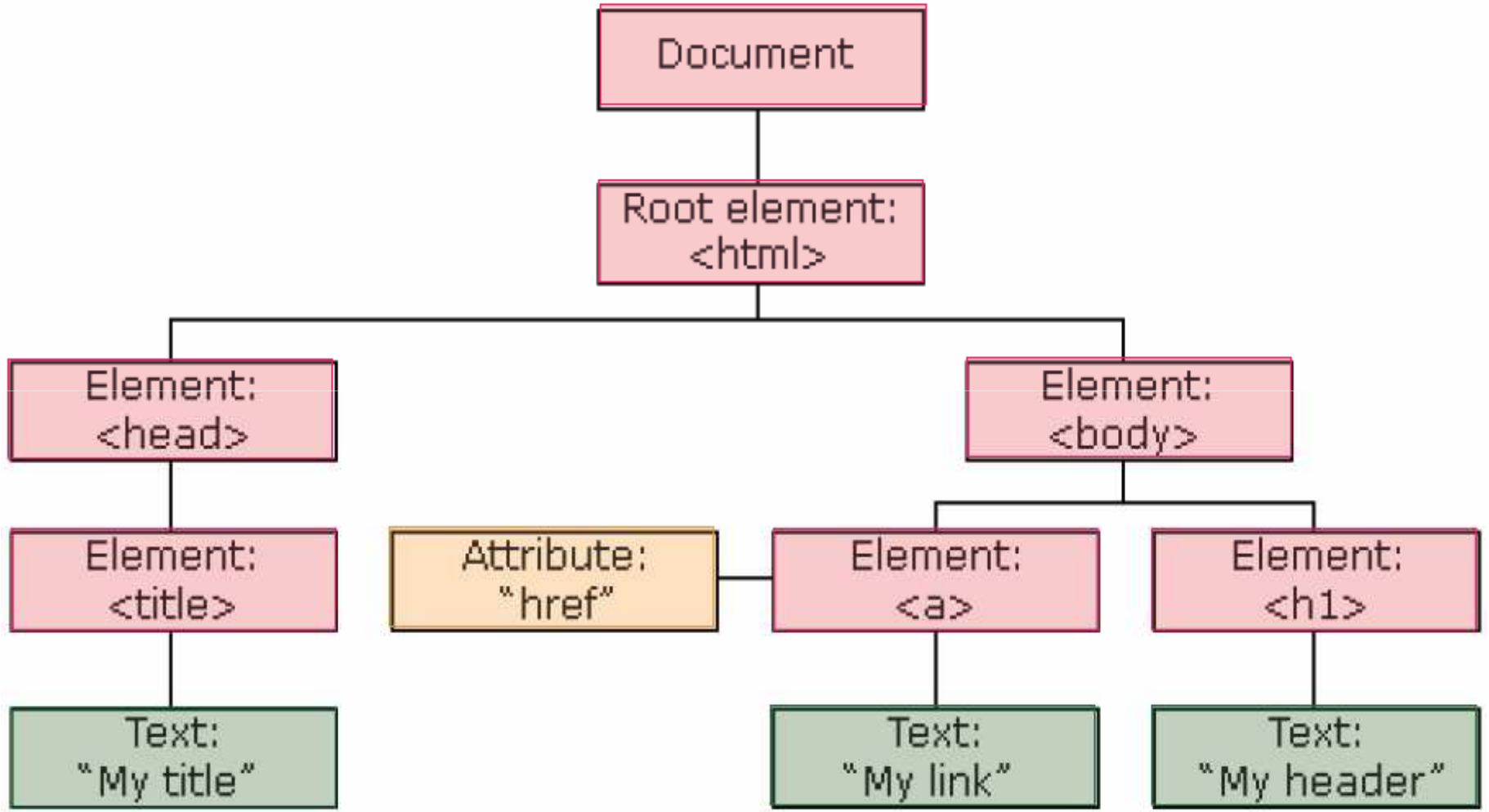
Nội dung

1. Tổng quan về DOM – HTML
2. Một số đối tượng
3. Làm việc với document - DOM
4. Xử lý sự kiện (even)
5. Một số ví dụ

1. Tổng quan về DOM – HTML

- HTML DOM = HTML Document Object Model
- Xem trang web như một cây gồm nhiều nút (node)
- Mỗi nút là một thành phần (tag HTML, thuộc tính, nội dung của tag)
- DOM định nghĩa một cách để truy xuất và điều khiển các thành phần trong 1 trang web

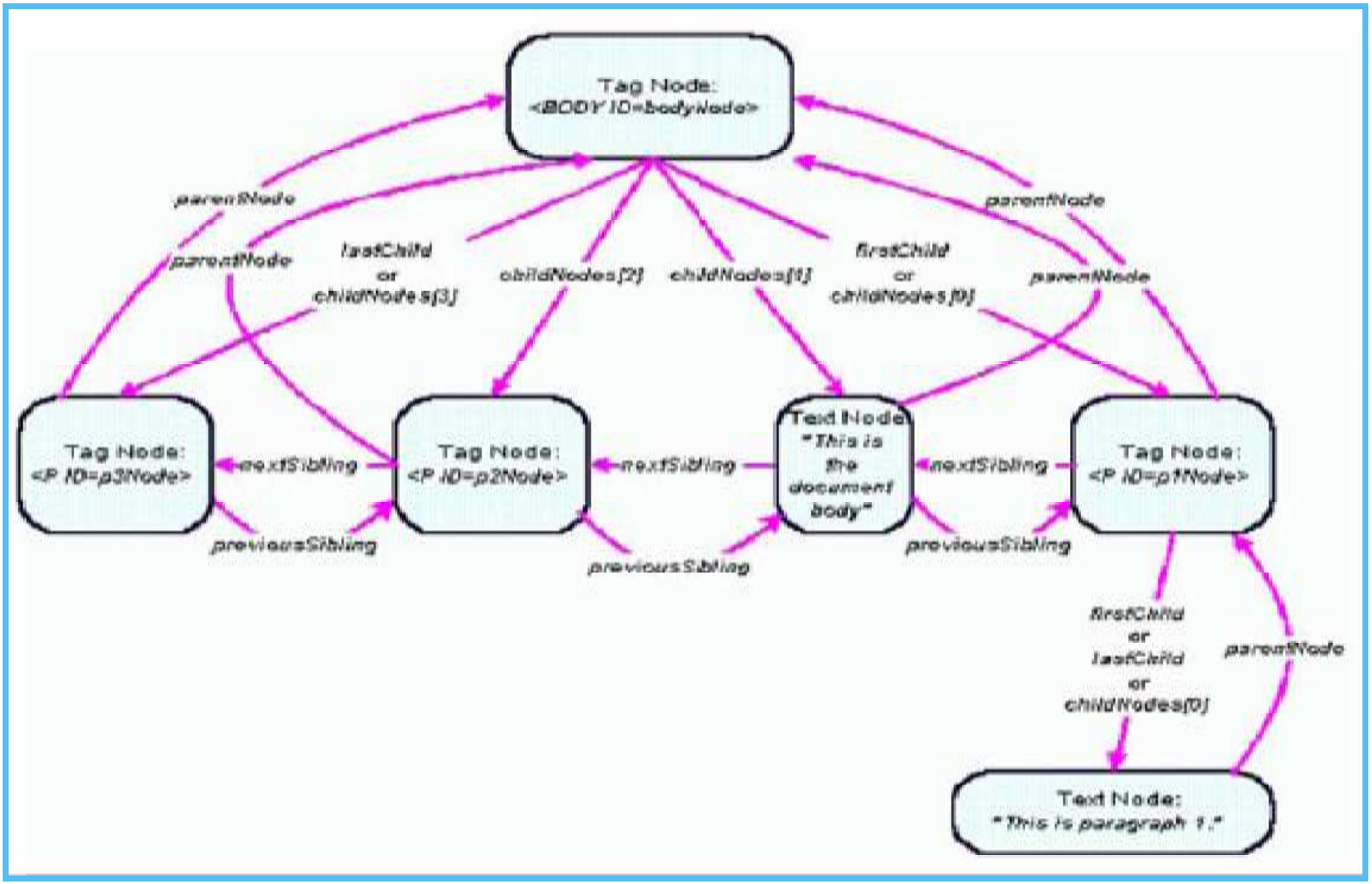
1. Tổng quan về DOM – HTML



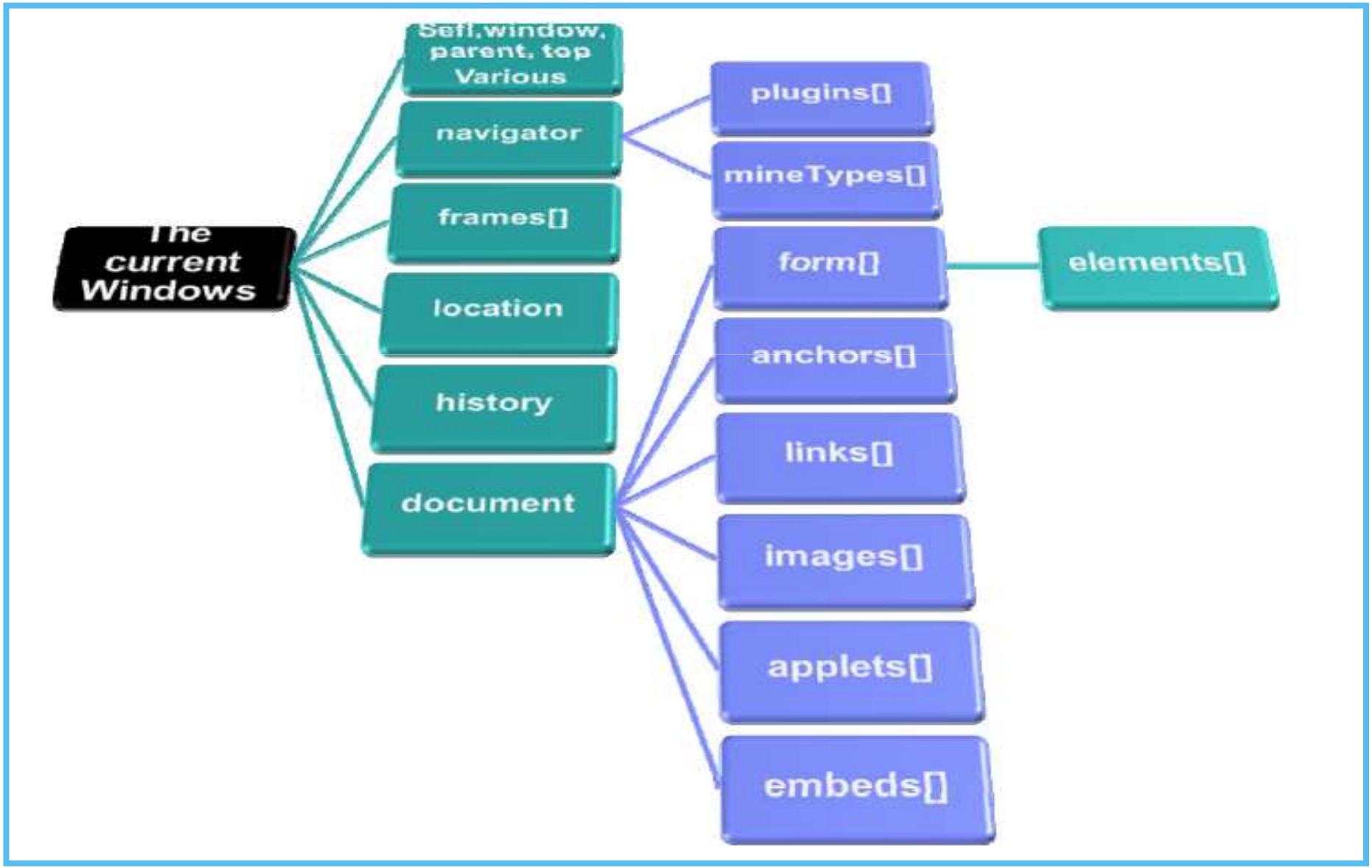
1. Tổng quan về DOM – HTML

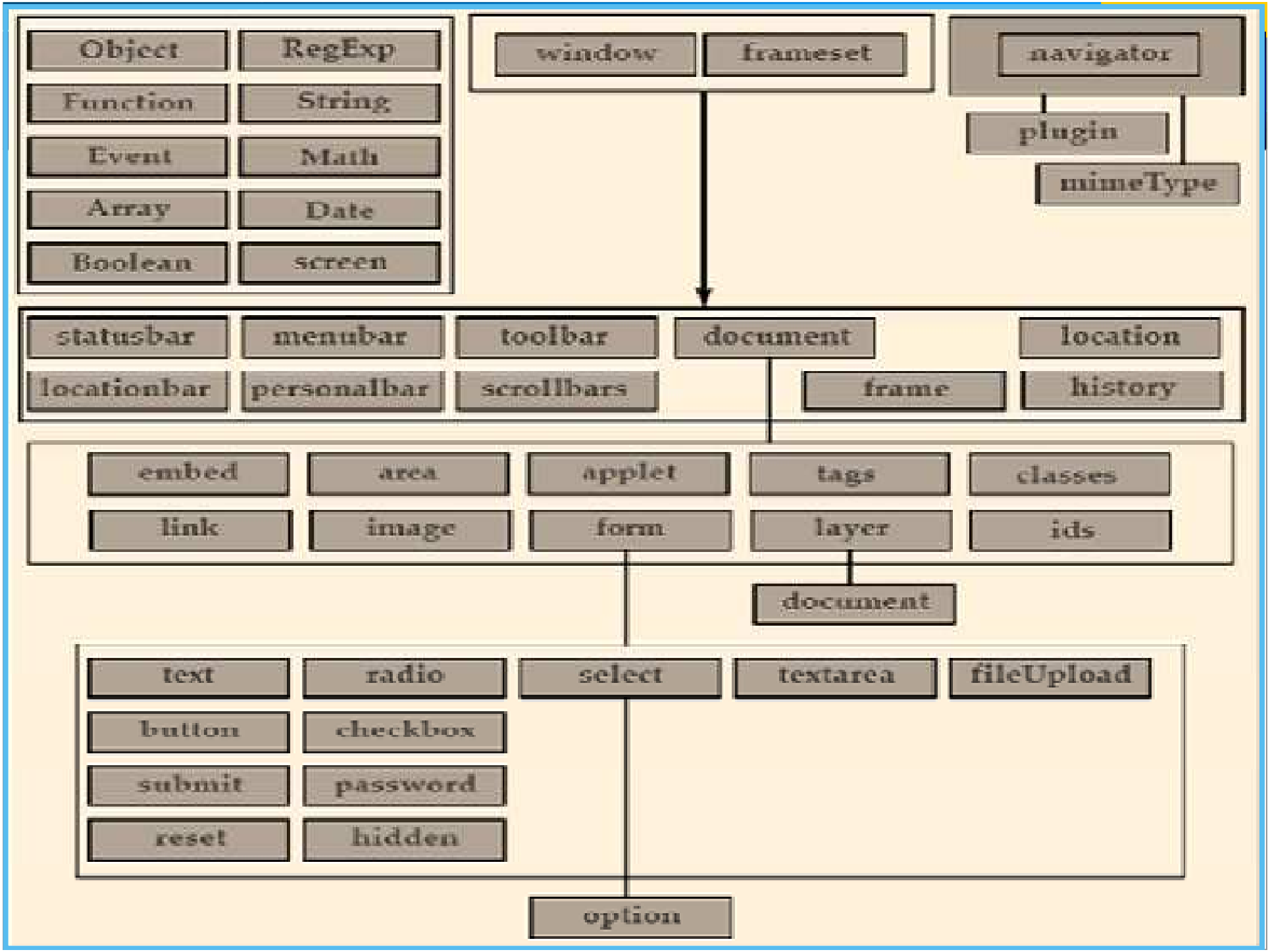

```
1  <html>
2  <head>
3      <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
4      <title>
5          Simple DOM Demo
6      </title>
7  </head>
8  <body id="bodyNode">
9      <p id="p1Node">
10         This is paragraph 1.
11     </p>
12
13     This is the document body
14
15     <p id="p2Node">
16     </p>
17
18     <p id="p3Node">
19     </p>
20 </body>
21 </html>
```

1. Tổng quan về DOM – HTML



1. Tổng quan về DOM – HTML




Object
RegExp
Function
String
Event
Math
Array
Date
Boolean
screen
window
frameset
navigator
plugin
mimeType
statusbar
menubar
toolbar
document
location
locationbar
personalbar
scrollbars
frame
history
embed
area
applet
tags
classes
link
image
form
layer
ids
document
text
radio
select
textarea
fileUpload
button
checkbox
submit
password
reset
hidden
option

1. Tổng quan về DOM – HTML

- Mỗi đối tượng DOM đều có danh sách thuộc tính (Properties) và danh sách các phương thức (Method) tương ứng.
- objectName.propertyName= value
- Vídụ: document.bgColor= “blue”;
- objectName.methodName()
- Vídụ: window.focus();

2. Một số đối tượng trong DOM

1. window
2. location
3. history
4. navigator
5. document
6. image
7. form
8. element

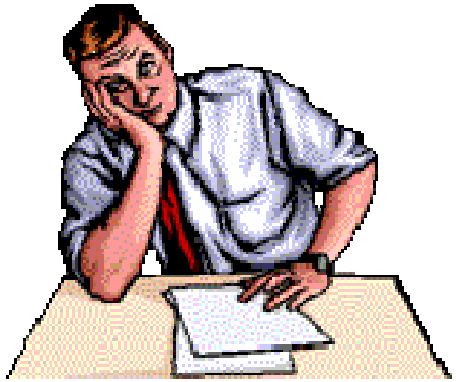
2.1. DOM - window

- Là thể hiện của đối tượng cửa sổ trình duyệt
- Tồn tại khi mở 1 tài liệu HTML
- Sử dụng để truy cập thông tin window
- Điều khiển các sự kiện xảy ra trong window
- Nếu tài liệu định nghĩa nhiều frame, browser tạo 1 window object cha và các window object con cho từng frame

2.1. DOM - window

Thuộc tính

- document
- history
- location
- parent
- frames[]
- name
- status
- event
- screen
- …



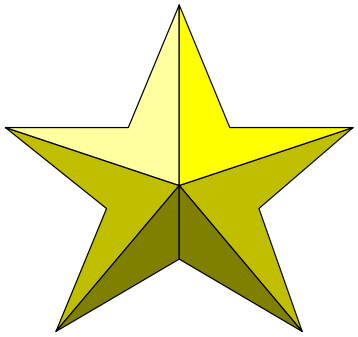
Phương thức

- alert
- confirm
- prompt
- blur
- focus
- open
- close
- setTimeout
- setInterval
- …

2.2. DOM - location

Thuộc tính

- hash
- host
- hostname
- href
- pathname
- port
- protocol
- search



Phương thức

- assign(url)
- reload()
- replace(url)

Chứa thông tin hiện tại của URL

2.3. DOM - history

Thuộc tính

- length

Phương thức

- back()
- go(url)
- forward()

Cung cấp danh sách các URL đã được duyệt bởi người sử dụng.

2.4. DOM - navigator

Thuộc tính

- appName
- appVersion
- appCodeName
- cookieEnabled
- online
- platform
- …

Phương thức

- javaEnabled()
- …

Cung cấp thông tin về trình duyệt.

2.5. DOM - document

- Biểu diễn cho toàn bộ các thành phần trong 1 tài liệu HTML
- Dùng để lấy thông tin về tài liệu, các thành phần HTML và xử lý sự kiện

2.5. DOM - document

Thuộc tính

- bgColor
- fgColor
- aLinkColor
- linkColor
- vlinkColor
- lastModified
- location
- referrer
- title

Phương thức

- clear()
- close()
- open(…,…)
- write(text)
- writeln(text)
- getElementById(“id”)
- getElementByName(“Name”)
- getElementByTagName(“tagName”)
- createTextNode(“ text ")
- createElement(“HTMLtag")

Tập hợp

anchors [ ]
forms [ ]
frames [ ]
images [ ]
links [ ]

2.6. DOM – image

- Truy xuất đến tag <img> trên trang web
- Có thể truy xuất thông qua:
  - document.images[index]
  - document.images[“ImageName”]
  - document.ImageName
- Một số thuộc tính của Image Obj
  - alt, border, classname, title,
  - width, height, hspace, vspace,
  - id, name, src, lowsrc, longDesc,
  - isMap, complete

2.7. DOM – form

- Dùng để truy xuất đến tag <form> trên trang web
- Có thể truy xuất thông qua
  - document.forms[index]
  - document.forms[“FormName”]
  - document.FormName
- **Một số thuộc tính**
  - action, method, id, name, target
  - classname, title, language, dir
  - elements[ ]
- **Một số phương thức**
  - reset( ), submit( )

2.8. DOM – element

- Tương ứng với form field.
- Cách truy xuất
  document.formName.*controlName*

```
<form name="searchForm" action="xuly.php">
    <input type="text" name="entry">
    <input type="submit" name="sender" value="Search">
</form>
---------------------------------------------------------------
document.searchForm.entry
document.searchForm.elements[0]
document.forms["searchForm"].elements["entry"]
document.forms["searchForm"].entry
```

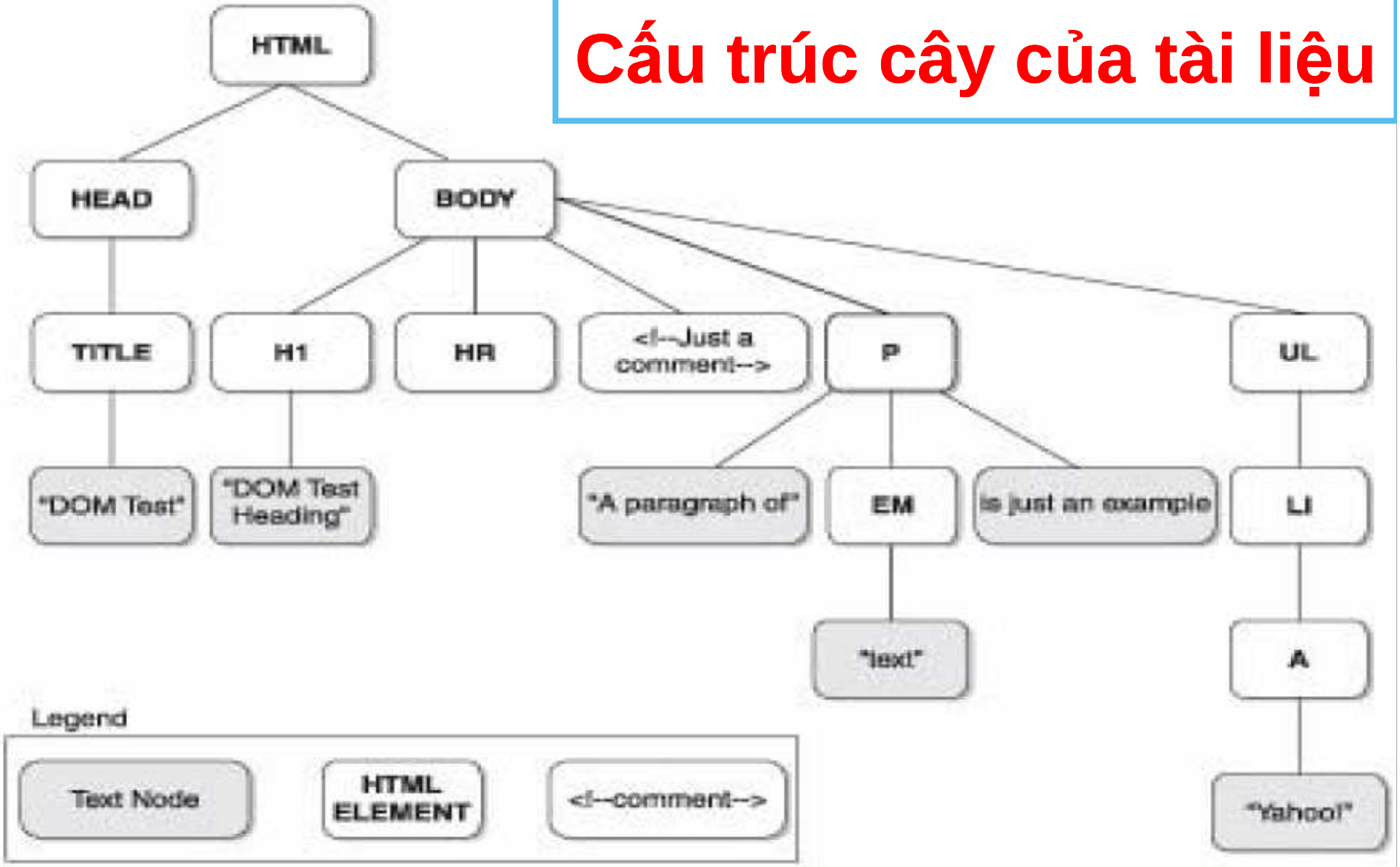
3. Làm việc với document - DOM

Biểu diễn nội dung tài liệu theo cấu trúc cây

```
1  <html>
2  <head>
3      <title>DOM Test</title>
4  </head>
5  <body>
6      <h1>DOM Test Heading</h1>
7      <hr />
8      <!--Just a comment -->
9      <p id="p1">A paragraph of <em>text</em>is just an example</p>
10     <ul>
11         <li><a href="http://www.yahoo.com"> Yahoo! </a></li>
12      </ul>
13 </body>
14 </html>
```

3. Làm việc với document - DOM

Cấu trúc cây của tài liệu



3. Làm việc với document - DOM

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

3. Làm việc với document - DOM

- **getElementById(“id1”):** trả về node có giá trị thuộc tính id = **id1**
- **Ví dụ**:

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

3. Làm việc với document - DOM

- **getElementsByName**(**name1**): trả về danh sách node có giá trị của thuộc tính name = **name1**
- Ví dụ:

```
var nodeList=document.getElementsByName("name1");
for(var i=0;i<nodeList.length;++i)
{
    var nodeName= nodeList(i).nodeName;
    var nodeType=nodeList(i).nodeType;
    var nodeValue=nodeList.item(i).nodeValue;
}
```

3. Làm việc với document - DOM

- **createElement**(**nodeName**): tạo ra một node HTML mới tùy theo đối số **nodeName**
- Ví dụ:

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

3. Làm việc với document - DOM

- **createTextNode**(**content**): tạo ra một textnode mới với nội dung tùy theo đối số **content**
- Ví dụ:

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

3. Làm việc với document - DOM

- **nodeName.appendChild(newNode)**: thêm node mới **newNode** vào cuối danh sách các node con của một node.
- Ví dụ:

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

3. Làm việc với document - DOM

- **nodeName.insertBefore(newNode,childRef**): chèn **newNode** vào trước **childRef** trong danh sách các node con của một node.

```
// <p id="id1"><img src="images/test.gif"></p>
//-------------------------------------------------------------
var pNode=document.getElementById("id1");
var firstChild=pNode.firstChild;
var newNode=document.createTextNode("Some text");
pNode.insertBefore(newNode,firstChild);
//-------------------------------------------------------------
// <p id="id1">
//  Some text <img src="images/test.gif">
//</p>
```

3. Làm việc với document - DOM

- **nodeName.removeChild(child)**: xóa **child** trong danh sách các node con của một node gọi phương thức, trả về node bị xóa.

```
// <p id="id1"><img src="images/test.gif">Hình ảnh</p>
//-------------------------------------------------------------
var pNode=document.getElementById("id1");
if (pNode.hasChildNodes())
        var rmNode=pNode.removeChild(pNode.lastChild);
//-------------------------------------------------------------
// <p id="id1">
//        <img src="images/test.gif">
//</p>
```

3. Làm việc với document - DOM

- **replaceChild (newChild, oldChild)**

  Thay thế node **oldChild** bằng node **newChild** trong danh sách các node con của node hiện hành

```
Ví dụ:
var replace = document.getElementById("isReplaced");
if (replace)
{
    var newNode = document.createElement("strong");
    var newText =
          document.createTextNode("strongelement");
    newNode.appendChild(newText);
    replace.parentNode.replaceChild(newNode,replace);
}
```

3. Làm việc với document - DOM

- **innerHTML**

  Chỉ định nội dung HTML bên trong một node.

```
Ví dụ:
//<p id="para1" >
// some text
//</p>
var theElement = document.getElementById("para1");
theElement.innerHTML = "Some <b> new </b> text";

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

3. Làm việc với document - DOM

- **innerText**

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

Ví dụ:

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

3. Làm việc với document - DOM

- **setAttribute (attributeName , value)**

  Chỉ định attribute của một node với giá trị là value.

Ví dụ:

```
<font id="font1" >
    Some text
</font>
<script language="javascript" >
    var fontNode = document.getElementById("font1");
    fontNode.setAttribute("color","red");
    fontNode.setAttribute("size","5");
</script>
<font id="font1" color="red" size="5">
    Some text
</font>
```

3. Làm việc với document - DOM

- **getAttribute (attributeName)**

  Lấy giá trị của một attribute trong node

```
Ví dụ:
   var font1 = document.getElementById("font1");
   alert(font1.getAttribute("color"));
```

- **removeAttribute (attributeName)**

  Hủy một attribute trong node

```
Ví dụ:
   font1.removeAttribute("color");
   font1.removeAttribute("size");
```

3. Làm việc với document - DOM

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

Ví dụ:

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

3. Làm việc với document - DOM

- Thay đổi định dạng css thông qua thuộc tính className

Ví dụ:

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

3. Làm việc với document - DOM

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

Ví dụ:

```
<head>
        <script language="javascript" >
         function changeSkin()
         {
              document.getElementById("myStyle").href =
                                        "css/style2.css";
         }
        </script>
      <link id="myStyle" rel="stylesheet"
      type="text/css" href="css/style1.css" />
</head>
<body>
       <p class="style1">
            Hello world
       </p>
      <input type="button" onclick="changeSkin()">
```

4. Xử lý sự kiện

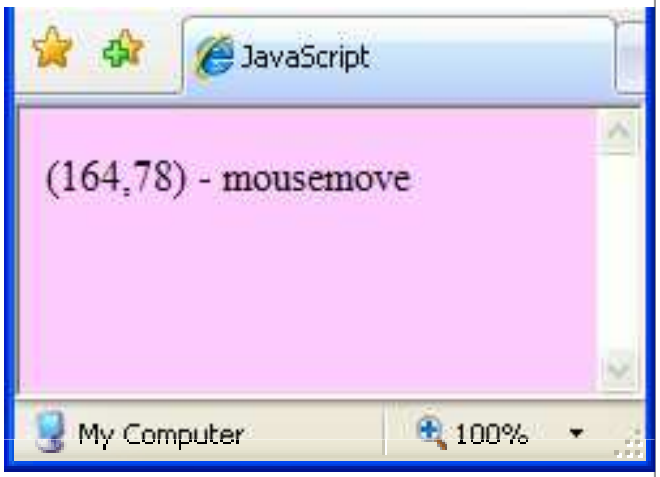
1. Event Object
2. Event Handle
3. Xử lý sự kiện

4.1. Event Object

- Events là các sự kiện xảy ra trên trang Web
- Do người dùng hoặc do hệ thống tạo ra
- Mỗi sự kiện sẽ liên quan đến một event object
- Event cung cấp các thông tin
  - Loại event
  - Vị trí con trỏ tại thời điểm xảy ra sự kiện
- Ví dụ: Xem Slide sau

4.1. Event Object

```
1  <html>
2  <head>
3  <title>JavaScript</title>
4  <script language='javascript'>
5  function doClick()
6  {
7      var x,y,type;
8      x=event.clientX;
9      y=event.clientY;
10     type=event.type;
11     var str="(" + x + "," + y + ") - " + type;
12     var element=document.getElementById("toado");
13     element.innerHTML=str;
14 }
15 document.onclick=doClick;
16 document.onmousemove=doClick;
17 </script>
18 </head>
19 <body>
20     <p id="toado"></p>
21 </body>
22 </html>
```



4.2. Event Handle

- Giúp cho người lập trình bắt và xử lý các sự kiện của các đối tượng trong trang web.
- Cú pháp

  <TAG eventHandler="JavaScript Code">
- Ví dụ:

```
<INPUT TYPE=“button” NAME=“docode”
       onclick="DoOnClick();”>
<INPUT TYPE="button” NAME=“Button1”
       VALUE="Open Sesame!”
       onClick=”window.open('mydoc.html','newWin')”>
```

4.2. Event Handle

onabort	onload
onblur	onmousedown
onchange	onmousemove
onclick	onmouseout
onerror	onmouseover
onfocus	onmouseup
onkeydown	onreset
onkeyup	onresize
onselect	onsubmit
	onunload

4.3. Xử lý sự kiện

- Cú pháp

  <TAG eventHandler= "JavaScript Code">

- Ví dụ:

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

- Lưu ý: Dấu "…" và '…'

4.3. Xử lý sự kiện

```
1   <html>
2   <SCRIPT LANGUAGE="JavaScript">
3       function greeting()
4       {
5           alert("Welcometo my world");
6       }
7   </SCRIPT>
8   <body onLoad="greeting()">
9   </body>
10  </html>
```

Ví dụ

object.eventhandler = functionname;

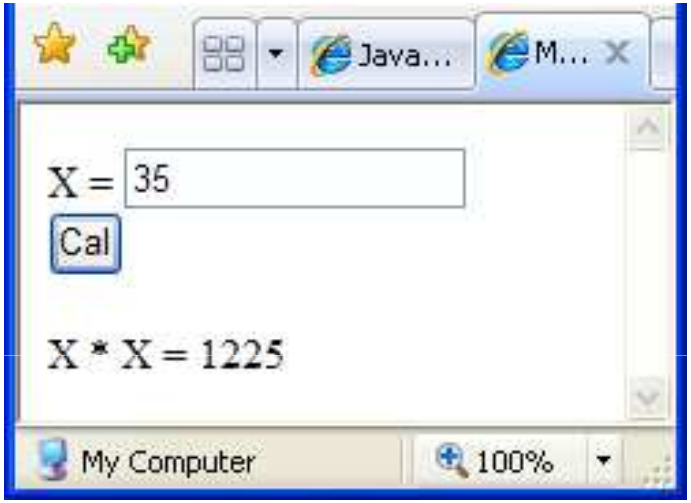
```
<SCRIPT LANGUAGE="JavaScript">
        function greeting() {
                alert("Welcome to my world");
        }
        window.onload = greeting;
</SCRIPT>
```

4.3. Xử lý sự kiện

	Blur	Click	Change	Focus	Load	Mouseover	Select	Submit	Unload
Button		x							
Checkbox		x							
Document					x				x
Form								x	
Link		x				x			
Radio		x							
Reset		x							
Selection	x		x	x					
Submit		x							
Text	x		x	x			x		
Textarea	x		x	x			x		

5. Một số ví dụ

```
<script language="JavaScript">
function compute(frm)
{
      var x = frm.expr.value;
      result.innerHTML= x*x;
}
</script>
<form>
      X = <INPUT TYPE="text" NAME="expr" SIZE=15> <BR>
      <input type="button“ value=“Cal” ONCLICK="compute(this.form)">
      <p>X * X = <SPAN ID="result"></SPAN></p>
</form>
```



5. Một số ví dụ

```
1  <html>
2  <head>
3  <script language=JavaScript type="text/javascript">
4  function myPara_onmouseout()
5  {
6     myPara.innerText = "Roll your mouse over my text";
7     myPara.style.color = "Black"
8  }
9  function myPara_onmouseover()
10 {
11    myPara.innerText = "Wow that feels good!!!";
12    myPara.style.color = "Red"
13 }
14 </script>
15 </head>
16 <body>
17 <P id="myPara"  onmouseout="return myPara_onmouseout()"
18                 onmouseover="return myPara_onmouseover()">
19       Roll your mouse over my text
20 </P>
21 </body>
22 </html>
```

5. Một số ví dụ

- Ví dụ: onFocus – onBlur
- Xảy ra khi một thành phần HTML bị focus (onFocus) và mất focus (onBlur)
- Ví dụ:

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

5. Một số ví dụ

```
1  <html>
2  <head>
3  <script language="javascript">
4  function showLink(num)
5  {
6      if(num==1)
7      {
8          document.forms[0].elements[0].value="You have selected Aptech";
9      }else{
10         document.forms[0].elements[0].value="";
11     }
12 }
13 </script>
14 </head>
15 <body>
16     <form>
17         <input type=text size=60>
18     </form>
19     <a href="#"  onMouseOver="showLink(1)" onMouseOut="showLink(0)">
20         Aptech
21     </a>
22 </body>
23 </html>
```

Tham khảo

Bài giảng này có tham khảo:

- **Slide bài giảng "Thiết kế và lập trình Web", ĐH KHTN TpHCM, version 2007.**
- **Slide bài giảng "Thiết kế Web", Đào Việt Cường, Khoa CNTT, ĐH Sư Phạm Hà Nội.**

Câu hỏi và thảo luận