

227 Nguyễn Văn Cừ – Quận 5 – Tp. Hồ Chí Minh Tel: 8351056 – Fax 8324466 – Email: ttth@hcmuns.edu.vn

# TÀI LIỆU HƯỚNG DẪN GIẢNG DẠY



# CHƯƠNG TRÌNH KỸ THUẬT VIÊN Học phần 5 LẬP TRÌNH WEB CƠ BẢN



# Mục lục

Müc	; lục	1
GIỚI	THIỆU	5
GIÁC	O TRÌNH LÝ THUYẾT	6
TÀI	LIỆU THAM KHẢO	6
HƯỚ	NG DẪN PHẦN LÝ THUYẾT	7
Bài	1 TẠO NỘI DUNG TRANG WEB VÀ ĐỊNH DẠNG	7
I.	Giới thiệu bài học	8
II.	Giới thiệu về WWW và HTML	9
	II.1. Các khái niệm về WWW	10
	II.2. Các khái niệm về HTML	11
III.	Cấu trúc của một trang Web	12
IV.	Các tag cơ bản trong HTML	14
٧.	Định dạng Text	15
	V.1. Căn bản về định dạng văn bản và cách hiển thị	16
	V.2. Font chữ, màu sắc và canh lề	18
	V.3. Định dạng trước nội dung văn bản	19
	V.4. Quotation, deleted và inserted	20
	V.5. Một số tag định dạng khác	21
VI.	Liên kết các trang web	22
	VI.1. Mối liên kết và địa chỉ liên kết	23
	VI.2. Tạo liên kết trong trang web	25
VII.	Danh sách	26
VIII.	Hình ảnh	29
	VIII.1. File hình ảnh và trang web	30
	VIII.2. Đưa hình ảnh vào trang web	31
Bài	2 XÂY DỰNG CẤU TRÚC TRANG WEB	33
I.	Giới thiệu bài học	34
II.	Bång (Table)	35
	II.1. Cấu trúc của bảng	36
	II.2. Sử dụng các thuộc tính của bảng	38
III.	FrameSet và Frame	39
	III.1. Hiển thị nhiều trang trên web browser	40



	III.2.	Sử dụng Frameset và Frame	41
	III.3.	Frame "tĩnh" và frame "động"	42
IV.	Bố cụ	ục của website	43
	IV.1.	Bố trí các thành phần trên trang web	44
	IV.2.	Xây dựng bố cục cho website	44
	IV.2.	Xây dựng bố cục cho website	45
Bài	3 THIẾ	ÉT KÉ FORM NHẬP LIỆU	46
l.	Giới t	thiệu bài học	47
II.	НТМІ	L Form	48
	II.1.	Các điều khiển trên form	49
	11.2.	Tag <input/>	50
	II.3.	Các điều khiển cơ bản	51
	11.4.	Submit & Reset	52
	11.5.	Tag <textarea>&lt;/td&gt;&lt;td&gt; 54&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;II.6.&lt;/td&gt;&lt;td&gt;Tag &lt;select&gt;&lt;/td&gt;&lt;td&gt; 55&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;III.&lt;/td&gt;&lt;td&gt;Quá&lt;/td&gt;&lt;td&gt;trình xử lý form&lt;/td&gt;&lt;td&gt;56&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;IV.&lt;/td&gt;&lt;td&gt;Ưu và&lt;/td&gt;&lt;td&gt;à nhược điểm của các bước xử lý&lt;/td&gt;&lt;td&gt;57&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;Bài&lt;/td&gt;&lt;td&gt;4 MÔ I&lt;/td&gt;&lt;td&gt;HÌNH ỨNG DỤNG WEB&lt;/td&gt;&lt;td&gt;58&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;I.&lt;/td&gt;&lt;td&gt;Giới t&lt;/td&gt;&lt;td&gt;thiệu bài học&lt;/td&gt;&lt;td&gt;59&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;II.&lt;/td&gt;&lt;td&gt;Mô h&lt;/td&gt;&lt;td&gt;ìình ứng dụng&lt;/td&gt;&lt;td&gt;60&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;II.1.&lt;/td&gt;&lt;td&gt;Mô hình ứng dụng 2 lớp&lt;/td&gt;&lt;td&gt; 61&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;11.2.&lt;/td&gt;&lt;td&gt;Mô hình ứng dụng 3 lớp&lt;/td&gt;&lt;td&gt; 62&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;III.&lt;/td&gt;&lt;td&gt;Ứng d&lt;/td&gt;&lt;td&gt;dụng Web với mô hình Client/Server&lt;/td&gt;&lt;td&gt;63&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;IV.&lt;/td&gt;&lt;td&gt;Micro&lt;/td&gt;&lt;td&gt;osoft Internet Information Server&lt;/td&gt;&lt;td&gt;64&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;IV.1.&lt;/td&gt;&lt;td&gt;Giới thiệu Internet Information Server&lt;/td&gt;&lt;td&gt; 65&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;IV.2.&lt;/td&gt;&lt;td&gt;Demo: Cài đặt Web server&lt;/td&gt;&lt;td&gt; 66&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;٧.&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;niệm về Script&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;VI.&lt;/td&gt;&lt;td&gt;Quá&lt;/td&gt;&lt;td&gt;trình xử lý một yêu cầu của Web Server&lt;/td&gt;&lt;td&gt;69&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;Bài&lt;/td&gt;&lt;td&gt;5 SỬ [&lt;/td&gt;&lt;td&gt;DỤNG CÁC ĐỐI TƯỢNG CƠ BẢN TRONG ASP&lt;/td&gt;&lt;td&gt;70&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;I.&lt;/td&gt;&lt;td&gt;Giới t&lt;/td&gt;&lt;td&gt;thiệu bài học&lt;/td&gt;&lt;td&gt;71&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;II.&lt;/td&gt;&lt;td&gt;Giới t&lt;/td&gt;&lt;td&gt;thiệu MS Visual InterDEV&lt;/td&gt;&lt;td&gt;72&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;II.1.&lt;/td&gt;&lt;td&gt;Demo: Tạo web project bằng Visual InterDEV&lt;/td&gt;&lt;td&gt; 73&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;III.&lt;/td&gt;&lt;td&gt;Ngôn&lt;/td&gt;&lt;td&gt;n ngữ VBScript&lt;/td&gt;&lt;td&gt;74&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;III.1.&lt;/td&gt;&lt;td&gt;Tham khảo về ngôn ngữ VBScript&lt;/td&gt;&lt;td&gt; 75&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;IV.&lt;/td&gt;&lt;td&gt;File .&lt;/td&gt;&lt;td&gt;asp&lt;/td&gt;&lt;td&gt;76&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;IV.1.&lt;/td&gt;&lt;td&gt;· '&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;&lt;/td&gt;&lt;td&gt;IV.2.&lt;/td&gt;&lt;td&gt;Chọn VBScript để viết trang .asp&lt;/td&gt;&lt;td&gt; 80&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;٧.&lt;/td&gt;&lt;td&gt;Đối tư&lt;/td&gt;&lt;td&gt;ượng Response và Request&lt;/td&gt;&lt;td&gt;81&lt;/td&gt;&lt;/tr&gt;&lt;/tbody&gt;&lt;/table&gt;</textarea>	



	V.1.	Xử lý nhập/xuất dữ liệu trên trang Web	82
	V.2.	Sử dụng đối tượng Response	85
	V.3.	Sử dụng đối tượng Request	87
Bài	6 SỬ D	ŲNG CÁC ĐỐI TƯỢNG QUẢN LÝ ỨNG DỤNG	89
l.	Giới th	niệu bài học	90
II.	Đối tư	rợng Session, Application và file Global.asa	91
III.	Đối tư	ong Session	92
	III.1.	Sử dụng đối tượng Session	93
	III.2.	Biến toàn cục trong Session	95
IV.	Quản	lý các Session	97
	IV.1.	SessionID và Cookies	98
	IV.2.	Sử dụng Cookies	99
٧.	Đối tư	drg Application	101
VI.	File G	lobal.asa	103
	VI.1.	Các sự kiện của đối tượng Application	104
	VI.2.	Các sự kiện của đối tượng Session	106
VII.	Đối tư	ong Server	108
	VII.1.	Sử dụng đối tượng Server	109
	VII.2.	MapPath	111
	VII.3.	Transfer	112
Bài	7 SỬ D	ŲNG ADODB: KẾT NỐI VÀ HIỂN THỊ DỮ LIỆU	113
l.	Giới th	niệu bài học	114
II.	Thư v	iên ADO	115
III.	Kết nó	ối với CSDL bằng ADODB	116
	III.1.	Cơ sở dữ liệu	
	III.2.	Thông tin kết nối đến CSDL	118
	III.3.	Demo: Tạo DSN	
	III.4.	Kết nối đến CSDL bằng đối tượng ADODB.Connection	
	III.5.	Bảo mật thông tin kết nối đến CSDL	122
	III.6.	Tạo kết nối đến CSDL sử dụng file đính kèm	123
IV.	Truy o	cập và hiển thị dữ liệu	
	IV.1.	Truy cập dữ liệu bằng RecordSet	125
	IV.2.	Sử dụng đối tượng RecordSet	127
	IV.3.	Di chuyển giữa các mẩu tin	128
	IV.4.	Thêm, xoá, cập nhật dữ liệu	130
	IV.5.	Sử dụng tập hợp Fields	132
	IV.6.	Hiển thị dữ liệu trong RecordSet	133



Bài 8 SỬ DỤNG ADODB: CẬP NHẬT DỮ LIỆU			136
l.	Giới t	thiệu bài học	137
II.	Thên	n, xoá, cập nhật dữ liệu	138
		Cập nhật dữ liệu trong ứng dụng web	
	II.2.	Cấu trúc dữ liệu thử	140
	II.3.	Xác định thông tin cần cập nhật	141
	11.4.	Các câu truy vấn cập nhật dữ liệu	142
	11.5.	Thực hiện cập nhật dữ liệu	143
III.	Đối t	ượng Connection	144
IV.	Đối t	ượng Command	145
ĐỀ 1	THI M <i>Å</i>	ÃU CUỐI MÔN	149



# GIỚI THIỆU

Sau khi hoàn thành khóa học, học viên sẽ có khả năng:

- 1. Nắm vững ngôn ngữ HTML
- 2. Xây dựng các trang web động với ASP 3.0
- 3. Nắm vững và sử dụng bộ thư viện chuẩn ADODB để lập trình CSDL trong ứng dụng web

Với thời lượng là 36 tiết LT và 60 tiết TH được phân bổ như sau:

STT	Bài học	Số tiết LT	Số tiết TH
1	Tạo nội dung trang web và định dạng	4	8
2	Xây dựng cấu trúc trang web	2	2
3	Thiết kế form nhập liệu	4	8
4	Mô hình ứng dụng web	2	2
5	Sử dụng các đối tượng cơ bản trong ASP	6	8
6	Sử dụng các đối tượng quản lý ứng dụng	4	8
7	Sử dụng ADODB: Kết nối và hiển thị dữ liệu	6	12
8	Sử dụng ADODB: Cập nhật dữ liệu	8	12

Tổng số tiết: 36 60



# GIÁO TRÌNH LÝ THUYẾT

Sử dụng giáo trình "Lập Trình Ứng Dụng Web" của tác giả Nguyễn Thái Hưng, in lần thứ 1.

# TÀI LIỆU THAM KHẢO



# HƯỚNG DẪN PHẦN LÝ THUYẾT

# Bài 1

# TẠO NỘI DUNG TRANG WEB VÀ ĐỊNH DẠNG

# Tóm tắt

Lý thuyết 4 tiết - Thực hành 8 tiết

Mục tiêu		Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Trình bày với học viên	I. Gió	ri thiệu bài học	1.1 - 1.4	
những khái niệm cơ bản liên quan đến World Wide	II. Gió	ri thiệu về WWW và HTML		
Web và HTML. Hướng dẫn cho học viên cách xây	III. Cấu	ı trúc của một trang Web		
dựng định dạng một trang	IV. Các	c tag cơ bản trong HTML		
web có văn bản và hình ảnh.	V. Địn	h dạng Text		
Sau khi hoàn tất bài học	VI. Liêr	n kết các trang web		
này, học viên có thể:	VII. Dar	nh sách		
Trình bày các khái niệm liên quan đến WWW.	VIII. Hìn	h ảnh		
<ul> <li>Mô tả cấu trúc cơ bản của một trang HTML</li> </ul>				
<ul> <li>Sử dụng các tag</li> <li>HTML để định</li> <li>dạng văn bản</li> </ul>				
<ul> <li>Xây dựng mối liên kết giữa các trang HTML</li> </ul>				
<ul> <li>Đưa các hình ảnh vào trang HTML</li> </ul>				



# I. Giới thiệu bài học

# Nội dung bài học

- Giới thiệu về WWW và HTML
- Cấu trúc của một trang web
- Các tag cơ bản trong HTML
- Định dạng Text
- Liên kết các trang web (Link)
- Danh sách (List)
- Hình ảnh (Image)

Iternet bao gồm một hệ thống mạng các máy tính được kết nối toàn cầu đề trao đổi thông tin với nhau. Có thể phân chia các máy tính trên Internet thành hai loại là Web Server và Web client (chiếm đa số). Mỗi web site trên Internet cần một hay một số Web Server để cung cấp thông tin cho mọi người. Các thông tin trao đổi giữa người dùng và Web Server là những trang HTML hay đơn giản là những trang web.

Bài học này sẽ trình bày cho bạn những khái niệm cơ bản về Web site, Web server, ...và cấu trúc của một trang web. Tiếp theo, bạn sẽ được hướng dẫn cách xây dựng các trang web, định dạng nội dung các đoạn văn bản, đưa hình ảnh vào trang web và tạo ra các mối liên kết giữa các trang.

Sau khi kết thúc bài học, học viên có thể:

- Định nghĩa được thế nào web site, phân biệt chức năng của Web Server và Web client
- Định nghĩa được cấu trúc cơ bản của một trang web
- Thực hiện việc định dạng văn bản như đổi font chữ, màu sắc, heading, danh sách,...
- Đưa hình ảnh vào trong trang web
- Tạo mối liên kết giữa các trang web



# II. Giới thiệu về WWW và HTML

# Giới thiệu về WWW và HTML

- Các khái niệm về WWW
- Các khái niệm về HTML

WWW và HTML là hai khái niệm cơ bản nhất mà người lập trình web phải nắm. Phần này trình bày cho bạn tổ chức của www và cách làm việc của nó, qua đó giúp bạn hiểu được rằng www chính là cơ sở hạ tầng giúp các máy tính chia sẻ thông tin còn ngôn ngữ HTML là công cụ để thể hiện thông tin.



# II.1. Các khái niệm về WWW

#### Các khái niệm về WWW

- Là tên thường dùng khi nói về Internet
- Internet bao gồm nhiều máy tính kết nối với nhau để trao đổi thông tin.
- Các máy tính liên lạc với nhau dựa trên giao thức HTTP để trao đổi các trang web
- Máy lưu trữ và cung cấp trang web gọi là web server, máy nhận các trang web gọi là web client.
   Người dùng sử dụng web browser để truy cập Internet

#### WWW là gì?

#### www

WWW - **World Wide Web** (mạng toàn cầu) thường dùng thay cho Internet nhưng thực ra đây chỉ là một dịch vụ phổ biến nhất trong các dịch vụ mà Internet cung cấp cho người sử dụng như <u>FTP</u>, <u>Mail</u>,....

#### Internet

Internet là một hệ thống gồm nhiều máy tính trên khắp thế giới được kết nối với nhau để người sử dụng máy có thể trao đổi các thông tin như email, chat, file tài liệu, ...

Khái niệm các máy tính kết nối với nhau gọi là **mạng máy tính**. Trong một mạng, các máy tính có thể liên lạc với nhau dựa trên một quy tắc chung gọi là giao thức (mà bạn có thể tạm hiểu là một nghi thức giao tiếp). Internet sử dụng giao thức có tên là **HTTP**: HyperText Transfer Protocol.

# WWW làm việc như thế nào?

Các thông tin được lưu trữ dưới dạng các trang web.

Một trang web là một file được lưu trên máy tính gọi là **web server**. Web server thường ít được nói tới, thay vào đó người ta hay đề cập tới thuật ngữ **website**, là địa chỉ của một web server trên Internet.

Máy tính mà bạn sử dụng để truy cập vào Internet gọi là **web client**.

Khi bạn muốn xem một trang web, web client sẽ làm việc với web server chứa trang web đó, yêu cầu server gửi file chứa trang web về cho mình. Khi client nhận được trang web, nó sẽ hiển thị trang web trên màn hình nhờ vào một chương trình gọi là **web browser**.

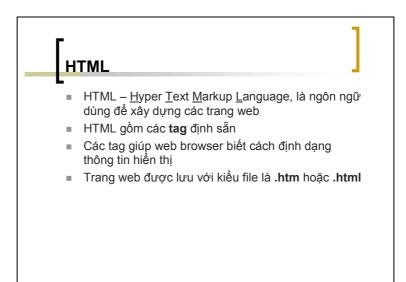
Các web browser phổ biến là: Internet Explorer (IE), Netscape Navigator, Opera,...

## 🖁 Mở rộng:

- Trình bày ứng dụng của FTP và Mail.
- Trình bày các đồ hình mạng và giới thiệu tên gọi một số thiết bị mạng.



# II.2. Các khái niệm về HTML



#### Trang web

Trang web đơn giản là một file văn bản viết theo ngôn ngữ HTML. <u>H</u>yper <u>T</u>ext <u>M</u>arkup <u>L</u>anguage là một ngôn ngữ được sử dụng để giúp web browser biết cách hiển thị và định dạng các dòng chữ, hình ảnh,... trong một trang web.

Ngôn ngữ HTML rất đơn giản, chỉ bao gồm các tag, hiểu đơn giản là những ký hiệu quy định sẵn, mà web browser có thể hiểu được.

Có thể tạo ra trang web bằng HTML sử dụng một trình soạn thảo văn bản đơn giản như Notepad.

#### ⁴ Ví dụ:

<b>Đây là một tag của HTML</b>

<br/> là một tag trong HTML cho biết phần văn bản nằm giữa <b> và </b> phải<br/> được in đậm, khi hiển thị dòng trên, web browser sẽ hiển thị: Đây là một tag<br/> của HTML

Một trang web luôn được lưu với phần mở rộng là .htm hoặc .html.

## ⁴ Thực hành

Sử dụng Notepad để soạn một trang HTML với nội dung dưới đây và lưu với kiểu file là .htm.

```
<hr/>
<html>
<head><title>Trang Web dau tien</title></head>
<BODY>
        HTML, HEAD, TITLE, BODY la nhung <b>Tag</b> co ban
</BODY>
</HTML>
```

Dùng Internet Explorer mở file vừa tạo để xem kết quả.



# III. Cấu trúc của một trang Web

# Cấu trúc của một trang web

- Tag
  - Tag được dùng để quy định cách định dạng thông tin hiển thị trên trang web.
  - o Tag có thể có các thuộc tính.
- Cấu trúc trang web
  - Các thành phần của trang web nằm trong tag <a href="https://html">html></a>
  - Một trang web gồm 2 phần: head và body

#### **HTML Tag**

HTML sử dụng các ký hiệu để định dạng văn bản và hình ảnh. Mỗi ký hiệu được gọi là một tag. Một tag thông thường gồm 2 phần: mở đầu và kết thúc. Ví dụ, <b> và </b>, <html> và </html>,... nhưng thì không cần .

HTML không phân biệt chữ HOA/thường, ví dụ <HTML> và <html> là như nhau. Bạn nên dùng chữ thường khi viết các tag và không để khoảng cách giữa các ký tự trong tag.

#### Thuộc tính và giá trị của thuộc tính

Một tag có thể có nhiều thuộc tính được đặt trong phần mở đầu của tag. Giá trị của các thuộc tính được đặt trong cặp dấu nháy kép (")

Vi du: <body background="red">

#### Cấu trúc một trang web

Nội dung của một trang web được viết bằng ngôn ngữ HTML và luôn đặt trong cặp tag <html></html>.

Trang web gồm 2 phần chính:

- Đầu trang, nằm trong cặp tag <head></head>
- Thân, nằm trong cặp tag <body></body>

Phần head dùng để chứa một số thông tin về trang web khi cần thiết như tiêu đề trang web, nội dung mà web browser hiển thị chỉ nằm trong phần body.

#### Cấu trúc trang web



**Chú ý:** HTML không phân biệt chữ HOA/thường cũng như cách trình bày đoạn mã HTML. Tất cả đều có thể viết trên cùng một dòng hay không cần canh lề.

#### ⁴ Ví dụ:

Trong ví dụ trên, phần in đậm là những giá trị đặc biệt của HTML. Để web browser có thể hiển thị: <head> hay <body>, bạn không thể viết <head> hay <body> trong đoạn code vì web browser sẽ lầm tưởng là một tag. Ký tự < và > là những ký tự đặc biệt mà HTML phải dùng những giá trị thay thế là &gt; và &lt;

Các giá trị đặc biệt trong HTML được bắt đầu bằng & và kế thúc bởi;

gt (greater than) là giá trị đặc biệt của > tương tự như lt (lower than) là giá trị đặc biệt của >.



# IV. Các tag cơ bản trong HTML

# Các tag cơ bản trong HTML

- Xác định cấu trúc trang web
  - o <html>,<head>,<body>
- Định dạng văn bản
  - o ,<br>,<h1>,<h2>,<h3>
- Ghi chú
  - o <!-- ghi chú -->

#### Các tag cấu trúc

HTML bao gồm 3 tag để xác định cấu trúc của trang web bao gồm:

- <htm></html>
- <head></head>
- <body></body>

#### Các tag định dạng văn bản

Mặc dù có rất nhiều tag để định dạng văn bản, những tag sau đây là những tag cơ bản nhất mà gần như bất cứ một trang web nào cũng phải sử dụng:

- : bắt đầu một đoạn văn bản mới
- <br> : xuống dòng
- <h1></h1>, <h2></h2>,...: đặt dòng văn bản nằm trong cặp tag là tiêu đề (heading).

## Tag ghi chú

Cũng như các ngôn ngữ lập trình, để cho phép người viết trang web đặt những ghi chú dành riêng cho mình vào trong trang web, HTML cung cấp tag ghi chú. Đây là tag đặc biệt so với những tag khác:

<!-- nội dung ghi chú -->

## Nghi nhớ tag qua ý nghĩa

HTML 4.0 có tương đối nhiều tag, để nhớ được nhiều, người viết thường phải hiểu được ý nghĩa tên của mỗi tag. Các tag trong HTML thường là viết tắt của những từ gợi nhớ như: Paragraph, BReack,...



# V. Định dạng Text

# Định dạng Text

- Căn bản về định dạng văn bản và cách hiển thị
- Font chữ, màu sắc và canh lề
- Định dạng trước nội dung văn bản
- Quotation, deleted và inserted
- Một số tag định dạng khác

Thông tin trao đổi trên www cũng giống như những tài liệu trong đời thường chủ yếu là những văn bản và hình ảnh. Để có được một trang web được trình bày rõ ràng, đẹp và dễ hiểu bạn cần biết cách định dạng các đoạn văn bản.

Phần này trình bày tất cả các kỹ thuật định dạng văn bản trên trang web. Một điểm bạn nên ghi nhớ ngay từ đầu là việc định dạng văn bản và cách mà văn bản được hiển thị trên các trình duyệt là độc lập với nhau. Mỗi trình duyệt web có thể hiểu và hiển thị văn bản bạn đã định dạng theo cách riêng của mình.



# V.1. Căn bản về định dạng văn bản và cách hiển thị

#### Căn bản về định dạng văn bản và cách hiển thị

- Định dạng kiểu chữ
  - o <b>,<u>,<i>,<strong>,<em>
  - <small>,<big>
  - o <sup>,<sub>
- Web browser hiển thị trang web
  - Mỗi web browser có cách hiển thị khác nhau cho một định dạng của HTML

#### Các kiểu chữ

Trong các tài liệu, văn bản chúng ta thường sử dụng các kiểu chữ **đậm**, *nghiêng*, gạch dưới,...ví dụ sau minh hoạ các tag được dùng định dạng kiểu chữ:

```
<html>
<body>
<b>In dâm</b> <br>
<strong>In rất dâm </strong><br>
<font size="4">c</font><big>hữ lớn </big><br>
<em>nhấn mạnh</em><br>
<i>iin nghiêng </i><br>
<small>chữ nhỏ</small><br>
Công thức hoá học của nước: H<sub>2</sub>O<br>
X bình phương: X<sup> 2 </sup>
</body>
</html>
```

#### Kết quả:

In đâm

In rất đậm
Chữ lớn
nhấn mạnh
in nghiêng
chữ nhỏ
Công thức hoá học của nước: H<sub>2</sub>O
X bình phương: X<sup>2</sup>

Học phần 5 – Lập Trình Web Cơ Bản



#### Web browser hiển thị nội dung trang web như thế nào?

Kết quả của ví dụ trên được hiển thị trong Internet Explorer. Một web browser khác, chẳng hạn Netscape Navigator có thể sẽ cho một cách trình bày nội dung văn bản khác.

Với mỗi tag, các web browser khác nhau có thể có cách định dạng nội dung khác nhau. Điều này cũng tương tự như việc bạn có một cuộn phim được chụp từ một máy ảnh nhưng khi đem rửa trên mỗi loại giấy ảnh khác nhau có thể cho hình ảnh khác nhau.

Việc mỗi web browser hiển thị trang web theo một cách khác nhau làm một trong những vấn đề phức tạp nhất đối với người xây dựng trang web. Cách giải quyết thông thường nhất là ứng với mỗi loại web browser, ta xây dựng một trang web có cùng nội dung nhưng cách trình bày phù hợp.

#### H Thực hành

- Có nhiều cách để viết trang html, tuy nhiên, khi mới thực hành bạn nên sử dụng notepad.exe hay Microsoft Front Page (trong bộ MS Office)
- Với Front Page, bạn có thể vừa viết code HTML, vừa xem kết quả hiển thị bằng cách chuyển qua lại giữa 2 tab HTML và Preview. Với notepad, mỗi khi sửa đổi nội dung file html bạn sẽ phải lưu lại file, chuyển qua web browser (Internet Explorer IE) và chọn Refresh
- Để xem code HTML của một trang web đã có từ IE, trên menu View, chọn mục Source. Bạn có thể học hỏi được nhiều điều bằng cách xem code HTML của những trang web được thiết kế chuyên nghiệp nhưng hãy nhớ rằng những trang web đẹp luôn được viết rất công phu và thường sử dụng nhiều công cụ (tool) hỗ trợ.



# V.2. Font chữ, màu sắc và canh lề

# Font chữ, màu sắc và canh lề

- <font face="..." size="..." color="#HHHHHH">...</font>

#### ⁴ Ví dụ:

#### Kết quả:

#### COMPUTER JOKE

Kỹ thuật viên: Máy tính của anh có ổ đĩa mềm chứ?

Khách: Tôi không nhìn thấy bên trong. Có chữ "Intel Pentium Inside"

## Thuộc tính của một tag

Một thông tin định dạng có thể gồm nhiều chi tiết, trong ví dụ trên, font chữ sẽ hiển thị cho một chuỗi văn bản được chỉ định qua tag <font> tuy nhiên, font chữ lại gồm nhiều chi tiết như: tên font, kích thước, màu sắc,...

Các thông tin chi tiết được gọi là các thuộc tính của tag. Một tag có thể có nhiều thuộc tính. Bạn nên đặt giá trị của thuộc tính trong dấu ngoặc kép.

Trang web có thể trình bày được bao nhiêu màu?

Giá trị của color là một số gồm 6 chữ số hexa, mỗi cặp 2 số dành cho các màu Đỏ (R), Xanh lá (G), Xanh dương (B). Có thể có  $256 \times 256 \times 256 = hơn 16.7$  triệu màu.



## V.3. Định dạng trước nội dung văn bản

#### Định dạng trước nội dung văn bản

- Web browser bo qua các khoảng trắng trong code HTML
- Tag giúp người viết nhìn thấy trước các web browser sẽ hiển thị văn bản trên trang web

Web browser sẽ không quan tâm đến các bạn trình bày đoạn code HTML trong file .html mà chỉ dựa vào các tag để trình bày nội dung trang web.

Ví dụ, trong file .html, bạn có thể viết 5 hay 10 khoảng trắng liên tiếp thì web browser vẫn coi như 1 khoảng trắng. Bạn có thể viết toàn bộ đoạn code trên một dòng duy nhất hay chia nhỏ trên nhiều dòng khác nhau, web browser vẫn sẽ trình bày nội dung trên một dòng trừ khi bạn có tag hay <br/> (hay một số tag khác mà HTML quy định là web browser phải xuống dòng khi trình bày.)

Tag được dùng khi bạn muốn yêu cầu web browser "tôn trọng" các khoảng trắng và xuống dòng trong đoạn code HTML của mình.

#### ⁴ Ví dụ:

```
Không
    có
    tag
    <pre&gt;

    Doạn văn bản này
    nằm trong
    tag &lt;pre&gt;
```

#### Kết quả:

```
Không có tag 
Doạn văn bản này
nằm trong
tag 
tag 
tag 
tag 
tag 
tag 
tag 
tag
```

Bạn cũng nên chú ý rằng các văn bản nằm trong tag <pre> được hiển thị theo font chữ riêng.



## V.4. Quotation, deleted và inserted

#### Quotation, deleted và inserted

- Trích dẫn: <blockquote>
- Văn bản bị xoá: <del>
- Văn bản mới được thêm vào: <ins>

Trong các văn bản, đôi khi bạn muốn trích dẫn các câu nói, định dạng một vài từ giống như đã được viết ra rồi bị xoá bỏ hay một vài từ khi viết đã bỏ sót rồi mới được thêm vào. HTML cung cấp một số tag mô tả cách trình bày văn bản với ý nghĩa thể hiện quá trình soạn thảo văn bản của tác giả.

Bạn phải luôn ghi nhớ rằng cách trình bày văn bản theo định dạng mà mỗi tag quy định là do web browser quy định. Ví dụ như trong trường hợp tag <ins> (inserted), đa số web browser trình bày bằng cách gạch dưới chuỗi văn bản, giống như tag <u>.

#### √ Ví dụ:

Lenin:<blockquote>Hoc, <ins>hoc nữa</ins>, <del>hoc mãi</del></blockquote>

#### Kết quả:

Lenin:

Học, học nữa, học mãi

Điểm đáng chú ý khi dùng <ins> và <del> là ý nghĩa của việc sử dụng các tag này. Nếu bạn thử copy đoạn văn bản trong web browser để paste vào trình soạn thảo văn bản, bạn sẽ thấy phần inserted không bị gạch dưới còn phần deleted thì bị bỏ qua. Trong ví dụ trên, khi paste, chữ học mãi sẽ không có.

Chú ý rằng với một số tag, web browser sẽ xuống dòng và canh lề giống như với tag <br/> <br/> canh lề giống như với tag <br/> canh lề giống như với



# V.5. Một số tag định dạng khác

# Một số tag định dạng khác

- Định dạng chuẩn của máy tính
  - o <code>, <kbd>, <tt>, <samp>,<var>
- Đia chỉ
  - o <addr>
- Viết chữ từ phải sang trái
  - o <bdo dir="rtl">
- Viết tắt
  - <abbr>, <acronym>. Tool tip sẽ hiển thị nội dung chữ viết tắt khi người đọc đặt mouse trên từ viết tắt.

#### Các định dạng chuẩn của máy tính

<code>Computer code</code><br>
<kbd>Keyboard input</kbd><br>
<tt>Teletype text</tt><br>
<samp>Sample text</samp><br>
<var>Computer variable</var>

#### Định dạng địa chỉ

<address>Donald Duck<br>
BOX 555<br>Disneyland<br>
USA</address>

#### Viết chữ từ phải sang trái

<bdo dir="rtl">
Tu phai sang trai
</bdo>

#### Viết tắt

<abbr title="United Nations">UN </abbr><br> <acronym title="World Wide Web">WWW</acronym>

V.5.1. Computer code Keyboard input Teletype text Sample text Computer variable

Donald Duck BOX 555 Disneyland USA

V.5.2. iart gnas iahp uT

V.5.3. UN

V.5.4. WWW



# VI. Liên kết các trang web

# Liên kết các trang web Mối liên kết và địa chỉ liên kết Tạo liên kết trong trang web

Trang web không chỉ giới hạn nội dung muốn truyền đạt đến cho người đọc những thông tin trong bản thân nó mà còn có thể mở rộng hơn cho người đọc những thông tin từ những trang web khác. Các trang web có thể liên kết với nhau dựa vào các mối liên kết.



## VI.1. Mối liên kết và địa chỉ liên kết

# Mối liên kết và địa chỉ liên kết

- Hyperlink
  - Được đặt trong trang web để liên kết tới một trang web khác
- URL
  - Đường dẫn chỉ tới một trang web được liên kết trên Internet.
  - Địa chỉ website chỉ tới thư mục gốc chứa các trang web trên webserver
  - URL là đường dẫn tương tự như đường dẫn của DOS nhưng sử dụng dấu / thay vì \

#### **Hyperlink**

Những phiên bản HTML đầu tiên chỉ đơn giản trình bày các trang tài liệu văn bản và các mối liên kết giữa các trang. Những mối liên kết này gọi là **hyperlink**.

Với hyperlink, HTML trở thành công cụ hiệu quả để chuyển tải thông tin tới cho người đọc. Bạn có thể đang đọc một trang web này rồi đi theo một liên kết đến một trang web khác sau đó quay lại trang cũ ngay tại nơi đã dừng lại.

#### **URL**

Universal Resource Locator, là một đường dẫn được dùng trên Internet để chỉ tới một trang web cụ thể nào đó. Thuật ngữ thường dùng thay cho url là : "địa chỉ"

Ví dụ: http://www.yahoo.com là địa chỉ tới trang chủ của Yahoo!

Trong ví dụ trên, http: là tên giao thức mà bạn dùng để truy cập tới trang web. // mang ý nghĩa đường dẫn chỉ ra sau đó bắt đầu từ gốc. www.yahoo.com gọi là domain name (tên miền).

Domain name là tên dễ nhớ của một địa chỉ. Những tên này được quản lý bởi một tổ chức quốc tế, đảm bảo không có hai địa chỉ khác nhau nào có cùng tên. Nếu bạn muốn website của mình có một tên gợi nhớ để mọi người có thể truy cập, bạn sẽ phải đem tên đó đi đăng ký.

Trong domain name, phần cuối cùng dùng để phân loại các website:

- Com : commercial website thương mại, kinh doanh
- Edu: education website về giáo duc, đào tao
- Gov : government website của chính phủ
- Vn, uk, au, ...: vietnam, united kingdom, austratlia website của quốc gia nào.
- **.**...



#### URL và cấu trúc thư mục lưu trữ trang web trên Web Server

Trên webserver, tất cả các trang web phải được tổ chức lưu trong một thư mục gốc, ví dụ: **wwwroot**. Trong wwwroot, bạn có thể tổ chức cấu trúc thư mục con tuỳ ý, tuy nhiên, các trang web nằm ngoài wwwroot thì người dùng trên Internet sẽ không thể truy cập được.

Domain name là địa chỉ tới thư mục wwwroot của bạn. Thực ra, để hiển thị trang web, web server đã quy định sẵn một trang mặc định - thường là **index.html** hay **default.html**. Chỉ cần web client đưa địa chỉ tới thư mục wwwroot thì index.html hay default.html sẽ được trả về cho client.

Để chỉ ra một trang web nằm trong thư mục con của wwwroot, bạn sẽ phải chỉ ra tên thư mục và tên file chứa trang web.

<u>Ví du</u>: Bạn có một website với địa chỉ: <u>www.myweb.com</u>. File gioithieu.html nằm trong thư mục AboutMe. URL chỉ tới file gioithieu.html của bạn sẽ là: <u>www.myweb.com/AboutMe/gioithieu.html</u>

URL có quy ước thư mục tương tự như đường dẫn trên DOS, tuy nhiên ta dùng dấu / thay vì \. Cạn cũng có thể dùng đường dẫn tương đối khi liên kết hai trang web trong cùng website của mình. Đường dẫn tương đối không cần chỉ ra domain name và có thể dùng đấu .. để đi ra ngoài một thư mục.

<u>Ví du</u>: Trong wwwroot, bạn có thư mục Data chứa file *lichbieu.html*. trong file này bạn muốn có một liên kết tới gioithieu.html thì URL sẽ dùng là /AboutMe/gioithieu.html hoặc ../AboutMe/gioithieu.html.



#### VI.2. Tạo liên kết trong trang web

# Tạo liên kết trong trang web

- Tag <a> dùng để tạo các mối liên kết
  - o href: Địa chỉ liên kết
  - o target: Cửa sổ hiển thị nội dung
- <a href="..." target="...">
- <a name="#...">
- <a href="mailto:support@yahoo.com">

HTML dùng tag <a> (anchor) để tạo liên kết tới một trang web. Tag <a> có ba thuộc tính chính là:

href : địa chỉ của trang web muốn liên kết

■ target : cửa sổ sẽ hiển thị trang web

■ name : tên của mối liên kết

Ví du:

<a href="http://www.yahoo.com">Liên kết tới Yahoo!</a>

#### Kết quả:

#### Liên kết tới yahoo

Thuộc tính target chỉ ra cửa sổ sẽ dùng để mở trang web mới. Nếu không đặt giá trị cho target, trang web bạn đang xem sẽ bị thay thế bằng trang web mới.

Để mở trang web trong một cửa sổ mới, đặt target="\_blank"

Target chủ yếu được dùng trong các trang web có nhiều frame, bạn sẽ học cách sử dụng frame ở phần sau.

## Liên kết trong cùng trang web

Nếu như cho bạn được quyền đặt tên cho các tag của HTML, có lẽ bạn sẽ thay <a> bằng <l> (Link) thì đúng hơn. Tuy nhiên <a> thực sự mang ý nghĩa là một mỏ neo (anchor) khi bạn dùng để liên kết tới một đoạn văn bản nào đó trong chính bản thân trang web.

Thuộc tính name của <a> dùng để đặt tên cho đoạn văn bản sẽ liên kết tới. Chú ý, giá trị của name có dấu # đứng trước.

#### ൌ Ví dụ:

```
<a href="#EndOfPage">Đến cuối trang</a>
......
<!-- nhiều dòng -->
<a name="#EndOfPage">cuối trang</a>
```

## Liên kết với địa chỉ email

để liên người đọc gửi mail cho bạn bằng cách click vào liên kết, gán giá trị "mailto:địa chỉ email" cho thuộc tính href.



## VII. Danh sách

# Danh sách (List)

Danh sách gồm 2 loại: có thứ tự và không có thứ tự

<Ii>HTML

JavaScript

Active Server Page (ASP)

</n/>

- Các tag phải nằm trong một tag danh sách: , , <menu>, <dir>
- Thuộc tính type cho phép thay đổi cách đánh số thứ tự hay bullet bắt đầu mỗi mục trong danh sách

Danh sách trong HTML tương tư như bạn định dạng Bullets and Numbering trong Word.

Thông thường, chúng ta ít phân biệt giữa danh sách có thứ tự và không có thứ tự. Với danh sách có thứ tự, mỗi mục sẽ được đánh thứ tự 1, 2, 3 hay a, b, c, ... trong khi với danh sách không có thứ tự, mỗi mục sẽ bắt đầu bằng dấu −, •, •, ○,...

Trong HTML, mỗi mục trong danh sách được bắt đầu bằng tag Lác mục trong danh sách lại được đặt trong một tag danh sách. HTML có các tag danh sách:

- : order list danh sách có thứ tự
- : unorder list danh sách không có thứ tự
- <menu> : danh sách dạng menu
- <dir>: directory danh sách thư mục

#### ⁴ Ví dụ:

#### Kết quả:

Nội dung môn học lập trình web cơ bản

- 1. HTML
- 2. JavaScript
- 3. Active Server Page (ASP)



#### ⁴ Ví dụ:

```
Nội dung môn học lập trình web cơ bản 
    HTML 
    JavaScript 
    Active Server Page (ASP)
```

#### Kết quả:

Nội dung môn học lập trình web cơ bản

- HTML
- JavaScript
- Active Server Page (ASP)

#### The vi dụ: Các danh sách có thể lồng nhau

```
Nội dung môn học lập trình web cơ bản

    HTML 
    Cơ bản về HTML 
    HTML nâng cao 
    JavaScript 
    JavaScript 
    Cól>
    Hi>Karaga cơ sở 
    Các đối tượng cơ sở 
    Hi>Các Val>
```

#### Kết quả:

Nội dung môn học lập trình web cơ bản

- HTML
  - 1. Cơ bản về HTML
  - 2. HTML nâng cao
- JavaScript
  - 3. Ngôn ngữ JavaScript
  - 4. Các đối tượng cơ sở
- Active Server Page (ASP)

<menu> và <dir> có định dạng tương tự như do đó, chỉ quan trong về mặt ý nghĩa của nội dung văn bản đối với người viết còn đối với người đọc là hoàn toàn tương tự nhau.

Thuộc tính type của các tag danh sách cho phép bạn định lại các số thứ tự hay bullet hiển thị đầu mỗi mục trong danh sách. Các giá trị của type:



<ol> - Order list</ol>	<ul><li>ul&gt; - Unorder list</li></ul>
"A": A, B, C, "a": a, b, c, "I": I, II, III, "i": i, ii, iii, "1": 1, 2, 3, (măc đinh)	"disk": ● "circle": O "square": ■

#### ⁴ Ví dụ:

```
Nội dung môn học lập trình web cơ bản

  HTML 

  JavaScript 
  Active Server Page (ASP)
```

#### Kết quả

Nội dung môn học lập trình web cơ bản

- HTML
- JavaScript
- Active Server Page (ASP)



# VIII. Hình ảnh



Hình ảnh là một dạng media được đưa vào trang web theo dạng liên kết. Hiểu được cách các trình duyệt thể hiện hình ảnh trong trang web bạn cũng sẽ nắm được cơ chế để "xuất bản" các tài liệu media khác như Word document, PDF document, Mpeg,... lên Internet.



## VIII.1. File hình ảnh và trang web

# File hình ảnh và trang web

- Hình ảnh và các thông tin phi văn bản đều không được lưu chung vào trang web mà như những file độc lập
- Thông tin phi văn bản muốn hiển thị chính xác phải có sự hỗ trợ của web browser
- <img src="URL chỉ ra file hình ảnh">

HTML những phiên bản đầu tiên không hỗ trợ tác giả đưa hình ảnh vào các trang web. HTML giờ đây đã cho phép bạn đưa vào trang web không chỉ hình ảnh mà cả các file "nhúng" như video, âm thanh,... Tuy nhiên, khi đưa những hình ảnh, video hay âm thanh vào trang web, bạn phải cân nhắc một số điểm sau:

- Hình ảnh không được lưu chung với nội dung trang web
- File hình ảnh thường có kích thước lớn hơn nhiều so với file .html và làm cho trang web hiển thị chậm.
- Hình ảnh hiển thị trên máy tính của bạn có thể khác với trên máy tính của người khác
- Nên sử dụng các định dạng file thông dụng mà web browser hỗ trợ như GIF, JPEG, BMP, PNG



## VIII.2. Đưa hình ảnh vào trang web

# Đưa hình ảnh vào trong trang web

- <img src="URL chỉ ra file hình ảnh">
- Chú thích và thay đổi kích thước cho hình ảnh
  - o atl: chú thích cho khi không hiển thị được
  - o width, heigh : định lại kích thước cho hình
- Vị trí tương đối giữa hình ảnh và dòng văn bản
  - o align = "top/middle/bottom"
  - o algin = "left/right"

HTML sử dụng tag <img> (image) để hiển thị hình ảnh. Thuộc tính quan trọng nhất của <img> là src (source) có giá trị là một URL chỉ ra đường dần tới file hình ảnh muốn hiển thị.

Ví dụ: Yahoo! <img src="yahoo.gif">

#### Kết quả:

Yahoo!



**Chú ý:** Nếu file hình ảnh của bạn không nằm chung thư mục với file .html thì bạn phải chỉ ra đường dẫn tới file đó.

## Thuộc tính atl

Trong các trường hợp hình ảnh không được hiển thị do không tìm thấy file hay web browser không nhận được file từ phía web server, có thể sử dụng thuộc tính alt (alternate) với giá trị là nội dung mô tả tóm tắt hình ảnh muốn thể hiện để người dùng dễ nhận biết.

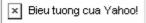
#### ⁴ Ví du:

Yahoo!

<img src="yahoo.gif" alt="Bieu tuong cua Yahoo!">

#### Kết quả: (khi không có file c:\yahoo.gif)

Yahoo!



## Các thuộc tính xác định chiều rộng và chiều cao

Để thay đổi chiều rộng và chiều cao của hình ảnh, sử dụng hai thuộc tính width và height. Giá trị của width và height thường dùng là pixel (mặc định) và %.

#### ⁴ Ví du:

Yahoo!

<img src="c:\yahoo.gif" width="30" height="30">



#### Kết quả:



## Định dạng hình ảnh và dòng văn bản

Mặc định, khi bạn đặt hình ảnh trên một dòng văn bản thì dòng văn bản sẽ nằm ngang với cạnh dưới của hình. Thuộc tính align cho phép thay đổi vị trí tương đối của dòng văn bản so với hình ảnh. Các giá trị thường dùng của align là: bottom (cạnh dưới), middle (giữ), top (cạnh trên)

#### ⁴ Ví dụ:

Yahoo!

<img src="yahoo.gif" width="100" height="100" align="middle">

#### Kết quả:

Yahoo!



#### Hình ảnh Floating

Hình ảnh khi đặt vào trang web trên một dòng văn bản ở dạng inline, chiều cao của dòng văn sẽ tự động giãn ra bằng chiều cao của hình. Ở dạng floating, hình ảnh có thể nằm trên nhiều dòng và văn bản sẽ tự động bao quanh khung chữ nhật có chứa hình.

Để đặt hình ảnh ở dạng floating, đặt giá trị cho thuộc tính align là right hoặc left.

#### ⁴ Ví dụ:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diem nonummy nibh euismod tincidunt ut lacreet dolore magna aliguam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper



suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis te feugifacilisi. Duis autem dolor in hendrerit in vulputate velit esse molestie



# Bài 2

# XÂY DỰNG CẤU TRÚC TRANG WEB

# Tóm tắt

Lý thuyết 2 tiết - Thực hành 2 tiết

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học này tiếp theo bài học trước trình bày cách sử dụng các tag HTML để tạo bảng. Tiếp theo là phần trình bày cách sử dụng các frame để hiển thị nhiều trang web trong một cửa sổ trình duyệt. Phần cuối cùng của bài học muốn giới thiệu cho học viên khái niệm về việc tổ chức bố của một website.  Sau khi hoàn tất bài học này, học viên có thể:  Mô tả cấu trúc của một table trên trang web  Sử dụng các tag HTML để tạo table và frame  Trình bày khái niệm về bố cục của một website  Sử dụng Table hay Frame để tạo bố cục cho một website	I. Giới thiệu bài học II. Bảng (Table) III. FrameSet và Frame IV. Bố cục của website	2.1 – 2.6	



# I. Giới thiệu bài học



Table và frame là hai thành phần thiết kế trang web rất quan trọng mà hầu như bất cứng lập trình viên ứng dụng web nào cũng phải thành thạo. Ngoài việc được dùng để trình bày các thông tin dạng bảng, table và frame được dùng bố trí các thành phần trong trang web giúp tiết kiệm các khoảng trống, đặt thêm nhiều thông tin và lập ra một bố cục rõ ràng cho trang web.



# II. Bảng (Table)

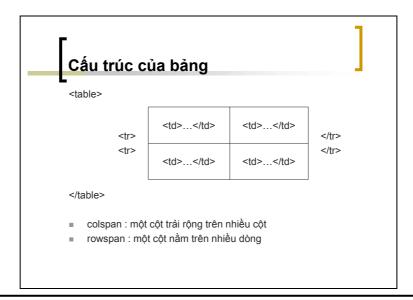


Khác với bảng được tạo ra bằng Word hay Excel dựa trên cấu trúc **dòng** và **cột**, table trong trang web dựa trên cấu trúc dòng và các cell (ô trên một dòng). Ngoài việc sử dụng table để trình bày dữ liệu theo dạng bảng, table còng dùng để bố trí các khu vực hiển thị dữ liệu trong trang web. Khi đó, table được đặt ở dạng ẩn và chỉ nhìn thấy lục thiết kế.

Phần này trình bày với bạn cấu trúc của table và cách sử dụng các thuộc tính của nó để trình bày dữ liệu dạng bảng. Sử dụng table để tạo bố cục cho trang web sẽ được nói tới ở cuối bài và bạn sẽ thực hiện việc phân vùng hiển thị trên trang web trong bài tập xây dựng form ở bài sau.



### II.1. Cấu trúc của bảng



HTML sử dụng bộ một cấu trúc tag gồm có <table>, <tr> và <td> để định dạng các bảng:

- : phần nằm trong tag là một cấu trúc các dòng và cột của bảng
- table row : phần nằm trong tag là cấu trúc các cột của một dòng
- table data : phần nằm trong tag là nội dùng của một cell (một cột của một dòng)

#### ⁴ Ví dụ:

#### Kết quả:

Cột 1 dòng 1	Cột 2 dòng 1	Cột 3 dòng 1
Cột 1 dòng 2	Cột 2 dòng 2	Cột 3 dòng 2

#### Chú ý:

- Đặt thuộc tính boder cho tag để bảng có đường kẻ khung
- Mỗi cột có độ rộng đủ để chứa nội dung bên trong nếu không đặt giá trị cho thuộc tính width của

Web browser luôn vẽ table với các dòng có số cột bằng nhau và độ rộng mỗi cột trên các dòng bằng nhau.



#### ⁴ Ví dụ:

#### Kết quả:

Cột 1 dòng 1	Cột 2 dòng 1	
Cột 1 dòng 2	Cột 2 dòng 2	Cột 3 dòng 2

Để một cột 2 dòng 1 có chiều rộng bằng phần còn lại trên dòng (trải rộng trên 2 cột), bạn nhờ đến thuộc tính colspan của . Đặt colspan="2", ta sẽ có kết quả hiện trên web browser như sau:

Cột 1 dòng 1	Cột 2 dòng 1	
Cột 1 dòng 2	Cột 2 dòng 2	Cột 3 dòng 2

Tương tự colspan, ví dụ sau sử dụng thuộc tính rowspan, chắc bạn cũng đoán ra rowspan="2" được đặt trong tag nào rồi!

Cột 1 dòng 1	Cột 2 dòng 1	
Cột I dong I	Cột 2 dòng 2	Cột 3 dòng 2



### II.2. Sử dụng các thuộc tính của bảng

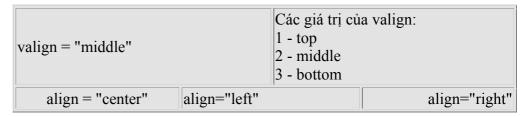
### Sử dụng các thuộc tính của bảng

- Width, CellSpacing và CellPadding
  - o width : chỉ định độ rộng của table hay các cột
  - o cellspacing : chỉ định khoảng cách giữa các cell
  - cellpadding: chỉ định khoảng cách từ biên của cell tới nội dung trong cell
- Align, Caption, Border, BGColor
  - border, bgcolor : độ dày đường kẻ và màu nền của bảng
  - o valign, align : canh lè cho văn bản trong cell

#### ⁴ Ví du:

#### Kết quả:

Align, Caption, Border, BGColor



<caption> dùng để đặt tiêu đề cho một table. Thuộc tính border của dùng để xác định độ dày của đường kẻ bảng.

Bgcolor cho phép chỉ ra màu nền của bảng, bạn cũng có thể dùng bgcolor để chọn màu nền cho một cell hay toàn bộ trang web. Nếu bạn đã từng xem những trang web thay vì sử dụng những màu nền đơn điệu là các hình ảnh nền rất trang nhã và đẹp, bạn có thể thay bgcolor bằng background với giá trị tương tự như thuộc tính scr của <image>.

Ví dụ trên cũng trình bày cách sử dụng thuộc tính valign và align trong table:

- valign vertical align : canh lề trong cell theo chiều dọc
- align horizontal align : canh lè trong cell theo chièu ngang.



# III. FrameSet và Frame

### FrameSet và Frame

- Hiển thị nhiều trang web trên web browser
- Sử dụng FrameSet và Frame
- Frame "tĩnh" và frame "động"

Đôi khi, việc hiển thị một trang web trong toàn bộ màn hình không được hiệu quả. Trong trường hợp các trang web tương tự như các trang trong văn bản có header và footer giống nhau, bạn có thể nghĩ tới việc thiết kế và duy trì sự đồng nhất của header và footer giữa các trang là điều nhàm chán và tốn thời gian. Bằng cách sử dụng frameset để xác định cấu trúc một trang tài liệu và dùng các frame để thể hiện riêng lẻ từng phần header, nội dung và footer bạn có được một cách giải quyết dễ dàng hơn.

Sau phần này, bạn sẽ hiểu được cấu trúc của Frameset và cách sử dụng các tag <Frameset>, <frame> để tạo nên một bộ khung các trang web trong cửa sổ web browser.



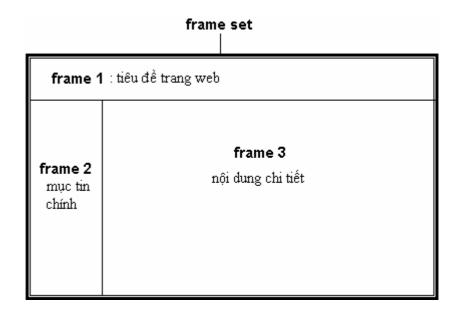
### III.1. Hiển thị nhiều trang trên web browser

#### Hiển thị nhiều trang web trên web browser

- Web browser hiển thị nhiều trang web trên cùng một cửa sổ dựa trên các frame.
- Mỗi frame hiển thị một trang web độc lập và có thể cập nhật lại nội dung từ web server mà không ảnh hưởng các frame khác

Các web browser phổ biến hỗ trợ việc trình bày các trang web với các frame. Frame là một "khung hình", được dùng để chia nhỏ một cửa sổ hiển thị của web browser thành nhiều phần độc lập với nhau. Mỗi frame có thể hiển thị một trang web độc lập.

Việc nhiều trang web cùng hiển thị trong một cửa sổ web browser tại một thời điểm giúp mang lại nhiều thông tin hơn cho người dùng, đồng thời, khi một phần thông tin nào đó cần cập nhật, chỉ frame chứa trang web đó là cần phải cập nhật lại mà thôi.



Đây là một trong các cấu trúc frame mà bạn thường thấy trên Internet. Trang web thường xuyên được cập nhật là trang web chứa trong frame 3. Frame 1 thường chứa tiêu đề (banner) về web site và rất ít khi phải cập nhật lại. Frame 2 là nơi trình bày một phần cấu trúc của website, giúp bạn dễ dàng tìm ra những thông tin mà mình quan tâm trên web site.



### III.2. Sử dụng Frameset và Frame

### <mark>Sử dụng FrameSet và Frame</mark>

- FrameSet
  - o Là khung chứa các frame
  - Frameset được chia thành các frame nhỏ theo hàng hoặc cột
- Frame
  - Dùng để hiển thị các trang web chỉ ra trong thuộc tính src.
  - Thuộc tính name xác định tên của frame trong frameset.

Frame set hay một bộ các frame là một trang web chỉ có các thông tin về cách tổ chức các frame bên trong cửa sổ của web browser.

#### ⁴ Ví dụ:

```
<frameset rows="108,*">
  <frame name="banner" scrolling="no" noresize
target="contents" src="banner.html">
  <frameset cols="150,*">
        <frame name="contents" target="main" src="news.html">
        <frame name="menu" src="menu.html" target="contents">
        </frameset>
</frameset></frameset></frameset>
```

### Thuộc tính rows và cols của frameset

Một frameset chỉ có thể được chia thành nhiều frame con theo chiều dọc (cols) hoặc ngang (rows). Giá trị của rows hay cols chỉ ra kích thước của mỗi frame. Trong ví dụ trên, frameset lớn nhất được chia thành 2 frame theo chiều ngang

- Frame 1: tên là "banner" và cao 108 pixel.
- Frame 2: là một frameset khác, có chiều cao là phần còn lại trong cửa sổ của web browser (đấu \*).

#### 3 thuộc tính quan trọng của <frame>

- **src** source : đường dẫn tới file .html sẽ hiển thi trong frame
- target : tên của frame sẽ dùng để hiển thị trang web khi người dùng chọn một liên kết trong frame chứa thuộc tính target. Trong ví dụ trên, bạn thấy cả hai frame banner và menu đều muốn các liên kết được hiển thị trong frame contents.
- name : tên của frame, được các frame khác sử dụng khi đặt giá trị cho thuộc tính target.



### III.3. Frame "tĩnh" và frame "động"

#### Frame "tĩnh" và frame "động"

- Frame tĩnh và động giúp tổ chức trang web theo một bố cục hợp lí
- Thuộc tính scrolling định cách hiển thị các thanh cuộn trên frame
- Thuộc tính noresize không cho phép người dùng thay đổi kích thước của frame

Nếu bạn đang thiết kế trang web cho công ty của mình, có lẽ bạn sẽ thích một cách tổ chức các frame như sau:

- Top: logo của công ty và các sản phẩm hay dịch vụ quan trọng
- Bottom : địa chỉ liên lạc của công ty, ngày tháng trang web được cập nhật
- Left : một số liên kết quan trọng như sản phẩm/dịch vụ, đối tác, nhân sự, tuyển dụng,...
- Main : frame chính để hiển thị nội dung

Cấu trúc trang web như trên khá phổ biến và bạn cũng thấy ngay rằng top, bottom, left rất ít khi cần cập nhật thông tin. Top, bottom và left có thể gọi là các frame tĩnh trong khi main là một frame "động".

### Scrolling và NoResize

Có thể coi frame như một cửa sổ, bạn có thể thay đổi kích thước hay đặt các thành scroll bar trong trường hợp frame nhỏ và không hiểnt hị hết nội dung.

Các frame tĩnh nên được thiết kế với kích thước phù hợp để hiển thị được nội dung bạn mong muốn mà không cần thay đổi kích thước frame hay kéo các thanh cuộn. Đồng thời, bạn cũng sẽ đặt các thuộc tính của frame để người dùng không thể thay đổi kích thước và làm cho trang web của bạn bị "biến dạng".

- noresize: không cho phép người dùng thay đổi chiều rộng hoặc chiều cao của frame
- scrolling: "no/yes/auto" tương ứng với frame sẽ không có thanh scroll bar, luôn hiển thị scroll bar hay chỉ hiển thị khi nội dung trang web dài hơn phần màn hình của frame.

Ngoài ra, bạn cũng có thể đặt thuộc tính *border* để ngăn web browser vẽ các đường phân cách giữa các frame.



# IV. Bố cục của website

# Bố cục của website

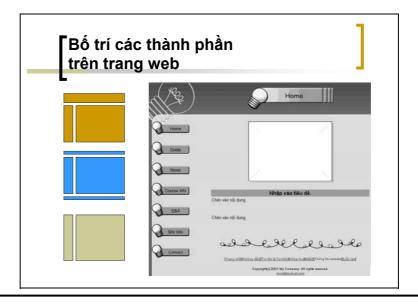
- Bố trí các thành phần trên trang web
- Xây dựng bố cục cho website

Một website khác với một trang web trước hết ở chỗ nó gồm nhiều trang web với nội dung khác nhau, cung cấp nhiều loại thông tin cho người dùng. Tuy nhiên, các trang web đều chia sẽ một hình thức giao diện chung và có thể phối hợp với nhau để cung cấp một chức năng nào đó như xem hàng, đặt hàng, điền thông tin mua hàng,... cho người dùng.

Website cần có một bố cục thống nhất cũng tương tự như hệ thống giao diện MDI hay SDI trong ứng dụng Windows để người dùng có thể dễ dàng sử dụng. Phần này trình bày với bạn cách bố cục một website sử dụng table hay frame đồng thời cũng đề cập đến những điểm khác biệt trong hai cách xây dựng bố cục này.



### IV.1. Bố trí các thành phần trên trang web



#### Bố cục của website

Một Website kết hợp nhiều trang web với các thông tin liên quan với nhau. Một website có thể chỉ liệt kê thông tin bằng những trang .html tĩnh nhưng cũng có thể là một ứng dụng web sử dụng để tra cứu, cung cấp dịch vụ mail, bán hàng, forum thảo luận,...

Tương tự như một ứng dụng bình thường trên Windows gồm các màn hình, menu, toolbar,... một website cũng cần được tổ chức theo một bố cục khoa học, thân thiện và tiện dụng với người dùng.

### Các thành phần trong bố cục của website

Bố cục của website bao gồm các thành phần được bố trí gọn trong một cửa sổ của trình duyệt. Các thành phần thường có trong bố cục là: tiêu đề, trang web chính và menu.

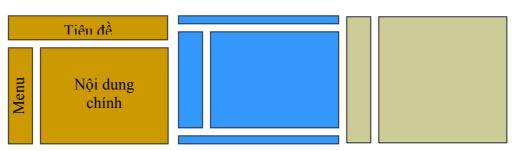
Tiêu đề thường được đặt ở trên cùng của cửa sổ và dùng để hiển thị tên của Website. Một phần màn hình lớn được dùng để hiển thị nội dung của trang web và phần còn lại dùng để đặt phần menu.

Menu trong một website có nhiều ý nghĩa. Menu có thể là danh sách các trang web khác hay danh sách các chức năng mà người dùng có thể sử dụng. Các chức năng như đăng nhập, tìm kiếm,... thường đặt trong phần menu.

### Các bố cục phổ biến

Để đặt các thành phần chính vào trong một cửa sổ, bạn thường phân chia trang web thành một table hay dùng một trang web có nhiều frame, mỗi frame là một trang chứa thành phần muốn hiển thị.

#### **IV.2.**





#### Xây dựng bố cục cho website

### Xây dựng bố cục cho website

- Web layout có thể xây dựng dựa trên FrameSet hay Table
- Lợi điểm của FrameSet
  - Các trang được load độc lập
  - Giảm bớt việc thiết kế lặp lại ở mỗi trang web

Bố cung trang web thường được xây dựng dựa trên frame hay table.

#### Xây dựng bố cục bằng table

Khi xây dựng bố cục bằng table, mỗi trang web trong website đều có những phần tĩnh giống nhau là tiêu đề và menu. Cách làm này tuy tốn nhiều công sức nhưng khá đơn giản khi thực hiện việc liên kết giữa các trang web.

Khi chon cách xây dưng bố cục bằng table, ban cần chú ý những điểm sau:

- Mỗi trang web đều lặp lại những phần giống nhau
- Khi chuyển từ trang này sang trang khác, toàn bộ các thành phân đều được nạp lại
- Vì mỗi thành phần lặp lại trên mỗi trang web là độc lập với nhau, sửa đổi trên một trang sẽ dẫn đến phải sửa trên nhiều trang khác
- Các công cụ hỗ trợ xây dựng Website đều sử dụng cách này vì đơn giản và dễ phát sinh code hàng loat

## Xây dựng bố cục bằng frame

Khi xây dựng bố cục bằng frame, các trang web chỉ cần chứa các nội dung chính cần hiển thị tương tự như các form trong một ứng dụng Windows. Các thành phần như tiêu đề và menu sẽ được đặt trong những trang web riêng. Cách làm này chỉ cần xây dựng các thành phần dùng chung một lần nên tốn ít thời gian, khi chọn cách xây dựng bố cục cho website bằng frame, ban cần chú ý những điểm sau:

- Chỉ có thể sử dụng với các trình duyệt hỗ trợ frame
- Khi chuyển từ trang này sang trang khác, chỉ có trang web nội dung chính cần phải nạp lại, các thành phần khác của bố cục sẽ giữ nguyên.
- Việc sửa đổi các thành phần dùng chung khá dễ dàng vì chỉ phải thực hiện một lần.



# Bài 3

# THIẾT KẾ FORM NHẬP LIỆU

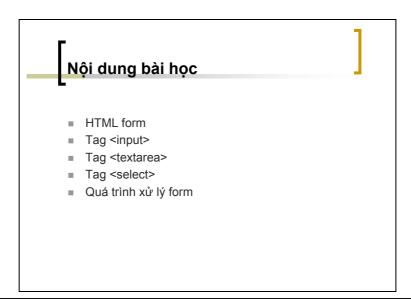
# Tóm tắt

Lý thuyết 4 tiết - Thực hành 8 tiết

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Do bài này được trình bày độc lập với phần JavaScript và ASP, giáo viên không phải bận tâm đến các vấn đề xử lý dữ liệu trên form, thay vào đó giáo viên nên tập cho học viên các thiết kế form dựa trên table và thói quen sử dụng thuộc tính name, value, để xác định tên điều khiển và đặt các giá trị mặc định cho form khi hiển thị. Điều này sẽ giúp học viên dễ dàng hơn khi bước sang các bài sau.  Sau khi hoàn tất bài học này, học viên có thể:  Sử dụng tag <form> để xác lập một form trong trang web  Sử dụng các tag HTML để tạo ra các điều khiển trên form  Hiểu được quá trình xử lý một form bao gồm hai giai đoạn: xử lý tại client và xử lý tại server  Trình bày được ưu và nhược điểm của các giai đoạn xử lý form.</form>	<ul> <li>I. Giới thiệu bài học</li> <li>II. HTML Form</li> <li>III. Quá trình xử lý form</li> <li>IV. Ưu và nhược điểm của các bước xử lý</li> </ul>	3.1, 3.2	them



# l. Giới thiệu bài học



Một website tĩnh thường chỉ có những trang web với các văn bản và hình ảnh. Mặc dù đem lại cho người đọc nhiều thông tin, điểm hạn chế nhất là website hoàn toàn không quan tâm đến bạn là ai, đã từng truy cập tới website bao giờ chưa,... Các website ngày nay đã không còn dừng lại ở mức đó, vượt xa hơn, chúng là những ứng dụng web có khả năng xác định người dùng thông qua tên người dùng và mật khẩu, cho phép bạn gửi và nhận các thông tin phản hồi như đặt mua sách báo, gửi nhận tin trong các forum hay cung cấp các dịch vụ mail, chat,...

Để làm điều này, công cụ đầu tiên cần phải có là các form trên trang web cho phép người dùng nhập thông tin. Trong phần này, chúng tôi sẽ trình bày các kiến thức về HTML form, các bài tiếp theo sẽ lần lượt hướng dẫn bạn các kỹ thuật xử lý dữ liệu của người dùng, biến website của bạn thực sự trở thành những ứng dụng web linh hoạt và "thông minh".



#### II. HTML Form

### HTML Form

HTML cung cấp bột tag tạo form, giúp trang web có thể nhận dữ liệu từ người dùng nhưng không có khả năng xử lý các điều khiển trên form

<form name="...">
<!-các điều khiển trong form-->
</form>

Một trang web có thể có 1 hay nhiều form. name giúp phân biệt các form khác nhau trên trang web

Để cho phép người dùng nhập dữ liệu vào trang web, HTML cung cấp các tag để xây dưng form nhập liêu.

Các tag về form của HTML giúp bạn tạo ra các màn hình nhập liệu như các form trên Windows, tuy nhiên, **bạn không thể dùng HTML để xử lý bất cứ dữ liệu nhập nào của người dùng**.

Để tạo một form, bạn sử dụng tag <form></form>. Vì trên một trang web bạn có thể có 1 form hay 100 form, tag <form> giúp bạn khai báo với web browser rằng tất cả những điều khiển textbox, checkbox, button,... nằm trong <form></form> là thuộc về một form duy nhất.

Để phân biệt giữa các form trên một trang Web, bạn cần gán giá trị cho thuộc tính name của <form>. Dù chỉ có một form trên trang web bạn cũng nên đặt tên cho form vì khi xử lý dữ liệu của người dùng, bạn sẽ cần đến tên form.

3 tag cơ bản trong form để tao ra các điều khiển là:

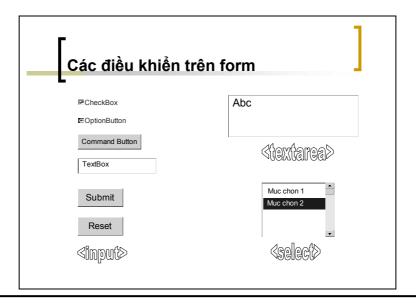
<input> : Dùng để tạo Button, TextBox, CheckBox,...

<textarea> : Tao TextBox nhiều dòng, côt

<select> : Tạo ComboBox và ListBox



### II.1. Các điều khiển trên form



Web form, tương tự như các form trong ứng dụng Windows cũng cung cấp cho người lập trình đầy đủ các điều khiển cơ bản mà một ứng dụng thường có như TextBox, Command Button, CheckBox, ComboBox,...

Như bạn vừa thấy ở phần trước, HTML cung cấp 3 tag để tạo các điều khiển trên form là <input>, <textarea> và <select>. Trong phần này, chúng ta sẽ lần lượt xem xét các sử dụng các tag này.



### II.2. Tag <input>

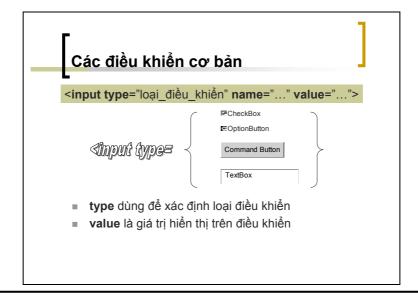


Hầu hết các điều khiển cơ bản trên form như textbox, command button, checkbox,... đều được tạo ra bằng cách sử dụnt tag <input>. Điểm tạo ra sự khác nhau giữa các điều khiển là thuộc tính **type** của tag <input> trong khi các thuộc tính khác như **name** hay **value** đều tương tự như nhau.

Trong phần này, bạn sẽ được trình bày cách sử dụng tag <input> đồng thời làm quen với hai điều khiển đặc biệt của form là Submit và Reset mà việc xử lý những điều khiển này do trình duyệt tự thực hiện.



#### II.3. Các điều khiển cơ bản



Hầu hết các điều khiển cơ bản trong <form> đều được tạo bằng tag <input>, cấu trúc của tag <input> như sau:

<input type="loại\_điều\_khiển" name="..." value="...">

- type : loại điều khiển muốn tạo. Có 5 loại điều khiển là:
  - TextBox "text"
  - CheckBox "checkbox"
  - OptionBox "radio"
  - Button "button"
  - Submit/Reset "submit"/"reset"
- name : tên của điều khiển. Tương tự như các form bạn lập trình trên Windows, mỗi điều khiển nên có một tên riêng biệt. Riêng với trường hợp OptionBox, để gom nhiều option thành một nhóm, các OptionBox sẽ có giá trị của thuộc tính name giống nhau.
- value : chuỗi văn bản hiển thị trên điều khiển. Với TextBox là nội dung của TextBox, với Button (kể cả Submit và Reset) là tiêu đề của điều khiển.

#### 🗷 Chú ý:

Với CheckBox và OptionBox, thuộc tính *checked* dùng để đánh dấu chọn vào CheckBox hay OptionBox khi trang web hiển thị.

Thuộc tính size của textbox dùng để chỉ định chiều rộng của textbox, đơn vị của size là số ký tự. Tuy nhiên, nội dung của textbox không bị giới hạn bởi size.

#### II.3.1.



#### II.4. Submit & Reset



Về mặt trình bày, submit và reset cũng là hai button tương tự như khi tạo ra bằng <input> với type="button". Tuy nhiên, đây là hai nút đặc biệt của form và được xử lý bởi web browser.

Khi bạn click vào nút reset, các giá trị mà bạn nhập vào trên điều khiển đều bị huỷ bỏ và trở về trạng thái ban đầu khi trang web hiển thị.

Khi click vào submit, bạn sẽ thấy màn hình bị nháy. Đây là lúc web browser thu thập các thông tin bạn nhập vào form để gửi về web server chờ xử lý và nhận kết quả.

#### ⁴ Ví dụ:

```
<form name="form1">
Tên đăng nhập
  <input type="text" name="txtUsername">
  </t.r>
  Mật khẩu
    <input type="text" name="txtPassword">
   
    <input type="checkbox" name="chkNewUser">Tao người
dùng mới
  <input type="submit" value="Đăng nhập" name="cmdSubmit">
<input type="reset" value="Xoá trắng" name="cmdReset">
</form>
```



### Kết quả:

| Tên đăng nhập<br>Mật khẩu | □ Tạo người dùng mới |
|---------------------------|----------------------|
| Đăng nhập                 | Xoá trắng            |

Qua ví dụ trên, có thể thấy rằng trong <form> </form> bạn được phép sử dụng các tag định dạng để trình bày form như <table>, <p>,...



#### II.5. Tag <textarea>



Điều khiển TextBox mà bạn tạo bằng tag <input> chỉ có khả năng nhận vào **một** dòng văn bản. Để có một TextBox cho phép nhập nhiều dòng bạn sử dụng tag <textarea></textarea>.

```
<textarea rows="..." cols="..." name="...">Nội dung...</textarea>
```

Khác với tag <input> là một tag đơn, <textarea> cần kết thúc bởi </textarea>. Nội dung của TextBox tạo bằng <textarea> cũng không định bởi giá trị của thuộc tính value, thay vào đó, phần nội dung này nằm giữa cặp tag.

Thuộc tính cols của <textarea> tương tự như size của <input type="text"> xác định chiều rộng của TextBox tính bằng số ký tự. Thuộc tính rows cho biết chiều cao của TextBox.

#### ⁴ Ví du:

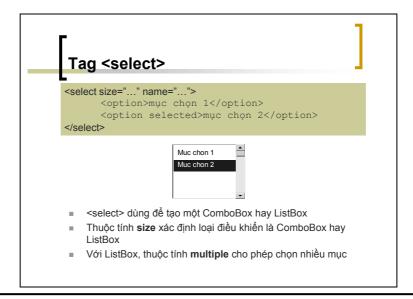
```
<form name="form1">
  <input type="text" name="T1" value="Xin chào">
  <textarea rows="3" name="S1" cols="50">Hello
  </textarea>
</form>
```

#### Kết quả:





#### II.6. Tag <select>



Điều khiển ComboBox và ListBox phức tạp hơn so với các điều khiển cơ bản vì cần định nghĩa các mục chọn. HTML sử dụng tag <select> để khai báo cả hai loại điều khiển này:

#### ⁴ Ví dụ:

#### Kết quả:

#### Môn học: Visual Basic Visual C++ Java

Thuộc tính size giúp web browser xác định điều khiển là một ComboBox (size="1") hay ListBox (size>1).

Tag <option>, như bạn thấy trong ví dụ, được đặt trong cặp tag <select> để định nghĩa các mục chọn.

- Tên mục chọn được đặt trong trong cặp tag <option>...</option>
- Mục chọn mặc định được biểu thị qua thuộc tính selected.
- Thuộc tính value cho biết giá trị của mục chọn. Bạn sẽ cần biết giá trị của mục chọn khi xử lý dữ liệu nhập của form. (phần này được trình bày ở chương dành cho ASP và JavaScript)

Nếu bạn muốn ListBox có thể được chọn nhiều mục cùng một lúc, hãy sử dụng thuộc tính *multiple* của tag <select>.



# III. Quá trình xử lý form

### Quá trình xử lý form

- Xử lý dữ liệu trong form có thể chia làm 2 bước
  - Tại Client: các xử lý tính toán, kiểm tra dữ liệu
  - Tại Server: các xử lý tìm kiếm, thêm, xoá, cập nhật dữ liệu
- Các xử lý được thực hiện dựa vào ngôn ngữ lập trình gọi là script
  - Client side script phổ biến nhất là JavaScript
  - Server side script được hỗ trợ mạnh nhất là ASP

Bạn đã quen sử dụng các ứng dụng trên Windows hay trên Web, các trang web không chỉ đem thông tin đến cho bạn mà còn nhận các thông tin từ phía người dùng để xử lý. Các thông tin này được nhập thông qua các form nhập liệu mà ta vừa mới đề cập. Nhưng làm thế nào để xử lý những thông tin đó? HTML không đủ khả năng làm việc này, cần phải có những công cụ khác cũng như một cơ chế giúp phối hợp HTML với những công cu này.

Việc xử lý form có thể chia làm 2 bước, tại client và tại server. Người lập trình sẽ phải sử dụng một ngôn ngữ gọi là script để đọc và xử lý giá trị của các điều khiển trong form.

- Ngôn ngữ script xử lý form tại client gọi là client side script
- Ngôn ngữ script xử lý form tại server gọi là server side script

Vì mỗi người trong chúng ta có thể chọn dùng những loại Web browser khác nhau, việc một ngôn ngữ Client side script được nhiều browser hỗ trợ là rất quan trọng. Nhờ vào sự hỗ trợ đó, người xây dựng ứng dụng web có thể đảm bảo rằng ứng dụng của mình chạy đúng trên nhiều máy. Client side script thông dụng nhất là **JavaScript**.

Server side script thì có thể được người lập trình chủ động lựa chọn hơn vì nó chỉ chạy tại Web Server do người lập trình quản lý. Tuy vậy, lựa chọn một Server side script lại dựa trên những khả năng của ngôn ngữ đó cùng với các công cụ hỗ trợ lập trình.

**Active Server Page** (ASP) là một ngôn ngữ thông dụng bởi là sản phẩm của Microsoft và được sự hỗ trợ của hệ điều hành Windows và bộ công cụ Visual InterDEV. Bên cạnh đó, ASP khá đơn giản và dễ học với nhiều người lập trình hiện nay đã quen với môi trường lập trình Windows sử dụng Visual Basic. Tuy vậy, bạn cũng nên biết rằng ASP không có được tốc độ nhanh nhất và trên Internet, không có nhiều Web Server miễn phí cung cấp dịch vụ hosting cho ứng dụng web viết bằng ASP.



# IV. Ưu và nhược điểm của các bước xử lý

#### Ưu và nhược điểm của các bước xử lý

- Client
  - Kiếm tra các lỗi nhập liệu
  - o Hiển thị dữ liệu
  - Không thể thực hiện các xử lý logic phức tạp
- Server
  - Server phải đáp ứng các yêu cầu từ nhiều client
  - Thực hiện các xử lý logic phức tạp, liên quan đến cơ sở dữ liệu

Tương tự như một ứng dụng Client/Server, việc phân chia nhiệm vụ xử lý form thành 2 bước có những ưu và nhươc điểm.

Xử lý tại client:

- Nhanh, dữ liệu không mất thời gian chuyển về server rồi quay lại
- Chỉ thực hiện được những xử lý đơn giản và độc lập.

Tại client, bạn không thể truy cập vào CSDL trên server để lấy các thông tin liên quan.

Bạn cũng không thể lưu lại thông tin mới nhận được thành file hay vào CSDL.

Xử lý tại server:

- Chậm, dữ liệu phải được chuyển về server và chờ nhận kết quả trả về. Nếu dữ liệu bị sai, người dùng phải nhập liệu từ đầu.
- Thực hiện đầy đủ các xử lý cần thiết, có thể truy cập CSDL và file.

Do có những ưu và nhược điểm riêng, việc phân chia công việc sẽ thực hiện ở client hay server là rất cần thiết. Thông thường, client sẽ kiểm tra tính đúng đắn của dữ liệu nhập vào. Nếu dữ liệu đã nhập là hợp lệ, client sẽ gửi dữ liệu đó về cho server tiếp tục xử lý, ngược lại client sẽ báo lỗi để người dùng nhập lại mà không cần phải chờ đợi lâu.

Việc xử lý tại Client được thực hiện sử dụng các đoạn script viết bằng JavaScript đặt trong bản thân trang web. Tại server, có nhiều công cụ để xử lý dữ liệu gửi về từ form. ASP là một trong những công cụ đó.

Trong các bài tiếp theo, chúng ta sẽ tiếp tục tìm hiểu cách xử lý form tại Server bằng ASP. Tiếp đó, trong phần JavaScript, chúng ta sẽ tìm cách tăng tốc độ và hiệu quả của ứng dụng web bằng cách chuyển một số công việc xử lý từ phía Server về Client.



# Bài 4

# MÔ HÌNH ỨNG DỤNG WEB

# Tóm tắt

Lý thuyết 2 tiết - Thực hành 2 tiết

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài giảng này chủ yếu bao gồm các khái niệm lý thuyết về mô hình ứng	I. Giới thiệu bài học		
	II. Mô hình ứng dụng		
dụng với mục tiêu giúp học viên phân biệt được	III. Úng dụng Web với mô hình Client/Server		
nơi sẽ thực hiện các chức năng xử lý khác nhau trong một ứng dụng Web.	IV. Microsoft Internet Information Server		
Giờ thực hành, học viên	V. Khái niệm về Script		
sẽ thực hiện việc cài đặt Web Server và chuyển các trang web, website mà mình đã thực hiện lên Web Server để chạy thử nghiệm.	VI. Quá trình xử lý một yêu cầu của Web Server		
Sau khi hoàn tất bài học này, học viên có thể:			
<ul> <li>Hiểu được khái niệm về mô hình ứng dụng và trình bày các lớp ứng dụng trong mô hình</li> </ul>			
<ul> <li>Trình bày được mô hình ứng dụng được sử dụng cho ứng dụng Web</li> </ul>			
<ul> <li>Cài đặt thành công Web Server trên hệ điều hành Windows</li> </ul>			



# I. Giới thiệu bài học

## Nội dung bài học

- Mô hình ứng dụng
- Úng dụng Web với mô hình Client/Server
- Microsoft Internet Information Server
- Khái niệm về Script
- Quá trình xử lý một yêu cầu của Web Server

Trong các bài trước, chúng ta đã xây dựng những trang web tĩnh bằng HTML. Bạn cũng đã được trình bày những kiến thức về form, mô tả về quá trình xử lý dữ liệu nhập từ form và những khái niệm về mô hình Client/Server.

Trong chương này, chúng ta tiếp tục xem xét mô hình Client/Server khi nó được áp dụng cho ứng dụng Web đồng thời, bắt tay vào cấu hình Internet Information Server - Web Server của Microsoft trên hệ điều hành Windows để xem ứng dụng web chạy như thế nào.



# ll. Mô hình ứng dụng

Mô hình ứng dụng

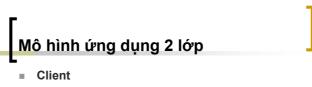
Mô hình 2 lớp
Mô hình 3 lớp

Một ứng dụng lớn khó lòng có thể đóng gói tất cả các chức năng về tính toán, xử lý cùng với các dịch vụ về CSDL, truyền thông trên mạng. Thay vào đó, ứng dụng thường được phân chia thành nhiều lớp, mỗi lớp thực hiện các chức năng riêng và sử dụng các ứng dụng nền tảng có sẵn như Database Server, Web Server, Transaction Server,... để thực hiện các xử lý cấp thấp.

Việc phân chia ứng dụng thành nhiều lớp là một cách xác định mô hình ứng dụng mà dựa trên mô hình đó, người ta mới bắt đầu đi xây dựng chương trình. Phần này giới thiệu với bạn hai mô hình thường gặp trong các ứng dụng phân tán là mô hình 2 lớp và mô hình N lớp trong đó, để đơn giản hoá khái niệm mô hình N lớp, chúng tôi sử dụng khái niệm ứng dụng theo mô hình 3 lớp để bạn dễ liên tưởng tới mô hình được sử dụng cho ứng dụng web.



### II.1. Mô hình ứng dụng 2 lớp

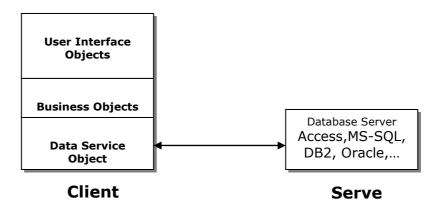


- Xử lý giao diện
- Thực hiện các chức năng tính toán, thống kê, tìm kiếm
- o Đọc, tìm kiếm, cập nhật dữ liệu
- Server
  - Lưu trữ dữ liệu tập trung
  - Cung cấp các dịch vụ để client đọc và cập nhật dữ liệu

Mô hình đơn giản và phổ biến của một ứng dụng phân tán là mô hình Client/Server 2 lớp. Trong mô hình này việc xử lý dữ liệu được thực hiện trên server gọi là database server do đó

- Dữ liệu trở thành tài nguyên được chia sẻ bởi nhiều người dùng.
- Xử lý dữ liêu tập trung.

Tuy nhiên, database server chỉ có thể lưu giữ và truy cập dữ liệu, phần công việc còn lại của ứng dụng bao gồm các thao tác trên dữ liệu và hiển thị kết quả cho người dùng vần còn thực hiện trên client.





### II.2. Mô hình ứng dụng 3 lớp

# Mô hình ứng dụng 3 lớp

- Mô hình 3 lớp hỗ trợ nhiều người dùng hơn so với mô hình 2 lớp
- Giảm bớt gánh nặng trên Client
- Úng dụng có thể thay đổi, chỉnh sửa dễ dàng vì không cần thiết phải sửa đổi trên phía Client

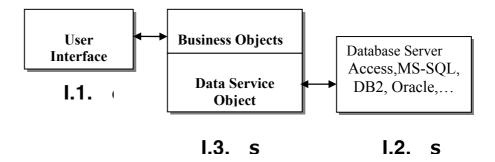
Mô hình 2 lớp không thể áp dụng với ứng dụng lớn có nhiều người dùng:

- Dữ liệu truyền trên mạng lớn.
- Database Server trở thành "cổ chai" khi cùng lúc nhiều người truy cập cùng một tài nguyên.

Mô hình Client/Server 3 lớp giải quyết vấn đề trên bằng cách dùng thêm một server tương tác giữa client và database server gọi là Application Server. Application server giữ nhiệm vụ tương tác với database server. Việc thao tác dữ liệu nhận được bao gồm nhiều xử lý khác nhau được phân chia trên application server hay trên client tuỳ theo từng ngữ cảnh cụ thể.

Ngoài việc giải quyết các vấn đề của mô hình 2 lớp, mô hình 3 lớp còn có ưu điểm:

- Dễ áp dụng: việc sử dụng nhiều Application server dễ thực hiện hơn việc dùng nhiều Database server vì khi đó dữ liệu cũng sẽ bị phân tán.
- Quản lý dễ dàng: việc nâng cấp, sửa đổi có thể chỉ cần thực hiện ở phía server.





# III. Úng dụng Web với mô hình Client/Server

#### 「Ứng dụng Web với mô hình Client/Server

- Úng dụng Web sử dụng mô hình Client/Server 3 lớp
  - Client là các Web Browser đã có sẵn
  - Browser không thể thực hiện chức năng xử lý, tính toán
  - Application Server là Web Server, trên đó thực hiện các yêu cầu mà Web Browser gửi tới
  - Kết quả xử lý của Web Server phải ở dạng HTML

Ứng dụng Web là ứng dụng phân tán do có nhiều người dùng, truy cập từ nhiều vị trí khác nhau.

Người dùng sử dụng các trình duyệt Web (Web browser) khác nhau để truy cập Internet. Do người dùng không chỉ truy cập một mà rất nhiều trang Web nên Browser không thể đảm nhận việc xử lý dữ liệu mà chỉ có thể:

- Nhận yêu cầu từ người dùng để chuyển tới Web Server.
- Hiển thị kết quả mà Web Server gửi về ở dạng HTML (cả XHTML và XML cho các Browser mới)

Trong mô hình ứng dụng Web:

- Client hay Web Browser đã có sẵn, chẳng hạn: Internet Explorer, Netscape Navigator, ...
- Application Server chính là Web Server, thực hiện các yêu cầu mà người dùng gửi đến thông qua Browser.

Vì Browser là có sẵn và chỉ hiểu HTML, các đối tượng xử lý yêu cầu gửi về từ Client trên Web server phải trả về kết quả ở dạng HTML. Trong khi đó, các làm việc giữa các đối tượng trên Web Server và Database server cũng tương tự như các đối tượng trong mô hình 3 lớp.



## **IV. Microsoft Internet Information Server**

# Microsoft Internet Information Server

- Giới thiệu Internet Information Server
- Cài đặt Web server trên Windows 2000
- Cài đặt Web server trên Windows 98
- Cấu hình Internet Information Server
- Cấu hình Personal Web Server

Một thành phần quan trọng trong mô hình ứng dụng Web là Web Server. Đây là thành phần giữa vai trò xử lý các yêu cầu của web client. Phần này sẽ giới thiệu với bạn phần mềm Internet Information Server chạy trên các hệ điều hành Windows và cách cài đặt nó.



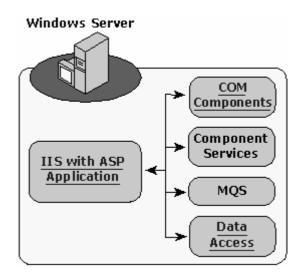
#### IV.1. Giới thiệu Internet Information Server

# Giới thiệu Internet Information Server

- IIS là phần mềm Web Server của Microsoft dành cho Windows
- Tích hợp cùng với Windows
- Kết hợp với ASP để xây dựng các ứng dụng Web
- Cung cấp khả năng sử dụng các COM component: ADO, FSO

Internet Information Server (IIS) là một Internet File và Application Server dành cho hệ điều hành Windows NT Server.

IIS có thể được sử dụng như một Web server, kết hợp với ASP để xây dựng các ứng dụng Web tận dụng các điểm mạnh của Server-side Script, COM component,... theo mô hình Client/Server.



IIS có rất nhiều phiên bản, đầu tiên được phát hành rời trong bản Service pack của WinNT.

- Các phiên bản Windows 2000 đã có tích hợp IIS 5.0.
- Windows XP tích hợp IIS 5.5
- Windows XP .NET Server tích hợp IIS 6 hỗ trợ các tính năng dành cho .NET của ASP.NET và Web Service.



# IV.2. Demo: Cài đặt Web server



Trong phần này, giáo viên sẽ minh họa cách cài đặt web server (tùy theo cấu hình máy của phòng thực hành):

- Cài đặt Personal Web Server trên Windows 98/Me
- Cài đặt IIS trên Windows 2000/XP
- Cấu hình IIS/Personal Web server
- Tạo một ứng dụng Web trên Web server



# V. Khái niệm về Script

### Khái niệm về Script

- Script là các đoạn chương trình con, được viết bằng những ngôn ngữ lập trình dành cho ứng dụng Web
- Các script được đặt trong trang web và thi hành tuỳ theo ngữ cảnh mà người dùng thao tác với trang web
  - o Client-side script
  - Server-side script

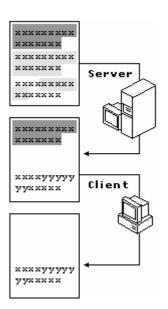
Để hiển thị văn bản, hình ảnh hay các liên kết trên trang web, chúng ta sử dụng ngôn ngữ HTML, tuy nhiên để tương tác với người dùng như:

- Đáp ứng lại sự kiện người dùng click một nút
- Thu thập các thông tin mà người dùng nhập vào
- Truy xuất dữ liệu từ một cơ sở dữ liệu
- Chuyển tới một trang web tuỳ theo từng điều kiện

chúng ta cần sự hỗ trợ của Web Broser và Web Server, cung cấp khả năng thi hành các câu lệnh trong chương trình tuỳ theo ngữ cảnh mà người dùng thao tác với trang web.

Cơ chế lập trình và thi hành các chương trình trong trang Web được Web Server và hầu hết các Web Browser hỗ trợ. Các chương trình mà bạn viết trong trang Web gọi là các Script. Ngôn ngữ để viết nết các Script như đã giới thiệu với bạn ở những chương trước là JavaScirpt, VBScript, ASP, PHP, JSP,...

Các Script là "trong suốt" đối với người dùng. Người duyệt web sẽ thấy trang web của bạn sinh động và "thông minh" như những ứng dụng trên Windows. Tuy nhiên, đối với người lập trình, bạn lại phải biệt giữa hai loại Script: Client side script và Server side script.



#### Các loại Script

#### Server script

- Là những đoạn script trong trang web được thực hiện trên Web server trước khi trang web được gửi đến browser.
- Khi trang web được gửi đến browser, server đã thi hành các đoạn server script, và xoá các đoạn code của chúng trong trang web.
- Server script thường dùng để truy cập cơ sở dữ liệu (CSDL), xử lý dữ liệu được người dùng nhập vào từ HTML form, tự động chuyển đến các trang web khác,...



#### **Client Script**

- Là các đoạn script được thực hiện tại browser.
- Client script là một phần của trang web thông thường được thi hành để đáp ứng lại một sự kiện, thay đổi thể hiện của trang web trên browser hay kiểm tra dữ liệu người dùng nhập vào form ngay tại browser.

Global script là các đoạn script sẽ được thi hành ngay khi trang web được xử lý.

#### Client-side Script - JavaScript

Bao trong tag <Script> </Script> Được thi hành trên browser

Phụ thuộc vào sự hỗ trợ của browser, và hệ thống người dùng đang sử dụng.

Hướng sự kiện, các script sẽ được thi hành khi có các sự kiện xảy ra.

Tương tác trực tiếp với người dùng

Source code được lưu trong trang web.

#### Server-side Script - ASP

Bao trong tag <% %>

Thi hành trên Web server.

Không phụ thuộc vào hệ thống của người dùng

Được Web server thi hành tuần tự từ trên xuống dưới

Không thể tương tác trực tiếp với người dùng.

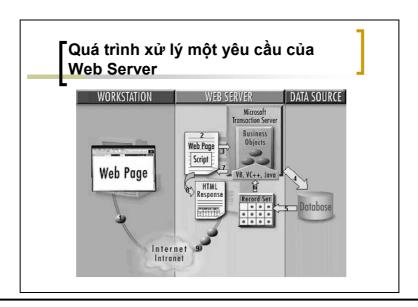
Souce code sẽ được Web server gỡ bỏ khỏi trang Web sau khi thực hiện.

Không thể truy xuất nội dung trang web nhưng có thể truy xuất các đối tượng trên server thông qua COM.

Có thể phát sinh ra các Client side script



# VI. Quá trình xử lý một yêu cầu của Web Server



- 1. Người dùng (Web Browser) gửi yêu cầu tới Web Server
- 2. Web Server xác định trang Web được yêu cầu
- 3. Nếu trong trang web có các đoạn server script, Web Server sẽ thi hành.
- 4. Các đối tượng trong server script có thể truy cập CSDL nếu cần
- 5. Database Server trả kết quả là RecordSet về cho Web Server
- 6. Các đối tượng trong server script xử lý kết quả của RecordSet
- 7. Kết quả của RecordSet được chuyển sang dạng HTML để đưa vào trang web
- 8. Web server hình thành trang web kết quả
- 9. Trang web kết quả được đóng gói và gửi về cho Web browser



# Bài 5

# SỬ DỤNG CÁC ĐỐI TƯỢNG CƠ BẨN TRONG ASP

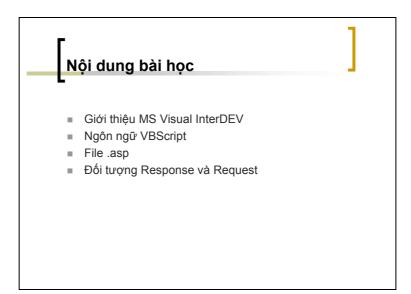
# Tóm tắt

Lý thuyết 6 tiết - Thực hành 8 tiết

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Sau khi hoàn tất bài học này, học viên có thể:  Sử dụng môi trường Visual InterDEV để tạo các trang ASP và các Web project.  Sử dụng được đối tượng Request với sự phân biệt giữa hai tập hợp Form và QueryString khi muốn truy cập các thông tin mà client gửi về  Sử dụng được đối tượng Response để thực hiện các kết xuất ra trang web.	<ul> <li>I. Giới thiệu bài học</li> <li>II. Giới thiệu MS Visual InterDEV</li> <li>III. Ngôn ngữ VBScript</li> <li>IV. File .asp</li> <li>V. Đối tượng Response và Request</li> </ul>	5.1 – 5.3	



# l. Giới thiệu bài học



Ở bài trước, chúng ta đã cài đặt được Web server và tổ chức được cấu trúc thư mục để lưu trữ các file trong một ứng dụng Web. Giờ đây, bạn đã sẵn sàng bắt tay xây dựng các trang ASP với năm đối tượng cơ sở: Request, Response, Session, Application và Server.

Kể từ bài này, bạn sẽ sử dụng các kỹ năng lập trình của mình để xây dựng các trang web. Chúng tôi coi những kỹ năng lập trình về cấu trúc điều khiển, thủ tục, hàm, sử dụng đối tượng và sự kiện là những kỹ năng cơ sở mà bạn đã có sẵn. Do đó, bài này sẽ giới thiệu với bạn môi trường làm việc để xây dựng ứng dụng Web: Visual InterDEV, sơ lược về ngôn ngữ VBScript mà bạn sẽ dùng để viết các đoạn script và cuối cùng, tập trung vào hai đối tượng chính trong ASP là Response và Request.



# II. Giới thiệu MS Visual InterDEV

# Giới thiệu MS Visual InterDEV

- MS Visual InterDEV là một công cụ thuộc bộ Visual Studio 98
  - Hỗ trợ tổ chức và quản lý một ứng dụng Web
  - Hỗ trợ xây dựng các trang HTML và ASP
  - Chạy và debug chương trình trực tiếp

Như bạn thấy, mỗi ngôn ngữ lập trình đều có những công cụ hỗ trợ gọi là: môi trường lập trình (Development Environment). Các ngôn ngữ như Java, C++ hay Visual Basic rất phổ biến và có các IDE (Integrated Development Environment) rất mạnh. Bộ công cụ MS Visual Studio 98 cũng là một trong những bộ công cụ mạnh dành cho người lập trình. Visual Studio 98 cung cấp tới 3 IDE:

- Visual C++
- Visual Basic
- Visual InterDEV

Visual InterDEV là IDE dành cho việc xây dựng các ứng dụng Web. Đối với những người đã quen với các môi trường lập trình của Microsoft như Visual Basic hay Visual C++ thì sử dụng Visual InterDEV khá đơn giản.

Với Visual InterDEV bạn có thể:

- Xây dựng, triển khai và quản lý ứng dụng Web
- Tạo các trang HTML, tạo và debug các trang ASP
- Viết các trang HTML sử dụng JScript (một ngôn ngữ của Microsoft tương tự như JavaScript của Nestcape) và VBScript
- Quản lý cơ sở dữ liêu được sử dụng trong ứng dụng Web
- **.**..

Đồng thời, trong quá trình sử dụng InterDEV bạn cũng sẽ nhận được các hỗ trợ rất tiện dụng như:

- Hổ trợ trong quá trình debug chương trình
- Tư đông hoàn chỉnh câu lênh
- Thiết kế trang web giống như Front Page
- **.**..



# II.1. Demo: Tạo web project bằng Visual InterDEV



Trong phần này, giáo viên sẽ minh họa các dùng Visual InterDEV để tạo một web project:

- Tạo mới một project
- Thêm/xóa các thành phần trong project
- Thêm các đọan server script vào trang web
- Sử dụng Script Outline



# III. Ngôn ngữ VBScript

# Ngôn ngữ VBScript

- VBScript là ngôn ngữ sử dụng để viết các script cho ứng dụng Web ở cả Client và Server
- Có hệ thống cú pháp lệnh giống với Visual Basic
- Hạn chế hơn so với Visual Basic vì chỉ là một ngôn ngữ Script.

Khác với những ngôn ngữ lập trình Web khác như Perl, PHP, Cold Fusion,... đều sử dụng các ngôn ngữ riêng của mình, ASP sử dụng các ngôn ngữ đã có sẵn và phổ biến như JavaScript hay VBScript.

# Ngôn ngữ Script

Các ngôn ngữ như C++, Visual Basic là những ngôn ngữ lập trình được dùng xây dựng các chương trình lớn và phức tạp. Ngôn ngữ Script là ngôn ngữ giới hạn dùng để tạo các chương trình nhỏ và đơn giản.

Các lệnh trong script sẽ được dịch và thi hành một cách tuần tự từ trên xuống dưới bởi một scripting engine. Ví dụ, Internet Explorer hay Nescape Navigator phải có JavaScript scripting engine để hỗ trợ thực hiện client-side script viết bằng JavaScript.

### **VBScript**

ASP cho phép người lập trình viết server-side script bằng nhiều loại ngôn ngữ script khác nhau. Hai ngôn ngữ phổ biến nhất là JavaScript và VBScirpt. VBScript gần như tương tự Visual Basic nên rất nhiều người có thể sử dụng. Trình duyệt phổ biến nhất thế giới của Microsoft: Internet Explorer hỗ trợ không chỉ JavaScript mà cả VBScript làm ngôn ngữ để lập trình script trên phía Client. JavaScript lại là ngôn ngữ mà gần như chắc chắn bạn sẽ phải biết khi xây dựng các ứng dụng Web. ASP tận dụng đặc điểm phổ biến của ngôn ngữ để cố gắng đem đến tính dễ dùng cho nhiều người, đồng thời giúp họ tập trung vào những đối tượng, cơ chế làm việc của ứng dụng Web trong ASP thay vì cất công học thêm ngôn ngữ lập trình mới.

**Chú ý:** Trong phần này, chúng tôi trình bày về VBScript ở dạng tham khảo. Nếu bạn không biết Visual Basic cũng không đáng lo ngại vì một khi đã từng lập trình với bất cứ ngôn ngữ nào, sử dụng VBScript sẽ không là vấn đề đối với bạn.

Nếu bạn là một lập trình viên Visual Basic, bạn sẽ thấy VBScript hạn chế hơn so với Visual Basic. Nguyên nhân là VBScript chỉ là một ngôn ngữ viết Script, không phải là một ngôn ngữ **lập trình** thực thụ. Điều này cũng giống như việc bạn so sánh giữa JavaScript và Java.



# III.1. Tham khảo về ngôn ngữ VBScript

# Tham khảo về ngôn ngữ VBScript

- Kiểu dữ liệu & biến
- Hàm & thủ tuc
- Cấu trúc điều khiển
- Các toán tử
- Tham khảo các hàm của VBScript

# Kiểu dữ liệu

VBScript chỉ có 1 kiểu dữ liệu chính là kiểu variant. Kiểu dữ liệu này có thể lưu giữ bất kỳ loại thông tin nào bao gồm kiểu chuỗi, số, ngày giờ và đối tượng.

Variant là kiểu dữ liệu mặc định nên không cần chỉ ra trong các câu khai báo biến hay kiểu dữ liệu trả về của hàm.

Tùy theo ngữ cảnh sử dụng, dữ liệu sẽ có kiểu thích hợp

### Biển

Sử dụng lựa chọn Option Explicit để bắt buộc khai báo biến trước khi dùng.

Có 3 từ khoá khai báo biến là : Dim, Public và Private.

### ₼ Vd:

Dim bienVBScript
Public mangVBScript(10)
ReDim mangVBScript(30)
ReDim Preserve mangVBScript(3)

# Hàm và thủ tục

Khác với JavaScript, VBScript cung cấp 2 cách viết hàm: Sub và Function.

Cú pháp khai báo một thủ tục:

End Function

Sub tên\_thử\_tực(các tham số)
...
...
End Sub
Cú pháp khai báo một hàm
Function tên\_hàm (các tham số)
...



# IV. File .asp



- File .asp có thể chỉ chứa các ASP Script hay kết hợp cùng với HTML
- Một đoạn ASP script phải đặt trong tag <% %>
- Thứ tự thực hiện các script của Web Server là từ trên xuống dưới
- Có thể tổ chức script thành sub/function và lưu thành file thư viện dùng chung

Các trang .asp là các trang Web Server sẽ thi hành trước khi gửi về cho Client. Trang .asp có thể chỉ chứa hoàn toàn các đoạn script nhưng cũng có thể là một trong HTML thông thường với các đoạn script xen kẽ bên trong.

# Phân biệt giữa Script và HTML

Web server nhận ra các đoạn script thông qua cặp tag <%%>. Các đoạn lệnh mà bạn viết bằng VBScript hay JavaScript phải đặt trong tag này, ngược lại, các đoạn HTML đặt trong <%%> lại không có ý nghĩa và sẽ bị coi như câu lệnh sai cú pháp:



# <sup>⁴</sup> Ví dụ 2

```
  Dim dtmHour

  dtmHour = Hour(Now())

  If dtmHour < 12 Then

%>
   Good morning!

<% Else %>
   Hello!

<% End If %>
```

Ví dụ 2 là một trang .asp chỉ gồm các ASP Script, khi thử ví dụ này, bạn sẽ thấy tuỳ vào giờ hiện hành trên máy tính của bạn mà trang web kết quả sẽ có câu **Good morning!** hay **Hello!** 

Ví dụ 2 cũng cho bạn thấy cách phối hợp giữa HTML và ASP Script, nếu nhìn kỹ, bạn sẽ thấy hai câu Good morning! và Hello! thực sự là code HTML chứ không phải là lệnh trong VBScript. Ở đây, người viết đã dựa vào đặc điểm của cấu trúc điều khiển  $\it If$  để in ra kết quả tương ứng.

# <sup>♣</sup> Ví dụ 3

```
<%@ LANGUAGE=VBScript%>
<% Option Explicit %>
<응
Dim I
I=0
Sub Dem(Solan)
   Dim c
   For c=0 to Solan
      response.write I & ", "
   Next c
End Sub
응>
<SCRIPT LANGUAGE=JScript RUNAT=SERVER>
function XuatI() {
  Response. Write (I)
</SCRIPT>
<HTML>
<BODY>
  Gia tri ban dau cua I: <% XuatI() %> <BR>
  Dem 10 lan: <% Dem 10 %> <BR>
  Dem 10 lan nua: <% Call Dem(10) %>
</BODY>
</HTML>
```



# IV.1. Cấu trúc một file .asp

```
Cấu trúc một file .asp
<% Language=VBScript%>
<% Option Explicit
<#include file=...>
Dim .
Sub A()
End Sub
%>
<%
asp script
%>
<HTML><BODY>
<html tag>...</html tag>
<% asp script %>
<html tag>...<% asp script %>...</html tag>
</BODY></HTML>
```

# Cấu trúc của một trang .asp

```
Chỉ định ngôn ngữ Script sử dụng.
<% Language=VBScript%>
Khai báo Option Explicit.
<% Option Explicit
Danh sách các file inlude.
<#include virtual=...>
<#include file=...>
Khai báo dữ liệu dùng chung trong trang web và gán giá trị ban đầu.
Danh sách các hàm dùng trong trang.
Sub A()
End Sub
Function X()
End Function
Các đoạn scipt sẽ được thi hành khi trang bắt đầu được xử lý.
<%
%>
HTML xen kẽ với các đoạn script.
<HTML>
<BODY>
<html tag>...</html tag>
<% asp script %>
<html tag>...</html tag>
<html tag ...<% asp script %> ...>...</html tag>
```



</BODY>

# Giải thích

- Câu khai báo đầu tiên chỉ định ngôn ngữ VBScript là ngôn ngữ mặc định mà bạn sẽ dùng để viết các đoạn script trong trang .asp.
- Các câu lệnh #include nhằm đưa nội dung của một file .asp khác vào trang .asp, ngay tại vị trí của câu lệnh #include. Bạn có thể lập nên một bộ thư viện các hàm thường dùng, đưa chúng vào các file độc lập như những module. Khi cần thiết sử dụng, bạn chỉ cần #include chúng vào trong trang .asp.
- Các Sub/Function không được tự động thực hiện trừ khi có câu lệnh khác gọi chúng.
- Các đoạn script còn lại được Web server thực hiện theo thứ từ lần lượt từ trên xuống dưới.



# IV.2. Chon VBScript để viết trang .asp

# Chọn VBScript để viết trang .asp

- VBScript có cú pháp đơn giản
- VBScript có sẵn một tập hợp phong phú các hàm xử lý chuỗi, ngày giờ, toán học,...
- Sử dụng các đối tượng theo công nghệ COM
  - Ví du: ADO
- Tận dụng những đoạn code có sẵn của nhiều người khác

IIS sẵn có hai scripting engine hỗ trợ JavaScript và VBScript, do đó ta có thể chọn lựa một trong hai ngôn ngữ này để viết các đoạn script.

JavaScript là ngôn ngữ gần gũi với người đã quen với C++ hay Java còn VBScript là ngôn ngữ mà những lập trình viên Visual Basic quen thuộc.

Để khai báo ta sẽ dùng ngôn ngữ nào để viết một trang ASP, sử dụng cú pháp

# <%@ LANGUAGE=ScriptingLanguage %>

trong đó ScriptingLanguage có thể là VBScript hay JavaScript.

Trong phần sau, khi viết các script trên Client, bạn sẽ sử dụng JavaScript vì đây là ngôn ngữ được nhiều trình duyệt hỗ trợ hơn VBScript, vậy tại sao ta nên chọn VBScript để viết các đoạn server script?

- VBScript có cú pháp đơn giản hơn JavaScript nên các đoạn script sẽ được viết một cách rõ ràng và trong sáng hơn.
- JavaScript là người em của Java, một ngôn ngữ thuần hướng đối tượng. Để tận dụng các hàm hỗ trợ xử lý chuỗi, ngày giờ, chúng ta sẽ phải dùng các đối tượng như String, Math, Date ,... VBScript có sẵn một tập hợp phong phú hơn nhiều những hàm loại này.
- Cú pháp của JavaScript phân biệt chữ HOA và chữ thường còn VBScript thì không.
- ASP sử dụng các đối tượng theo công nghệ COM (Componet Object Model). VBScript giúp ta sử dụng các đối tượng này đơn giản hơn là dùng JavaScipt. Ví dụ về đối tượng COM là ADO (Active Data Object), một đối tượng sử dụng để truy cập cơ sở dữ liệu.
- Một lý do khác là hầu như mọi người viết ASP đều sử dụng VBScript, tại sao ta không sử dụng VBScript để dễ dàng tận dụng những đoạn code của họ một cách hiệu quả.



# V. Đối tượng Response và Request

# Đối tượng Response và Request

- Xử lý nhập/xuất dữ liệu trên trang Web
- Sử dụng đối tượng Response
- Sử dụng đối tượng Request

Trong năm đối tượng cơ bản của ASP, Response và Request là hai đối tượng thường được sử dụng nhất. Request giúp xác định được các yêu cầu mà web browser yêu cầu trong khi Response là công cụ để thông qua đó, web server trả kết quả thực hiện yêu cầu về cho web browser.

Phần này bắt đầu bằng nhu cầu cần phải xử lý nhập/xuất dữ liệu của một trang web. Tiếp theo chúng ta sẽ đi sâu vào tìm hiểu các cách sử dụng hai đối tượng Response và Request.



# V.1. Xử lý nhập/xuất dữ liệu trên trang Web



Hai đối tượng cơ bản nhất của ASP là Response và Request. Bất kỳ chương trình nào cũng cần phải biết nhận dữ liệu nhập từ phía người dùng và xuất ra kết quả xử lý. ASP sử dụng:

- Đối tượng Request để nhận dữ liệu mà Client (Web browser) gửi về cho Web Server
- Đối tượng Response để ghi kết quả ra trang web trước khi gửi về cho web browser

### ⁴ Ví dụ:

# File chao.htm <BODY> <form name="XinChao" method="Post" Action="chao.asp"> <P>Ten cua ban <INPUT id=text1 name=txtTen></P> <P><INPUT type=submit value=Submit name=submit><INPUT type=reset value=Reset name=reset></P> </form> </BODY> Trang web khi hiển thị:

Ten cua ban Hạnh Dung

Submit Reset

File chao.asp
<% @Language=VBScript %><%
Dim Ten
Ten=Request.Form("txtTen")
Response.Write "Chào, " & ten
%>
Kết quả nhận được khi bạn nhấn nút Submit:
Chào, Hạnh Dung



**Chú ý:** Để chạy thử ví dụ, bạn lưu cả hai file chao.htm và chao.asp vào trong thư mục MyWebApp đã tạo trước đó. Trên IE, gỗ địa chỉ của file chao.htm: //localhost/MyWebApp/chao.htm.

Trước hết, hãy chú ý hai thuộc tính của form XinChao là **method** và **action**:

- action cho biết file .asp nào trong ứng dụng sẽ xử lý kết quả submit của form.
- **method** chỉ định cách Web Browser gửi dữ liệu về cho Web Server. Có hai giá trị cho method là POST và GET.

Ở dạng POST, bạn sẽ dùng đối tượng tương ứng trong Request là Form để truy xuất các giá trị trong form.

Ngược lại, bạn sẽ phải dùng đối tượng QueryString.

Tiếp theo, hãy xét cơ chế xử lý hai nút Submit và Reset của Web Browser

- Khi người dùng chọn nút Submit, Web browser sẽ gửi nội dung trang web hiện có về cho Web Server. Tiếp đó, Web server dựa vào tên file chỉ ra trong thuộc tính action của form để thực hiện xử lý dữ liệu gửi về.
- Khi người dùng chọn nút Reset, toàn bộ các thông tin nhập vào form chứa nút này sẽ bị xoá, form trở về tình trạng ban đầu khi trang web mới hiển thị.

Khi Web server đã xác định file xử lý dữ liệu nhập vào form XinChao là *chao.asp*, nó sẽ tìm và thực hiện các script có trong trang này. Ở đây:

- Ta khai báo một biến **Ten** để lấy giá trị mà người dùng nhập vào TextBox txtTen trên form.
- Do giá trị của thuộc tính method của form là POST, ta dùng đối tượng Form của Request để lấy giá trị trong txtTen.
- Để xuất ra câu: Chào, ... ta dùng đối hành động Write của đối tượng Response.

# Sử dụng cùng một file .asp để nhập liệu và xử lý

Mặc dù bạn hoàn toàn có thể sử dụng 2 trang web tách biệt để nhập liệu và xử lý như trong ví dụ trên, bạn cũng thấy rằng file .asp độc lập rất khó mà có thể trình bày đẹp mắt và tiện dụng. Lý do là với ASP, cách duy nhất để hiển thị trang web là dùng *Response.Write*. Trong Response.Write bạn sẽ phải kết hợp các tag HTML nội dụng muốn hiển thị, ví dụ:

Response.Write "<b>Dòng này được in đậm</b>"

Cách thường được dùng hơn là sử dụng một trang .asp để nhập liệu và xử lý dữ liệu. Bạn chỉ cần phối hợp giữa ASP và HTML một cách linh hoạt để có được một trang ASP giống như một màn hình ứng dụng trên Windows vậy.

### ⁴ Ví dụ:

### File chao.asp

```
<% @Language=VBScript %>
<%
dim Ten
Ten=Request.Form("txtTen")
%>
<HTML>
<BODY>
<form name="XinChao" method="Post" Action="chao.asp">
<P>Ten cua ban <INPUT name=txtTen value="<%=Ten%>"></P>
<P><INPUT type=submit value=Submit name=submit>
<INPUT type=reset value=Reset name=reset></P>
</form>
```



```
<%
if trim(Ten)<>"" then
    Response.Write "Chao, " & Ten
end if
%>
</BODY>
</HTML>
```

Bạn cũng phải để ý trong phần **value** của txtTen sử dụng cách gán giá trị từ ASP sang HTML. Viết <%= Ten%> cũng tương tự như viết:

<%Response.Write Ten%>

# Kết quả:



Dưới đây là một cách kết hợp ASP và HTML khác, ta sử dụng cấu trúc điều khiển If của ASP để định hướng Web Server xử lý đoạn code HTML nào sẽ được trả về cho Web Browser.

# ⁴ Ví dụ:

### File chao.asp

```
<% @Language=VBScript %>
< %
dim Ten
Ten=Request.Form("txtTen")
응>
<HTML>
<BODY>
<%if trim(Ten)<>"" then %>
Chao <%=Ten%>
<% else %>
<form name="XinChao" method="Post" Action="chao.asp">
<P>Ten cua ban <INPUT name=txtTen></P>
<P><INPUT type=submit value=Submit name=submit>
<INPUT type=reset value=Reset name=reset>
</form>
<% end if %>
</BODY>
</HTML>
```



# V.2. Sử dụng đối tượng Response

# Sử dụng đối tượng Response Write table border="1"> %for i=1 to 3 Response.Write " table> Redirec Redirec Response.Redirect "http://www.microsoft.com" %> End Response.End %>

# Hành động Write

Hành động dùng nhiều nhất của Response là **Write**, dùng để in ra một chuỗi trên trang Web.

### Response. Write variant

variant : Dữ liệu sẽ được ghi. Dữ liệu có thể là biến kết hợp với các chuỗi và số nhưng không được chứa chuỗi "%>".

### ⁴ Ví dụ:

```
<% Response.Write "Chào các bạn" %>
<%= Date %> <BR>
```

Ví dụ sau tạo một HTML tag. Chú ý cách dùng dấu \ sau ký tự % của giá trị 100%.

```
<% Response.Write "<TABLE WIDTH = 100%\>" %>
```

Khi bạn dùng Response.Write để in ra một chuỗi *tĩnh* thì cũng giống như bạn dùng HTML. Do đó, hãy cân nhắc để sử dụng **Write** linh hoạt nhất khi có thể. Hãy xem ví dụ sau:

### So sánh:

Ở cách thứ 2, bạn hoàn toàn có thể thiết kế một table trước, định dạng màu, chữ, canh lề,... sau đó, chuyển sang code để chèn thêm các đoạn ASP cần thiết.



# Hành động Redirect

Hành động Redirect gửi một thông điệp tới cho Web Browser, yêu cầu Web Browser truy cập một địa chỉ khác.

### Response.Redirect URL

URL: Uniform Resource Locator mà browser sẽ kết nối tới.

Tất cả các dòng lệnh trong trang asp nằm sau dòng **Redirect** sẽ bị bỏ qua, tuy nhiên, HTTP header vẫn được gửi cùng với một liên kết là URL đã chỉ định. Bằng cách giữ nguyên HTTP header, bạn có thể coi như Redirect sẽ chuyển quyền xử lý yêu cầu của Client từ trang asp hiện hành sang trang khác.

### ⁴ Ví du:

Chuyển kết nối client tới trang web của Microsoft.

```
<% Response.Redirect "http://www.microsoft.com" %>
```

Với các trang web nằm trong ứng dụng của mình, bạn không cần phải ghi địa chị tuyệt đối theo kiểu http://... mà chỉ cần sử dụng địa chỉ tương đối.

# ⁴ Ví dụ:

Nếu trang web của bạn nằm trong thư mục A là thư mục con của MyWebApp, bạn muốn chuyển về trang Login.asp nằm trong MyWebApp:

```
<%Response.Redirect "../login.asp"%>
```

# Hành động End

Hành động End yêu cầu Web server chấm dứt thức hiện các dòng lệnh tiếp theo trong trang asp và trả về các kết quả đã có cho Client.

### Response.End

## ⁴ Ví dụ:

<응

Response.Write "Chấm dứt xử lý trang asp ở đây."

Response.End

Response.Write "Dòng lệnh này không được xử lý."

응>



# V.3. Sử dụng đối tượng Request

# Sử dụng đối tượng Request QueryString http://www.myweb.com/login.asp?username=admin <% Request.QueryString("username") %> Form form name="login" method="POST" ...> input type="text" name="username" value=""> <% Request.Form("username") %>

Khi một Web browser gửi yêu cầu tới Web Server, các thông tin có thể chia làm 4 loại: Form, QueryString, Cookies và Server Variables. Đối tượng Request của ASP cho phép người dùng truy cập tất cả các thông tin này qua những tập hợp có cùng tên như trên. Hai tập hợp thường dùng là Form và QueryString

# QueryString

QueryString là hình thức Client gửi yêu cầu về cho Web Server trực tiếp qua chuỗi URL. Các thông tin gửi về là một tập hợp các cặp **TênBiến=GiáTrị** theo sau chuỗi URL bởi dấu ?.

Ví dụ: <a href="http://www.myweb.com/login.asp?username=admin">http://www.myweb.com/login.asp?username=admin</a>

Trong ví dụ trên, phần QueryString là username=admin. Nếu có nhiều cặp **TênBiến=GiáTrị** thì mỗi cặp phân cách nhau bởi dấu &.

Để lấy giá trị của biến username, trong trang login.asp bạn sử dụng cú pháp: <%Request.QueryString("username")%>

Cú pháp đầy đủ để lấy giá trị của biến là:

### Request.QueryString(Tên biến)[(Index)]

Bạn có thế chú ý thấy mục [(Index)] phía sau. Ở đây, Tên biến là tên một tập hợp và bạn phải dùng đến chỉ sổ để lấy giá trị trong một phần tử của tập hợp đó. Ví dụ cho tên tập hợp là một ListBox cho phép chọn nhiều mục.

# **Form**

QueryString là hình thức đơn giản để gửi yêu cầu cùng các thông tin về cho WebServer. Tuy nhiên hãy tưởng tượng nếu bạn muốn gửi giá trị của một biến Password về cho Web Server. Khi đó, giá trị của Password sẽ hiển thị trên chuỗi URL và không còn "bí mật" gì nữa. đây là một lý do tại sao ta hay sử dụng cách gửi thông tin qua Form.

Với Form có giá trị của thuộc tính **method="POST"**, thông tin sẽ được đóng gói và gửi về cho Web server.

Để truy cập giá trị của các biến, bạn cũng sử dụng cú pháp tương tự như với QueryString:

Request.Form(Tên biến)[(Index)]



# ⁴ Ví dụ:

```
<FORM action = "/scripts/submit.asp" method = "post">
<P>Tên của bạn: <INPUT NAME = "name" SIZE = 48>
<P>Ca sĩ bạn thích nhất: <SELECT NAME = "Singers">
<OPTION>Britney Spear
<OPTION>Mariah Carey
<OPTION>Ricky Martin
<OPTION>Briand Adam</SELECT>
<P><INPUT TYPE = SUBMIT>
</FORM>
```

Khi submit, thông tin gửi tới Web Server có dạng như sau:

```
name=James&singers=Britney+Spear
```

Vì method="post" nên dữ liệu này không đặt lên thanh Address cùng với URL. Trong file submit.asp, bạn có thể viết đoạn mã sau:

```
Chào, <%= Request.Form("name") %>.
Ca sĩ bạn thích nhất là: <%= Request.Form("Singers") %>.
```



# Bài 6

# SỬ DỤNG CÁC ĐỐI TƯỢNG QUẨN LÝ ỨNG DỤNG

# Tóm tắt

Lý thuyết 4 tiết - Thực hành 12 tiết

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài này hướng dẫn cho học viên các sử dụng các đối tượng toàn cục để quản lý ứng dụng và chia sẻ dữ liệu giữa các trang web. Giáo viên cần chú ý phân biệt điểm khác nhau giữa hai đối tượng Application và Session và cách sử dụng các biến dùng chung trong Application.	I. Giới thiệu bài học	6.1 – 6.5	
	II. Đối tượng Session, Application và file Global.asa		
	III. Đối tượng Session		
	IV. Quản lý các Session		
	V. Đối tượng Application		
	VI. File Global.asa		
	VII. Đối tượng Server		
Sau khi hoàn tất bài học này, học viên có thể:			
<ul> <li>Sử dụng các đối tượng Application và Session để quản lý ứng dụng</li> </ul>			
<ul> <li>Phân biệt được sự khác nhau giữa Application và Session</li> </ul>			
<ul> <li>Sử dụng file Global.asa để bẫy các sự kiện của Application và Session.</li> </ul>			



# Giới thiệu bài học

# Nội dung bài học

- Đối tượng Session, Application và file Global.asa
- Quản lý các session
- Đối tượng Server

Ứng dụng web là một ứng dụng phân tán và bạn không chỉ làm việc với một mà nhiều người dùng. Điều này dẫn đến bên cạnh việc thực hiện các xử lý nhập/xuất để đáp ứng các yêu cầu của một người dùng, bạn còn phải quản lý được các người dùng khác nhau cũng như toàn bộ ứng dụng của mình.

ASP bên cạnh hai đối tượng chính Response và Request còn cung cấp các đối tượng để quản lý ứng dụng là Session, Application và Server.



# II. Đối tượng Session, Application và file Global.asa

# ື Đối tượng Session, Application và file Global.asa

- Application là đối tượng toàn cục, quản lý toàn bộ ứng dụng Web
- Session là đối tượng toàn cục, quản lý mỗi người dùng kết nối với ứng dụng Web.
- File Global.asa dùng để khai báo các xử lý cho Application và Session

**Session** và **Application** là hai đối tượng ở mức ứng dụng, giúp các trang ASP của bạn có thể liên kết, trao đổi dữ liệu với nhau, đồng thời, cho phép bạn quản lý các đối tượng lập trình ở mức toàn cục.

Hãy tưởng tượng toàn bộ ứng dụng Web của bạn như một ngân hàng, **Application** là đối tượng quản lý toàn bộ ngân hàng của bạn. Mỗi người dùng tới ngân hàng để giao dịch cần mở tài khoản. Mỗi tài khoản sẽ do một đối tượng **Session** quản lý.

Đương nhiên khách hàng chỉ cần mở tài khoản một lần để thực hiện các giao dịch. Tuy vậy, nếu khách hàng muốn mở nhiều tài khoản thì cũng không có vấn đề gì, ứng với 2, 3, 4,... sẽ có 2, 3, 4,... Session quản lý các tài khoản đó. Điều này cũng giống như một người dùng sử dụng máy tính với nhiều Web Browser cùng một lúc truy cập vào ứng dụng Web vậy.

Nếu không còn muốn giao dịch với ngân hàng nữa, khách hàng có thể rút tiền và đóng tài khoản của mình. Tương tự như vậy với ứng dụng Web, nếu người dùng không làm việc với ứng dụng nữa thì đối tượng **Session** sẽ bị huỷ. Lẽ dĩ nhiên, bạn chẳng thể nào vì một khách hàng đóng tài khoản mà đóng cửa ngân hàng của mình. Vì vậy, đối tượng **Application** vẫn tồn tại khi người dùng chấm dứt kết nối. Nó chỉ bị đóng khi bạn tắt Web Server hay khởi động lại máy.

Do Application và Session là những đối tượng "toàn cục", không thuộc về một trang Web cụ thể nào, ASP sử dụng một file đặc biệt có tên là **Global.asa**. File này được đặt ở thư mục gốc của ứng dụng Web và dùng để quản lý hai đối tượng Application và Session. Trong Global.asa, bạn có quyền khai báo các thủ tục để xử lý những sự kiện của Application và Session. Mỗi đối tượng có hai sự kiện là **OnStart** và **OnEnd**.



# III. Đối tượng Session

# Đối tượng Session

- Sử dụng đối tượng Session
- Biến toàn cục trong Session
- Quản lý các session
- SessionID và Cookies
- Sử dụng Cookies

Session giữ vai trò quản lý người dùng. Sử dụng Session, bạn có thể liên kết các trang web trong ứng dụng với nhau để chia sẽ dữ liệu hay quản lý thời điểm người dùng bắt đầu và kết thúc làm việc. ASP session sử dụng SessionID để quản lý các yêu cầu xử lý mà một người dùng gửi tới web server. Phần này cũng sẽ trình bày với bạn cơ chế gửi nhận SessionID thông qua Cookies giữa web server và web browser.



# III.1. Sử dụng đối tượng Session

# Sử dụng đối tượng Session

- Timeout: Quy định khoảng thời gian tối đa duy trì đối tượng Session
- Abandon: Chủ động huỷ bỏ đối tượng Session

Đối tượng Session sẽ được Web server tạo ra khi một người dùng kết nối đến Web server lần đầu tiên để truy cập một trang ASP. Sau khi được tạo ra, Session được duy trì trên Web Server trong suốt quá trình người dùng sử dụng ứng dụng Web. Chừng nào Web Browser còn gửi các yêu cầu và nhận các kết quả từ Web Server mà ta còn gọi là "tương tác" với Web server thì Session còn được duy trì.

### **Timeout**

Thuộc tính Timeout xác định khoảng thời gian tính bằng phút mà ứng dụng sẽ duy trì đối tượng Session nếu người dùng không tương tác với server.

### Session.Timeout = 10

Nếu sau 10 phút kể từ lần tương tác cuối cùng giữa Web Browser và Web Server mà Web Browser không gửi yêu cầu nữa thì Web Server sẽ tự động huỷ bỏ Session. Kể từ sau đó, các yêu cầu gửi về từ Web Browser sẽ được coi như một người dùng mới và được lập một Session mới.

Măc đinh, Web Server đăt Timeout=20.

Đối tượng Session rất quan trọng, nó bổ sung cho người lập trình một công cụ đắc lực khi viết ứng dụng. Ứng dụng Web sử dụng giao thức HTTP là giao thức "phi trạng thái". Server hoàn toàn không ghi nhớ gì về Client mà nó vừa làm việc trước đó. Nếu không có Session thì giữa hai lần gửi yêu cầu liên tiếp của cùng một Client đến cho Sever, Server sẽ coi Client này là hai Client khác nhau, nó hoàn toàn không biết lần yêu cầu trước đó, Client đã gửi yêu cầu gì!

Bạn hãy tưởng tượng mình bước vào một nhà hàng, gọi món ăn, sau khi bạn thưởng thức món ăn và gọi tính tiền, anh chàng bồi bàn chẳng thể nhớ trước đó bạn đã gọi món gì!



# **Abandon**

Hành động Abandon sẽ huỷ Session và giải phóng tài nguyên cho Web server. Nếu bạn không thực hiện hành động Abandon, server sẽ tự động huỷ các dữ liệu vào thời điểm Timeout.

### Session.Abandon

Khi Session bị huỷ, mọi thông tin sẽ không được huỷ ngay lập tức cho tới khi tất cả các đoạn script trong trang chứa lời gọi Abandon đã được thực hiện xong. Ví dụ sau đây sẽ xuất ra chuỗi "Đang chờ được xoá".

# ⁴ Ví dụ

```
Session.Abandon
Session("X") = "Dang cho duoc xoa"
Reponse.Write(Session("X"))
```

Server sẽ tạo ra một Session mới sau khi hành động Abandon được thi hành nếu người dùng vẫn còn tương tác với server.



# III.2. Biến toàn cục trong Session

# Biến toàn cục trong Session

- Biến toàn cục là biến mà tất cả các trang ASP trong ứng dụng đều có thể truy cập được.
- Với biến toàn cục ở mức độ Session thì với mỗi người dùng, biến sẽ có một giá tri khác nhau.
- Khai báo và sử dụng một biến toàn cục: Session("Tên biến")

Trong ví dụ trên, bạn chú ý thấy cách dùng biến X: **Session("X")**. Ở đây, X là một biến mà bạn đã đặt nó vào trong Session để biến nó thành một biến toàn cục. Lúc này, một trang web khác có thể truy cập tới biến X để đọc hay thay đổi giá trị.

Biến toàn cục hay dùng nhất trong Session là Username. Khi người dùng mới đăng nhập thành công vào ứng dụng, bạn sẽ lưu ngay Username của người đó vào Session:

Session("Username") = Request. Form("txtUsername")

Bằng cách này, bạn sẽ kiểm tra một người dùng đã đăng nhập trước khi sử dụng ứng dụng hay chưa dựa vào việc kiểm tra biến Username. Nếu giá trị của nó là "" thì định hướng người đó đến trang Login:

```
If Session("Username")="" Then
    Response.Redirect("login.asp")
```

End if

Dòng ví dụ trên rất thường được viết ở đầu mỗi trang ASP. Nó đảm bảo cho ứng dụng của bạn không thể sử dụng được bởi những người không được cấp quyền sử dụng.

Bạn cũng cần phân biệt về mức độ toàn cục của biến. Trong ứng dụng web, có tới hai loại biến toàn cục: Biến toàn cục ở mức Session và biến toàn cục ở mức Application.

- Ó mức Session, biến là toàn cục đối với người dùng. Mỗi trang web đều có thể sử dụng biến nhưng với mỗi người dùng, biến sẽ có giá trị khác nhau.
- Ở múc Application, mỗi trang web cũng đều có thể sử dụng biến nhưng cho dù người dùng là ai, giá trị của biến cũng đều như nhau. Chỉ cần một người thay đổi giá trị của biến, tất cả những người dùng khác đều bị ảnh hưởng theo!
- Xét về số lượng, với mỗi người dùng, một biến Session được tạo ra dành riêng cho người đó, tuy nhiên, tất cả mọi kết nối tới ứng dụng đều dùng chung một biến toàn cục ở mức Application.



### SessionID

SessionID là một số nguyên Long mà Web server gán cho mỗi session vào lúc bắt đầu khởi tạo. Mỗi session có một ID duy nhất và được Web server duy trì trong suốt quá trình session tồn tại. Khi session bị huỷ, số ID của nó sẽ bị huỷ theo, Web server sẽ không dùng ID này để cấp cho session khác.

Để lấy giá trị SessionID:

### Session.SessionID

Với SessionID, bạn xác định mỗi người dùng là duy nhất. Bạn không thể dùng Username và Password của người dùng vào việc này. Tưởng tượng bạn đã tổ chức một CSDL để quản lý các tài khoản đăng nhập vào ứng dụng của mình. Trong bảng NguoiDung, cột Username làm khoá chính (hoặc dùng ràng buộc Unique) đảm bảo mỗi người dùng là duy nhất. Tuy vậy, khi một người được cấp một tài khoản, người đó có thể mở nhiều trình duyệt và đăng nhập với cùng một Username, password vào Web server. Rõ ràng lúc này, bạn có nhiều biến Username, Password giống nhau trong các session khác nhau.

Một điểm yếu của ASP là nó sử dụng cơ chế Cookies để quản lý các SessionID. Trong quá trình tương tác giữa Web server và client, hai máy tính luôn gửi SessionID qua lại cho nhau. Tại server, SessionID của mỗi Client được quản lý bởi Web server. Tại client, web browser sử dụng cơ chế Cookies để lưu lại SessionID.

Một số Web browser không hỗ trợ Cookies trong khi một số khác như Internet Explorer lại cho phép bạn bật hay tắt chế độ sử dụng Cookies. Một khi Web browser không dùng Cookies, ASP sẽ không còn quản lý Session được nữa.



# IV. Quản lý các Session

# Quản lý các session

- SessionID được Web server tạo ra và lưu tại Client bằng cookie
- Session có trạng thái là Session có lưu trữ dữ liệu (biến). Web server chỉ quản lý các Session có trạng thái
- Session phi trạng thái được Server tạo ra khi nhận yêu cầu và huỷ ngay khi xử lý xong trang ASP

Một ứng dụng Web được viết tốt phải quản lý hiệu quả các thông tin của người dùng lưu trong Session khi người dùng di chuyển giữa vác trang trong ứng dụng. HTTP là một giao thức phi trạng thái, nghĩa là Web server sẽ coi mỗi yêu cầu được gửi tới cho một trang như một yêu cầu độc lập, server sẽ không lưu lại bất cứ thông tin nào của những yêu cầu từ Client mà nó đã xử lý.

ASP cung cấp đối tượng Session giúp bạn quản lý các thông tin của người dùng. Tuy nhiên, ID mà ứng dụng Web của bạn quản lý được lưu dưới dạng một cookie (một file có kích thước nhỏ được lưu tại máy tính chạy trình duyệt), do đó nếu người dùng không có trình duyệt hỗ trợ cookie hoặc người dùng không cho phép trình duyệt sử dụng cookie thì Session ID không thể sử dụng được.

# Khởi đầu và kết thúc một Session

Như ban đã biết, một Session có thể bắt đầu bằng 1 trong 4 cách:

- Một người dùng mới yêu cầu một trang asp trong ứng dụng và file Global.asa của ứng dụng đó có chứa thủ tục cho sự kiện Session\_OnStart.
- Một người dùng lưu một giá trị dữ liệu nào đó vào Session.
- Một Session mới được tự động khởi tạo khi Web server nhận một yêu cầu mà không chứa một Session ID hợp lệ
- Người dùng yêu cầu một file .asp trong ứng dụng và file Global.asa của ứng dụng sử dụng tag <OBJECT> để khởi tạo một đối tượng có phạm vi Session.

Một Session sẽ tự động kết thúc nếu người dùng không có yêu cầu gửi tới Web server trong một khoảng thời gian là Session.Timeout

**Chú ý:** TimeOut chỉ có ý nghĩa với các Session có trạng thái. Session phi trạng thái (Session không chứa dữ liệu) sẽ chấm dứt ngay khi yêu cầu được xử lý và được khởi tạo lại khi yêu cầu được gửi tới sau đó phát xuất từ cùng trình duyệt.



# IV.1. SessionID và Cookies

# SessionID và Cookies

- Một SessionID khi được tạo ra sẽ được gửi qua lại giữa Server và Client mỗi lần tương tác.
- Giá trị của SessionID được gửi dưới dạng một cookie. Server sử dụng cookie này để truy cập đúng dữ liệu trong Session được yêu cầu
- Một trang ASP có thể đặt ra ngoài Session bằng thuộc tính EnableSessionState

Khi người dùng lần đầu tiên yêu cầu một file .asp trong một ứng dụng, Web server tạo ra một Session ID duy nhất dùng để phân biệt các Session với nhau. Khi Session được khởi tạo, Web server sẽ lưu giá trị Session ID tại trình duyệt của người dùng như một cookie.

Session ID làm việc như một chìa khoá. Các dữ liệu của người dùng được lưu giữ trong một "tủ" khi người dùng tương tác với ứng dụng. Session ID cookie sẽ được chuyển tới Web server trong HTTP request header cho phép mở tủ để lấy thông tin. Mỗi lần Web server nhận một yêu cầu cho một trang, nó sẽ kiểm tra HTTP request header để lấy giá trị Session ID.

Sau khi đã lưu trữ giá trị Session ID tại trình duyệt của người dùng, Web server sẽ sử dụng cookie này để kiểm tra Session, kể cả khi người dùng yêu cầu một trang ASP của một ứng dụng khác. Tương tự, nếu người dùng để cho Session bị TimeOut và sau đó gửi yêu cầu tới Web server yêu cầu một trang ASP, Web server tạo ra một session mới với cùng giá trị Session ID. Người dùng chỉ có thể nhận được giá trị Session ID mới khi Web server bị khởi động lại hoặc khi người dùng khởi động lại trình duyệt.

Bạn có thể ngăn Web server kiểm soát Session và gửi Session ID cookie tới trình duyệt của người dùng. Web server sẽ không gửi Session ID cookie trong trường hợp:

- Úng dụng không kiểm soát trạng thái của Session (không lưu biến vào Session)
- Trang ASP được chỉ định là phi trạng thái. (trang ASP có chứa đoạn script:

< @ EnableSessionState=False % > )



# IV.2. Sử dụng Cookies

# Sử dụng Cookies

- Sử dụng tập hợp Cookies của Request hay Respone để đọc hay ghi giá trị cho một cookie.
- Cookie được gửi tới Server tuỳ theo đường dẫn của file ASP

Cookies("Tên Cookie")=giá trị

Cookies.Path = đường dẫn của file sẽ gửi cookie tới

Cookies.Expires=Thời điểm sẽ xoá cookie trên Client

Bạn không nên coi Session ID là một công cụ để kiểm tra người dùng trong nhiều lần người dùng truy cập vào Website của mình. Giá trị Session ID lưu trong vùng nhớ của Web server rất dễ bị mất vì nhiều lý do. Chẳng hạn, khi file Global.asa được cập nhật, Web server tạm ngừng hoạt động và huỷ tất cả các session để khởi động lại.

Nếu muốn kiểm soát người dùng trong nhiều lần truy cập khác nhau tới Website của mình, bạn cần tạo một cookie riêng lưu tại trình duyệt của người dùng và lưu cookie đó trong cơ sở dữ liệu của ứng dụng.

# Định nghĩa

Một cookie là một mẩu thông tin ngắn mà Web server lưu trữ tại trình duyệt của người dùng như **một file**. Cookie cho phép bạn đính kèm một số thông tin cần thiết với một người dùng cụ thể. ASP có thể nhận hay đặt giá trị cho các cookie bằng cách sử dụng tập hợp **Cookies** của đối tượng Response và đối tượng Request.

Cú pháp:

Request.Cookies("Tên cookie")

Response.Cookies("Tên cookie")

Mỗi một cookie bao gồm:

- Giá trị của Cookie
- Thuôc tính Path: Đường dẫn tới thư muc chứa file ASP sẽ nhân được cookie
- Expires: Thời điểm file chứa cookie bị xoá trên máy Client

# Đường dẫn của Cookies

Mỗi cookie được lưu bởi ASP tại trình duyệt của người dùng sẽ có một đường dẫn. Khi yêu cầu một file lưu tại đường dẫn giống với đường dẫn trong cookie, trình duyệt sẽ tự động gửi giá trị của cookie tới Web server. Mặc định, đường dẫn là tên thư mục ứng dụng chứa file ASP đã tao ra cookie đó.



# ⁴ Ví dụ:

Đoạn mã sau đặt đường dẫn SalesApp/Customer/Profiles/ cho một cookie có tên là Purchases:

```
Response.Cookies("Purchases") = "12"
Response.Cookies("Purchases").Expires = "1/1/2003"
Response.Cookies("Purchases").Path =
"/SalesApp/Customer/Profiles/"
%>
```

Bất cứ khi nào trình duyệt yêu cầu một file lưu trong thư mục SalesApp/Customer/Profiles/ hoặc thư mục con của nó, trình duyệt sẽ gửi cookie Purchases tới Web server.

# Chú ý:

- Một số trình duyệt phân biệt chữ HOA/thường trong đường dẫn của cookie. Điều này dẫn đến việc nếu bạn đổi tên hay cấu trúc thư mục trên Web server thì nhiều khả năng cookie không còn có ý nghĩa gì nữa.
- Nếu đường dẫn của cookie được đặt bằng "/" thì trình duyệt sẽ gửi cookie tới Web server khi người dùng yêu cầu bất cứ file nào. Bạn đừng lợi dụng đặc điểm này vì cookie chiếm một số byte không nhỏ và sẽ làm cho thời gian hồi đáp giữa Web server và client bị chậm lại.

Tổ chức của Cookie khá phức tạp, nó có thể là một tập hợp nhiều **Tên=Giá trị** và được gọi là Dictionary. Web server cũng cho phép nhiều Cookie có cùng tên và các đường dẫn chồng chéo lên nhau. Chúng tôi không muốn trình bày chi tiết về Cookie vì bởi chức năng khá hữu dụng, nhiều người đã lợi dụng nó để dùng trong các hành động nguy hiểm dẫn đến các Web Browser cho phép người dùng tắt cơ chế gửi/nhận cookie.

Cách thông thường hơn được nhiều người sử dụng để thay thế cho Cookies là dùng các file Hidden trên form. Với Hidden field, bạn có thể quản lý trạng thái của từng trang ASP và gửi/nhận các thông tin do bản thân chương trình tính toán, phát sinh ra giá trị. Điều duy nhất Hidden field không thể làm nổi là ghi nhớ thông tin người dùng khi Client và Server đã chấm dứt tương tác. Hãy nhớ đến các Website vẫn lưu lại username và password của bạn để tự động đăng nhập mỗi lần bạn truy cập tới trang web của họ. Cơ chế này chỉ có cookie mới có khả năng thực hiện.



# V. Đối tượng Application

# Đối tượng Application

- Chỉ có một đối tượng Application duy nhất cho toàn bộ ứng dụng Web
- Biến đặt trong Application được chia sẽ cho tất cả các Session đang có trên Web server
- Khi muốn truy cập các biến trong Application, sử dụng cấu trúc Lock và Unlock.

Đối tượng Application là đối tượng quản lý toàn bộ ứng dụng Web của bạn. Sau khi Web server đã khởi động, người dùng đầu tiên khi truy cập một trang ASP trong ứng dụng sẽ khởi tạo đối tượng Application. Đối tượng này sẽ được server duy trì cho tới khi bạn khởi động lại Web server hay tắt máy.

Vì tất cả mọi người dùng đều dùng chung một Application, đây là nơi để đặt các biến toàn cục ở mức cao nhất. Những biến toàn cục đặt trong Application sẽ dùng chung cho tất cả các trang web và cho tất cả các Session hiện có trên Web server.

Để tạo một biến toàn cục trong Application, ta cũng dùng cách tương tự như với Session:

Application ("Tên biến")

# ⁴ Ví du:

Application("SoLuotTruyCap")=1

Biến SoLuotTruyCap ở trên sẽ được dùng để cho biết đã có bao nhiều người dùng truy cập đến Website của bạn. Với việc gán SoLuotTruyCap=1, dòng lệnh này nên đặt tại thời điểm Application khởi động mà thôi.

**Mở rộng:** Bạn sẽ tìm hiểu thời điểm Application và Session khởi động hay kết thúc trong phần chúng tôi nói về file Global.asa tiếp theo đây.

# Hành động Lock và Unlock

Lock, Unlock là cơ chế khoá-mở được dùng rất nhiều trong lập trình khi muốn sử dụng một tài nguyên dùng chung. Khi có một biến mà nhiều đối tượng đều có thể truy cập, tại một thời điểm, biến đó có thể được thay đổi giá trị bởi nhiều đối tượng sử dụng nó. Bằng cách sử dụng Lock và Unlock, hệ thống đảm bảo rằng giá trị của biến không thể bị sai sót hay không dự đoán trước được.



Để sử dụng biến, đối tượng sẽ gọi hành động Lock của Application. Sau hành động này, Application chỉ còn cho phép duy nhất đối tượng vừa gọi hành động được quyền đọc hay thay đổi giá trị của các biến có trong Application. Khi thực hiện xong, đối tượng gọi hành động Unlock, Application sẽ "giải phóng" các biến và cho phép đối tượng khác sử dụng.

Cú pháp của Lock và Unlock:

Application.Lock
Application.Unlock

# ⁴ Ví dụ:

# Nếu bạn quên gọi Unlock?

Điều gì sẽ xảy ra nếu bạn gọi Application.Lock, truy cập các biến trong Application nhưng không gọi Unlock? Web server sẽ tự động giải phóng Application cho bạn khi trang ASP kết thúc xử lý. Đây là cơ chế của Web server, tuy nhiên, bạn nên cố gắng nhớ thực hiện hành động Unlock vì nếu như trang ASP của bạn cần nhiều thời gian để xử lý, tính toán thống kê thì tất cả những trang ASP khác cũng sẽ phải đứng chờ xử lý.



# VI. File Global.asa

# File Global.asa

 Dùng để quản lý các đối tượng toàn cục trong ứng dụng Web

<SCRIPT LANGUAGE=VBScript RUNAT=Server>

Sub Session\_OnStart session("UserID")=0 End Sub

'Sub Session\_OnEnd
'End Sub
'Application\_OnStart
'Application\_OnEnd
</SCRIPT>

File Global.asa được dùng để quản lý các đối tượng toàn cục trong ứng dụng Web. Bạn có thể dùng hay không dùng file này nhưng nếu có, nó phải được đặt ở thư mục gốc của ứng dụng.

File Global.asa chỉ có thể chứa:

- Các sự kiện của đối tượng Application
- Các sự kiện của đối tượng Session
- Các khai báo <OBJECT>
- Khai báo các TypeLibrary

**Chú ý:** Chỉ đề cập đến việc quản lý các sự kiện của đối tượng Application và Session ở đây, cách khai báo OBJECT, TypeLibrary và các tập hợp Contents, StaticObjects sẽ được nói tới trong phần phụ lục của giáo trình



# VI.1. Các sự kiện của đối tượng Application

# Các sự kiện của đối tượng Application

- Úng dụng có hai sự kiện lúc khởi động và kết thúc:
  - Application\_OnStart: Khi người dùng đầu tiên truy cập trang web
  - Application OnEnd: Khi Web server tắt
- Các sự kiện này có thể được khai báo xử lý trong file Global.asa

Một ứng dụng ASP được tạo bởi các file trong thư mục gốc của ứng dụng và các thư mục con. Ứng dụng bắt đầu khởi động khi có người dùng đầu tiên yêu cầu một trang trong ứng dụng và kết thúc khi server tắt.

Đối tượng Application dùng để quản lý ứng dụng có hai sự kiện:

- Application\_OnStart
- Application\_OnEnd

Bạn có thể chỉ định các script cho các sự kiện này trong file Global.asa. Khi ứng dụng khởi động, server duyệt trong file này và thực hiện các đoạn script dành cho sự kiện Application\_OnStart. Khi ứng dụng kết thúc, server sẽ thực hiện đoạn script cho sự kiện Application\_OnEnd.

# Application\_OnStart

Sự kiện Application\_OnStart xảy ra trước khi session mới đầu tiên được tạo. Chỉ có đối tượng Application và Server là có thể sử dụng trong sự kiện Application\_OnStart.

### Cú pháp:

<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Application\_OnStart
End Sub
</SCRIPT>

# ⁴ Ví dụ:

Sub Application\_OnStart
 Application("NumberofVisitors") = 0
End Sub

Trong ví dụ này, ứng dụng khởi động biến NumberofVisitors để đếm số người dùng hiện đang "online". Bạn cũng nên chú ý rằng trong file Global.asa, các thủ tục phải đặt trong tag <SCRIPT>.



# Application\_OnEnd

Sự kiện Application\_OnEnd xảy ra khi ứng dụng kết thúc, sau sự kiện Session\_OnEnd cuối cùng. Chỉ có đối tượng Application và Server là có thể sử dụng.

Cú pháp

<SCRIPT LANGUAGE=ScriptLanguage RUNAT=Server>
Sub Application\_OnEnd

End Sub

Bạn không thể gọi hành động MapPath của đối tượng Server trong sự kiện này.



# VI.2. Các sự kiện của đối tượng Session

# Các sự kiện của đối tượng Session

- Session có hai sự kiện lúc khởi động và kết thúc:
  - Session\_OnStart: Khi người dùng truy cập trang web lần đầu tiên
  - Session\_OnEnd: Khi server kết thúc Session
- Các sự kiện này có thể được khai báo xử lý trong file Global.asa

Web server tự động tạo ra một Session mỗi khi có một người dùng lần đầu tiên gửi yêu cầu tới một trang trong ứng dụng. Server sẽ huỷ session khi hành động Abandon xảy ra hoặc khi session hết thời gian duy trì SessionTimeout mà không còn tương tác với người dùng nữa.

Session có 2 sự kiện:

- Session\_OnStart
- Session\_OnEnd

Các script dành cho sự kiện này nếu có được đặt trong file Global.asa. Sự kiện Session\_OnStart xảy ra khi session được tạo, ngược lại sự kiện Session\_OnEnd xảy ra khi session bị huỷ.

Sự kiện Session\_OnStart xảy ra khi server tạo một session mới. Server sẽ thi hành đoạn script dành cho sự kiện trong file Global.asa trước khi xử lý các yêu cầu do trình duyệt gửi tới. Tất cả các đối tượng cơ sở đều có thể sử dụng trong đoạn script dành cho sự kiện Session\_OnStart.

# Session\_OnStart

Cú pháp

<SCRIPT LANGUAGE=ScriptLanguage RUNAT=Server>
Sub Session\_OnStart

End Sub

</SCRIPT>

Bạn thường dùng hành động Response.Redirect trong sự kiện Session\_OnStart để đưa người dùng đến một trang nhất định mỗi khi họ bắt đầu truy cập.



```
<SCRIPT RUNAT=Server LANGUAGE=VBScript>
Sub Session_OnStart

startPage = "/MyApp/KhoiDau.asp"
currentPage = Request.ServerVariables("SCRIPT_NAME")
```

Response.Redirect(startPage)

If strcomp(currentPage, startPage, 1) Then

End If

<sup>↑</sup> Ví du:

End Sub

</SCRIPT>

Bạn để ý thấy trong ví dụ có sử dụng một biến có tên là SCRIPT\_NAME và tập hợp ServerVaraibles. Đây là một tập hợp khác của đối tượng Request bao gồm những biến cho biết thông tin về Web browser mà người dùng đang sử dụng. Bạn có thể xem phần này ở phụ lục.

Ví dụ trên chỉ làm trong trường hợp trình duyệt hỗ trợ Cookies. Ngược lại, trình duyệt sẽ không gửi SessionID tới cho server và server sẽ tạo một session mới mỗi khi trình duyệt gửi một yêu cầu, dẫn đến việc sự kiện Session\_OnStart được thực hiên.

Thông thường, ứng dụng của bạn nên thông báo cho người dùng biết việc sử dụng phải sử dụng Cookies ở trang khởi đầu.

# Session\_OnEnd

Sự kiện Session\_OnEnd xảy ra khi một session bị huỷ do hành động Abandon hoặc do server khi tới thời điểm SessionTimeout. Chỉ có đối tượng Application, Server và Session là có thể truy cập trong đoạn script dành cho sự kiện này.

### Cú pháp:

```
<SCRIPT LANGUAGE=ScriptLanguage RUNAT=Server>
Sub Session_OnEnd
. . .
End Sub
</SCRIPT>
```

Bạn không thể thực thi hành động MapPath của đối tượng Server trong sự kiện Session\_OnEnd.



# VII. Đối tượng Server

# Đối tượng Server Sử dụng đối tượng Server Hành động MapPath Hành động Transfer

Khác với Application và Session dùng để quản lý ứng dụng và người dùng, đối tượng Server cho phép bạn quản lý web server đang chạy ứng dụng web của mình.



#### VII.1. Sử dụng đối tượng Server

#### Sử dụng đối tượng Server

- ScriptTimeout: Khoảng thời gian tối đa thi hành một file ASP
- Execute: Thi hành một file ASP khác như một thủ tuc

Đối tượng Server cung cấp và cho phép bạn chỉnh sửa các thông tin quản lý Web Server. Nó cũng là đối tượng toàn cục hữu ích khi cần tương tác giữa các trang web với nhau.

#### ScriptTimeout

Thuộc tính ScriptTimeout xác định thời gian tối đa một đoạn script có thể thi hành mà không bị dừng lại bởi Web Server.

Cú pháp

Server.ScriptTimeout = Số giây (mặc định là 90)

#### ⁴ Ví dụ:

<% Server.ScriptTimeout = 100 %>

Tính năng quản lý Script timeout của Web server đảm bảo không có đoạn script nào trong trang ASP của bạn bị lặp quá lâu và chiếm dữ tài nguyên trên server dẫn tới các người dùng khác không được đáp ứng. Nếu trang ASP của bạn cần nhiều hơn 90 giây để xử lý, bạn có thể đặt lại giá trị cho ScriptTimeout, ngược lại, nên giữ nguyên giá trị này.

#### Execute

Hành động Excute gọi thi hành một file asp và xử lý file đó như một phần của đoạn script đã thi hành lệnh Excute. Sau khi thi hành file asp được yêu cầu, kết quả được trả về cho đoạn script cũ.

Cú pháp

Server. Execute ( Đường dẫn tới file ASP muốn thi hành )

Hành động Excute giúp phân chia một ứng dụng phức tạp thành nhiều phần nhỏ độc lập. Bạn có thể tạo sẵn một thư việc các file asp để gọi thi hành khi cần.

Khi quyền thi hành được chuyển tới file ASP mới, những thông tin sau trên file ASP gọi sẽ được gửi kèm:

- Các biến trong Application, Session.
- Đối tượng Request, Response và Server.



#### ⁴ Ví dụ:

```
Welcome.asp
<HTML>
<BODY>
< %
AcceptLang = Request.ServerVariables("HTTP_ACCEPT_LANGUAGE")
Lang = Left(AcceptLang, 2)
Server.Execute(Lang & "Welcome.asp")
응>
</BODY>
</HTML>
enWelcome.asp
<% Response.Write "Welcome to my website!" %>
deWelcome.asp
<% Response.Write "Willkommen zu meinem website!" %>
esWelcome.asp
<% Response.Write "Recepción a mi website!" %>
```

Kết quả trả về phục thuộc vào ngôn ngữ của hệ thống mà Web server đang chạy.

Tiếng Anh (US English):

Welcome to my website!

Tiếng Đức:

Willkommen zu meinem website!

Tiếng Tây Ban Nha:

Recepción a mi website!



#### VII.2. MapPath

#### MapPath

- MapPath ánh xạ đường dẫn cho trước thành một đường dẫn vật lý dựa vào thư mục gốc của ứng dụng.
- MapPath không kiểm tra sự tồn tại của đường dẫn được cho
- MapPath("\abc")=MapPath("\") & "abc"

Hành động MapPath ánh xạ một đường dẫn tương đối hoặc đường dẫn ảo thành một đường dẫn vật lý trên server.

Cú pháp

#### Server.MapPath( Path )

Nếu Path bắt đầu bằng (/) hoặc (\), kết quả trả về là đường dẫn ảo đầy đủ. Nếu không, kết quả trả về là đường dẫn tương đối so với thư mục chứa file asp gọi hành động MapPath.

#### ⁴ Ví du:

Giả sử C:\InetPub\wwwroot là thư mục gốc của server. File test.asp và data.txt đặt trong C:\InetPub\WWWroot\Script.

```
File test.asp
<%= Server.MapPath("\")%><BR>
<%= Server.MapPath("xyz")%>
```

#### Kết quả:

```
c:\inetpub\wwwroot
c:\inetpub\wwwroot\script\xyz<BR>
```

Bạn có thể đọc đâu đó các hướng dẫn tỉ mỉ và phức tạp về cách sử dụng hành động MapPath. Thực ra, MapPath rất đơn giản, chỉ cần ghi nhớ rằng:

- Dùng dấu \ (/) ở đầu, MapPath trả về đường dẫn đến thư mục gốc nối với chuỗi theo sau.
- Nếu không có dấu \ (/) ở đầu chuỗi, MapPath trả về đường dẫn đến thư mục chứa file ASP nối với chuỗi theo sau.



#### VII.3. Transfer

#### Transfer

- Hành động Transfer dùng để chuyển quyền xử lý từ file ASP gọi đến một file ASP khác được chỉ ra trong đường dẫn
- Toàn bộ các thông tin, dữ liệu trong các đối tượng Server, Response và Request đều được gửi kèm theo
- Server. Transfer chỉ được hỗ trợ bởi IIS 5 trở về sau

Hành động Transfer gửi tất cả dữ liệu và thông tin trạng thái của các đối tượng cơ sở gắn với file asp đang thi hành tới một file asp khác được chỉ định để xử lý.

Cú pháp

#### Server.Transfer (path)

Nếu file asp muốn chuyển tới lại là của một ứng dụng khác, file đó sẽ được thi hành như là một file thuộc về ứng dụng đang thực hiện.

Hành động Transfer hoàn toàn có thể thay thế cho hành động *Response.Redirect.* Redirect yêu cầu trình duyệt gửi yêu cầu tới một trang khác, do đó dữ liệu sẽ phải gửi lại một lần nữa giữa server và trình duyệt. Tuy nhiên, chỉ từ IIS 5.0 ta mới có thể gọi hành động Transfer.

#### ⁴ Ví dụ:

```
ASP1.asp
<%
Response.Write Session.SessionID
Response.Write ("<BR>")
Response.Write("chuyen toi file ASP2 <BR>")
Server.Transfer("ASP2.asp")
%>
ASP2.asp
<% Response.Write Session.SessionID %>
```



# Bài 7

# SỬ DỤNG ADODB: KẾT NỐI VÀ HIỂN THỊ DỮ LIỆU

# Tóm tắt

Lý thuyết 4 tiết - Thực hành 12 tiết

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Nội dung của bài học này tập trung vào trình bày các kỹ thuật để kết nối với CSDL sử dụng trong thư viện ADO. Giáo viên nên cố gắng trình bày các vấn đề liên quan đến việc bảo mật CSDL của ứng dụng.  Trong việc trình bày các kỹ thuật hiển thị dữ liệu từ RecordSet, giáo viên chú trọng vào kỹ thuật duyệt RecordSet và hiển thị các giá trị của các field trên table.  Sau khi hoàn tất bài học này, học viên có thể:  Thực hiện các hình thức kết nối đế CSDL bằng DSN và ConnectionString  Nử dụng RecordSet để truy cập dữ liệu  Duyệt và hiển thị dữ liệu trong RecordSet.	<ul> <li>I. Giới thiệu bài học</li> <li>II. Thư viên ADO</li> <li>III. Kết nối với CSDL bằng ADODB</li> <li>IV. Truy cập và hiển thị dữ liệu</li> </ul>	7.1 – 7.6	



# I. Giới thiệu bài học

# Nội dung bài học Thư viện ADO Kết nối với cơ sở dữ liệu Truy cập dữ liệu bằng RecordSet

Một ứng dụng bất kỳ thường không thể thiếu nguồn dữ liệu. Ưu thế mạnh nhất cửa Server Script là cho phép người lập trình khả năng lưu trữ và truy cập cơ sở dữ liệu trên file hay trên database.

Trong bài này, chúng tôi sẽ trình bày các khái niệm về thư viện ADODB, các cách mà bạn có thể sử dụng để kết nối vào CSDL và cuối cùng là kỹ thuật sử dụng đối tượng RecordSet của ADODB để truy cập và hiển thị dữ liệu.



#### II. Thư viên ADO

#### Thư viện ADO

- ADO là bộ thư viện để xử lý dữ liệu của Microsoft, đi kèm với hê điều hành Windows
- ADO sử dụng kỹ thuật OLE DB và cung cấp cho người dùng các đối tượng để thao tác với dữ liệu
- Windows 2000, XP cài đặt sẵn ADO và IIS thích hợp để xây dựng các ứng dụng Web chuyên nghiệp

ADO (Active Data Objects) là một thành phần của MDAC (Microsfot Data Access Component), một công cụ được sử dụng để tiến hành các thao tác xử lý dữ liệu với nhiều hệ quản trị cơ sở dữ liệu khác nhau và nhiều loại dữ liệu, phổ biến nhất là cơ sở dữ liệu quan hệ (relational database).

ADO sử dụng kỹ thuật OLE DB để làm việc với các cơ sở dữ liệu. Thông qua OLE DB, ADO có thể làm việc với các CSDL quan hệ như Access, SQL Server, Oracle,... hay các cơ sở dữ liệu phi quan hệ như cấu trúc thư mục, mail,... Điểm đặc biệt của ADO là bộ thư viện này cung cấp các **đối tượng** lập trình thay vì bắt người dùng phải gọi các hàm của OLE DB.

ADO gồm nhiều phiên bản khác nhau, phiên bản mới nhất hiện nay là ADO 2.7 và phiên bản dành cho công nghệ .NET của Microsoft – ADO.NET. Để có thể sử dụng ADO truy cập cơ sở dữ liệu với ASP, bạn phải cài đặt bộ thư viện ADO vào hệ thống, tuy nhiên việc cài đặt ADO thường đã được thực hiện khi bạn cài đặt các phiên bản của hệ điều hành Windows hay một số phần mềm của Microsoft:

- IIS 4.0/Personal Web Server 4.0 (có trong WinNT option pack 4) có ASP 2.0 và ADO 1.5
- Windows 98 cũng bao gồm PW 4.0 (Personal Web server, ASP 2.0 và ADO 1.5
- Microsoft Visual Studio 6 có sẵn ADO 2.0
- Microsoft Office 2000 và Internet Explorer (IE) 5.0 đi kèm với ADO 2.1
- Windows 2000 đi kèm với IIS 5.0, ASP 3.0 và ADO 2.5
- Windows XP cung cấp ADO 2.7 và IIS 5.1 cùng với IE 6.0

Nếu bạn muốn xây dựng các website chuyên dùng, nên sử dụng các phiên bản của Windows Server Farmily (phiên bản mới nhất là 2003).



# III. Kết nối với CSDL bằng ADODB

# Kết nối với CSDL bằng ADODB

- Cơ sở dữ liệu
- Thông tin kết nối đến CSDL
- Tạo lập DSN
- Kết nối đến CSDL bằng đối tượng ADODB.Connection
- Tạo kết nối sử dụng file đính kèm

Kết nối đến CSDL là công việc đầu tiên bạn cần thực hiện khi làm việc với dữ liệu. ADODB cần có thông tin về CSDL để thực hiện việc kết nối. Các thông tin này có thể cung cấp thông qua một đối tượng Data Source Name hay cung cập trực tiếp cho đối tượng Connection của ADODB.



#### III.1. Cơ sở dữ liệu

#### Cơ sở dữ liệu

- Để lưu trữ dữ liệu cần thiết cho ứng dụng, người lập trình sử dụng Cơ Sở Dữ Liệu
- Các loại CSDL được sử dụng phổ biến cho ứng dụng Web là Access, MySQL, SQL Server và Oracle
- Sử dụng CSDL để lưu trữ thông tin có nhiều thuận lợi so với file
  - o Tự động quản lý việc thêm, xoá, sửa
  - Thực hiện các truy vấn tìm kiếm
  - Backup/restore dữ liệu

Các ứng dụng Web cũng như các ứng dụng thông thường, sử dụng CSDL để lưu trữ các dữ liệu cần thiết. Trong một vài trường hợp, lưu trữ dữ liệu đơn giản trong file có thể dễ dàng hơn, tuy nhiên, các ứng dụng phức tạp cần có CSDL quan hệ được thiết kế với các bảng, thủ tục, quan hệ, ràng buộc,... để có thể lưu trữ nhiều đối tượng dữ liệu khác nhau và quản lý, bảo vệ những dữ liệu được lưu trữ.

Sử dụng CSDL có nhiều lợi thế so với lưu trữ dữ liệu trên file như bảo mật, tổ chức lưu trữ, thêm, xoá, sửa dữ liệu và quan trọng nhất là thực hiện các truy vấn dữ liệu. Với file, bạn phải tự mình tỗ chức các cấu trúc dữ liệu, kiểm soát việc thêm, xoá, sửa và tìm kiếm. Với mỗi loại dữ liệu khác nhau chẳng hạn khách hàng và sản phẩm, bạn sẽ cần các file riêng biệt để lưu trữ. Bạn cũng cần thêm file để lưu trữ thông tin phát sinh, ví dụ, khách hàng nào mua mặt hàng nào,... Các CSDL giúp bạn dễ dàng tổ chức các bảng lưu trữ và các ràng buộc cần thiết để kiểm tra việc dữ liệu thay đổi khi thêm, xoá, sửa.

Các CSDL thường sử dụng cho ứng dụng Web là MySQL, Access. Ứng dụng web chuyên nghiệp thường dùng SQL Server hay Oracle. Trong phạm vi sách này, chúng tôi sẽ sử dụng CSDL Access vì đây là CSDL phổ biến ở Việt nam, đi cùng với bộ Microsoft Office. Với Access, bạn có thể tổ chức dữ liệu cho hầu hết các ứng dụng Web, đưa ứng dụng của mình lên các Web server trên Internet và dễ dàng chuyển đổi dữ liệu sang SQL Server khi cần nâng cấp ứng dụng để đáp ứng các yêu cầu phức tạp hơn.



#### III.2. Thông tin kết nối đến CSDL

#### Thông tin kết nối đến CSDL

- ADO muốn làm việc với CSDL cần có thông tin kết nối tới CSDL
- Thông tin kết nối có thể sử dụng
  - Đối tượng Connection
  - o DSN (Data Source Name): an toàn hơn

Với Web server chạy trên Windows 2000/XP CSDL Access cần đặt trong thư mục có đủ quyền đọc/ghi với Internet User

Để tương tác được với CSDL, ADO cần thông tin để "tìm" ra vị trí CSDL cần kết nối, thông tin này gọi là thông tin kết nối. Có 2 cách thường dùng để cung cấp thông tin kết nối cho ADO:

- Sử dụng DSN (Data Source Name).
  - Thông tin kết nối được lưu trữ độc lập trong một file của hệ thống được bảo mật bởi Windows. Chỉ có các user có đủ quyền đăng nhập hệ thống mới truy cập được thông tin này. Internet User không truy cập trực tiếp vào thông tin này được.
  - DSN được tạo ra và thay đối sử dụng utility Data Source (ODBC) trong Control Panel | Administrative Tools.
- Gán chuỗi kết nối trực tiếp vào đối tượng kết nối của ADO
  - Trong cách làm này, thông tin kết nối được lưu trong các file ASP của ứng dụng. Thông tin này rất dễ bị gửi tới trình duyệt nếu người lập trình không sử dụng cẩn thận, nhất là khi việc kết nối không thành công.
  - Cách làm này không an toàn bằng cách sử dụng DSN và cũng khó viết hơn vì người lập trình cần biết tên của loại CSDL sẽ kết nối. Tuy nhiên, nếu bạn đặt ứng dụng của mình trên một Web server trên Internet mà bạn phải truy cập gián tiếp qua đường mạng bằng FPT hay HTTP thì đây là cách duy nhất để kết nối đến CSDL. Rõ ràng bạn không thể ngồi tại Web server để tạo ra một DSN.

#### Kết nối với Access

Microsoft Access là CSDL dạng độc lập, không tích hợp với Windows để cung cấp các chức năng CSDL. Vì Windows coi CSDL Access như một file thông thường nên để thêm, xoá, sửa được dữ liệu người dùng phải có quyền đọc và ghi file. Điều này có nghĩa là bạn phải đặt file Access (.mdb) trong thư mục mà Internet user có cả quyền đọc và ghi nếu Web server của bạn đang chạy trên Windows 2000/XP (Windows 98 không kiểm soát quyền đọc/ghi của người dùng).



# III.3. Demo: Tạo DSN



Trong phần này, giáo viên sẽ minh họa các tạo DSN để truy cập vào CSDL:

- Tạo DSN để truy cập CSDL Access
- Tạo DSN để truy cập CSDL SQL Server
- Tạo trang web kết nối với CSDL thông qua DSN vừa tạo



# III.4. Kết nối đến CSDL bằng đối tượng ADODB.Connection

#### Kết nối đến CSDL bằng đối tượng ADODB.Connection

 Đối tượng ADODB.Connection dùng để tạo kết nối tới CSDL

Set CN=Server.CreateObject("ADODB.Connection") CN.ConnectionString= "thông tin  $v\hat{e}$  CSDL" CN.Open

ADO cung cấp một đối tượng Connection cho phép bạn tạo một kết nối tường minh đến CSDL, sau đó, sử dụng một kết nối này trong nhiều lần truy cập CSDL khác nhau.

Đối tượng Connection tạo kết nối giữa ADO và CSDL, một Connection bao gồm 3 thông tin:

- CSDL muốn kết nối, Database server chứa CSDL.
- Phương thức kết nối (Driver/Provider).
- Username và Password để kết nối.

Các thông tin này có thể lấy từ một DSN hay được cung cấp qua chuỗi kết nối ConnectionString. Để tạo ra đối tượng Connection:

Dim Con

Set Con=Server.CreateObject("ADODB.Connection")

#### Tạo kết nối qua DSN

Con.Open "DSN=Pubs" 'DSN được tạo trước tên là Pubs

Để kết nối trực tiếp tới CSDL không thông qua DSN, bạn sử dụng thuộc tính ConnectionString để cung cấp thông tin kết nối:

#### Tạo kết nối với CSDL Access

```
Con.ConnectionString = "DRIVER={Microsoft Access Driver
          (*.mdb)}; Data Source=c:\inetpub\Northwind.mdb"
          'Access 97
Con.Open

Con.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;
          Data Source=" & Server.MapPath("Northwind.mdb")
          'Access 2000 su dung Virtual path
Con.Open
```

#### Tạo kết nối với CSDL SQL

```
Con.ConnectionString = "driver={SQL Server}; server=TIEUYEU; uid=sa;
database=forum"
'MS SQL Server 7
Con.Open
```



Con.ConnectionString = "driver={SQL Server}; server=SERVER\_NAME; uid=UID; pwd=PWD; database=DB\_NAME" 'MS SQL Server 2000 Con.Open

Con.ConnectionString = "driver=MySQL; server=SERVER\_IP; uid=UID;
pwd=PWD; database=DB\_NAME" 'MySQL
Con.Open

#### Chấm dứt kết nối với CSDL

Để chấm dứt kết nối với CSDL bạn có thể đóng Connection hay huỷ đổi tượng Connection:

Con.Close
Set Con = Nothing

# Các lỗi thường gặp khi tạo kết nối với CSDL bằng đối tượng Connection

- Không cung cấp thông tin kết nối: Không có DSN hay chuỗi kết nối
- Thông tin kết nối không chính xác:
  - DSN không tồn tại
  - Sử dụng chuỗi kết nối không chính xác: Sai tên CSDL, tên truy cập, tên driver/provider

Sau khi đã mở kết nối bằng Connection, đối tượng Con sẽ được sử dụng bởi các đối tượng khác của ADODB như RecordSet và Command khi cần thao tác với CSDL.



#### III.5. Bảo mật thông tin kết nối đến CSDL

# Bảo mật thông tin kết nối đến CSDL

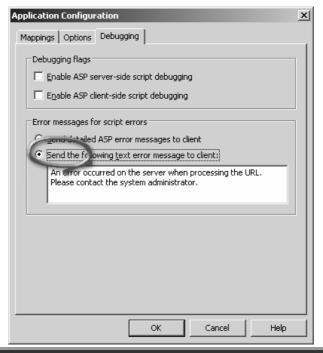
- DSN
  - Được lưu trữ độc lập trong một file của hệ thống
  - Chỉ có các user có đủ quyền mới truy cập được
  - Internet User không truy cập được
- ConnectionString
  - o Được lưu trong các file ASP
  - Rất dễ bị gửi tới trình duyệt của người dùng
  - Khó việt
  - Cách duy nhất để kết nối đến CSDL nếu đặt web server ở xa

Vị trí của CSDL dùng trong ứng dụng web cũng như việc kết nối đến CSDL đóng vai trò quan trọng trong việc bảo mật dữ liệu của ứng dụng.

Bạn biết rằng thành phần tối quan trọng trong một ứng dụng luôn là dữ liệu của nó. Các đọan code hay thư viện lập trình khó có thể bị thay đổi bởi người ngòai trong khi dữ liệu trong các hệ quản trị CSDL hòan tòan có thể bị thay đổi hay xóa sạch bằng cách gọi các lệnh SQL. Một trong những cách hiệu quả để bảo vệ dữ liệu là đừng cho hacker biết được CSDL của bạn được lưu trữ ở đâu. Tiếp theo là bạn đừng cung cấp các công cụ thự hiện câu lệnh SQL được truyền qua URL từ Web browser.

Để hạn chế việc web server "tiết lộ" thông tin về CSDL cho Web browser, thông tin kết nối CSDL cần được bảo mật bởi DSN. DSN sử dụng một file lưu trữ trong hệ thống để cung cấp thông tin kết nối. File này được Windows hạn chế người dùng truy cấp nên đảm bảo Internet user không thể sử dung được.

Thông tin kết nối được lưu vào ConnectionString tương đối dê bị gửi tới Web Browser khi có lỗi thao tác CSDL. Để ngăn không cho Web server gửi thông báo lỗi chi tiết tới Web browser, hãy cấu hình lại ứng dụng trên Web server của bạn.





#### III.6. Tạo kết nối đến CSDL sử dụng file đính kèm

#### Tạo kết nối sử dụng file đính kèm

- Chia sẻ hay cung cấp ứng dụng của mình cho nhiều người khác
- Xây dựng hay kiểm tra ứng dụng ở nhiều máy khác nhau
  - Tạo một file DB.inc
  - o Sử dụng #INCLUDE để đính kèm file DB.inc
  - Bỏ ghi chú cho chuỗi kết nối muốn dùng

Tạo kết nối sử dụng file đính kèm là kỹ thuật thường được nhiều lập trình viên sử dụng khi muốn chia sẻ hay cung cấp ứng dụng của mình cho nhiều người khác. Bạn cũng có thể sử dụng nó khi muốn phát triển hay kiểm tra ứng dụng ở nhiều máy khác nhau.

Trước tiên, bạn hãy tạo một file có tên là DB.inc đặt tại thư mục gốc của ứng dụng với nội dung là các chuỗi kết nối có thể sử dụng:

```
'Const CnxnStr="Provider={}; Server=Server1; Data
Source=Database Name; UID=username1; PWS=pwd1;"

'Const CnxnStr="Provider={}; Server=Server2; Data
Source=Database Name; UID=username2; PWS=pwd2;"

'Const CnxnStr="Provider={}; Server=Server3; Data
Source=Database Name; UID=username3; PWS=pwd3;"

'Const CnxnStr="Provider={}; Server=Server4; Data
Source=Database Name; UID=username4; PWS=pwd4;"
...
```

Các CSDL như Access cần chỉ định file CSDL, bạn nên dùng đường dẫn ảo với Server.MapPath.

Tiếp theo, trong các file .asp bạn muốn tạo kết nối, sử dụng chỉ thị #INCLUDE để đính kèm file DB.inc vào file .asp đó (Xem phần phụ lục để biết cách sử dụng chỉ thị #INCLUDE). Bạn phải nhớ gố ghi chú cho chuỗi kết nối mà bạn muốn dùng.

Trong file .asp, bạn có thể tạo kết nối đơn giản như sau

```
Dim Con
Set Con=Server.CreateObject("ADODB.Connection")
Con.ConnectionString=CnxnStr
Con.Open
```



# IV. Truy cập và hiển thị dữ liệu

# Truy cập và hiển thị dữ liệu Truy cập dữ liệu bằng RecordSet Sử dụng đối tượng RecordSet Hiển thị dữ liệu trong RecordSet

Đối tượng quan trọnh nhất trong ADODB là RecordSet. Nó cho phép bạn thực hiện các câu truy vấn Select để lấy dữ liệu từ CSDL về bộ nhớ và xử lý. Trong phần này, bạn sẽ học cách khai báo và sử dụng đối tượng RecordSet để hiển thị dữ liệu trên trang web, đặc biệt là trình bày thông tin theo dạng table.



#### IV.1. Truy cập dữ liệu bằng RecordSet

#### Truy cập dữ liệu bằng RecordSet

 ADODB.RecordSet dùng để truy cập dữ liệu và thực hiện các thao tác với dữ liệu trong bộ nhớ

Set rs=Server.CreateObject("ADODB.RecordSet")

Truy cập giá trị trong một field của dòng hiện hành:

rs("Ten\_Field").Value hoặc rs.Fields("Ten\_Field").Value

Dữ liệu lấy về từ CSDL được đặt trong đối tượng gọi là RecordSet, ta có thể tạm coi là một tập hợp (set) các mẩu tin (record). Để mở một RecordSet, chỉ cần 3 dòng lệnh như sau:

```
Dim rs
Set rs=Server.CreateObject("ADOVB.RecordSet")

rs.Open "Ten_Bang hoac Chuoi_SQL", g"DSN=Ten_DSN;uid=Ten_user;
pwd=Password"
```

Hoặc

rs.Open "Ten\_Bang hoac Chuoi\_SQL", Con

Ví dụ sau đây mở bảng Authors của CSDL Pubs trên SQL Server sử dụng username sa password rỗng, DSN cho CSDL này có tên là Pubs:

#### ⁴ Ví dụ

```
Dim rs
Set rs=Server.CreateObject("ADOVB.RecordSet")
rs.Open "Author", "DSN=Pubs;uid=sa"
```

Bạn cũng có thể thay thế tên bảng Author bằng một câu truy cấn SQL như "Select \* from Author Order by FirstName".

Sau khi đã mở RecordSet, giá trị dữ liệu trong các trường được truy cập bằng cú pháp:

```
Ten_RecordSet("Ten_field").value
Ten RecordSet.Fields("Ten field").value
```

#### ⁴ Ví dụ:

```
Do while (not rs.EOF)
Response.Write rs.Fields("FirstName").value & "<BR>"
rs.MoveNext
Loop
```

Đoạn mã trên sẽ lần lượt duyệt qua RecordSet rs đã mở và in ra phần tên của tất cả các tác giả trong bảng Author.





#### IV.2. Sử dụng đối tượng RecordSet

# Sử dụng đối tượng RecordSet

- Di chuyển gữa các mẩu tin
- Thêm, xoá, cập nhật dữ liệu
- Sử dụng tập hợp Fields

Sau khi kết nối với CSDL, đối tượng mà bạn thường sử dụng nhất là RecordSet. Với RecordSet, bạn có thể lấy dữ liệu từ Database Server về ứng dụng để thực hiện các xử lý tính toán hay hiển thị dữ liệu cho người dùng. RecordSet cũng cho phép bạn thêm, xoá và cập nhật dữ liệu vào CSDL tuy nhiên, trên ứng dụng Web, việc này ít khi xảy ra.

Trong phần này, bạn sẽ học cách sử dụng RecordSet để truy cập dữ liệu trong CSDL, cách làm việc với từng mẩu tin độc lập của RecordSet và thêm, xoá, sửa dữ liệu trong RecordSet.



#### IV.3. Di chuyển giữa các mẩu tin

# Di chuyển gữa các mẩu tin

- RecordSet.MoveNext | MovePrevious | MoveLast | MoveFirst
  - o Di chuyển tới mẩu tin cần xử lý
  - Nếu RecordSet rỗng sẽ bị lỗi
- RecordSet.EOF | RecordSet.BOF
  - Ngoài các mẩu tin chứa dữ liệu, RecordSet có hai mẩu tin đặc biệt là BOF và EOF
  - Cho biết mẩu tin hiện hành đang ở các vị trí Begin Of File hay End Of File

Bạn chỉ có thể làm việc với một mẩu tin hiện hành trong RecordSet tại một thời điểm. Để di chuyển giữa các mẩu tin khác nhau trong RecordSet, sử dụng các hành động di chuyển:

- MoveNext | MovePrevious: Tới mẩu tin tiếp theo | trước đó
- MoveLast | MoveFirst: Tới mẩu tin cuối cùng | đầu tiên

EOF và BOF là hai vị trí đặc biệt ở cuối và đầu RecordSet, đứng sau mẩu tin cuối cùng và trước mẩu tin đầu tiên. Trong ví dụ ở phần trên, bạn thấy cách áp dụng hành động Move... và thuộc tính EOF.

#### Kết hợp dữ liệu với HTML

Ví dụ sau minh hoạ cách tạo một ComboBox với các mục chọn là các tên sách lấy trong bảng Title của Pubs:

#### ⁴ Ví dụ

```
<%Dim rs
Set rs=Server.CreateObject("ADODB.RecordSet")
rs.Open "Select Title From Titles", "DSN=Pubs"
%>
<P>
Sach: <SELECT name=Titles>
<OPTION selected>Chon ten sach</OPTION>
<%Do While not rs.EOF%>
<OPTION value=<%= rs.Fields("Title").Value%>>
<%= rs.Fields("Title").Value%></OPTION>
<%rs.MoveNext
loop%>
</SELECT>
```

Ví dụ sau minh hoạ cách kiểm tra dữ liệu để đánh dấu một CheckBox:

#### ⁴ Ví du

<%if rs.Fields("Finished") then%>
Da hoan tat <INPUT name=chkFinished type=checkbox checked>
<%else%>



Da hoan tat <INPUT name=chkFinished type=checkbox> <%End if%>



#### IV.4. Thêm, xoá, cập nhật dữ liệu

#### Thêm, xoá, cập nhật dữ liệu

- Các hành động dùng để thay đổi dữ liệu trong một mẫu tin của RecordSet
  - o RecordSet.AddNew: Thêm mới
  - o RecordSet.Update: Cập nhật
  - RecordSet.Delete: Xoá
- RecordSet ít được dùng để cập nhật dữ liệu trong ngữ cảnh của ứng dụng Web
- Các câu truy vấn SQL thường dùng hơn

Sử dụng RecordSet, bạn có thể thay đổi dữ liệu trong CSDL bày cách thêm, xóa hay cập nhật dữ liệu trong các mẩu tin của RecordSet. Điểm hạn chế nhất của RecordSet là những hành động thay đổi dữ liệu chỉ có thể thực hiện trên một mẩu tin – chính là mẩu tin hiện hành.

Để đảm bảo RecordSet có thể dùng để cập nhật dữ liệu, tham số trong hành động Open không thể chỉ gồm câu SQL và đối tượng Connection, thêm vào đó, bạn phải chỉ ra các thuộc tính về CursorType và LockType.

Thông thường, trong các ứng dụng Web bạn sử dụng hai giá trị adOpenStatic và adLockOptimistic để mở RecordSet.

#### Thêm mấu tin mới

Một mẩu tin trắng có thể được thêm vào RecordSet sau khi gọi hành động AddNew. Sau khi gọi AddNew, mẩu tin hiện hành sẽ là mẩu tin vừa được thêm mới, bạn phải gán giá trị cho từng field trong mẩu tin trước khi gọi hành động Update để cập nhật xuống CSDL.

#### ⁴ Ví du

```
Rs.AddNew
Rs("Field1").Value = "..."
Re("Field2").Value = ...
Rs.Update
```

#### Xóa mấu tin

Để xóa mẩu tin hiện hành, bạn đơn giản chỉ cần gọi hành động Delete.

Rs.Delete

#### Cập nhật dữ liệu trong RecordSet

Dữ liệu của RecordSet được lưu trong bộ nhớ mà chương trình sử dụng RecordSet quản lý do đó, để cập nhật dữ liệu từ RecordSet xuống CSDL, bạn cần gọi hành động Update hoặc UpdateBatch.

Ngược lại, nếu không muốn cập nhật những thay đổi đã thực hiện trên RecordSet, bạn sử dụng hành động CancelUpdate hoặc CancelBatch.

#### Ngữ cảnh sử dụng



Rất ít khi bạn sử dụng RecordSet để cập nhật dữ liệu vì khác với ứng dụng windows, RecordSet được duy trì trên form, RecordSet trong ứng dụng Web bị xóa ngay sau khi trang web hiển thị dữ liệu được gửi về cho Web Browser.



#### IV.5. Sử dụng tập hợp Fields

#### Sử dụng tập hợp Fields

- Tập hợp Fields bao gồm tất cả các đối tượng Field đại diện cho các field trong RecordSet
- Một đối tượng Field, đại diện một field trong RecordSet
  - Name: Tên fieldValue: Giá tri của field
  - OriginalValue: Giá trị trước khi cập nhật
  - Type: Kiểu dữ liệu

Tập hợp Fields của đối tượng RecordSet bao gồm tất cả các đối tượng Field đại diện cho các field trong RecordSet. Bạn ít khi phải sử dụng đến tập hợp Fields và đối tượng Field trừ khi cần truy cập dữ liệu của một trường. Tuy nhiên, trong các thủ tục dùng chung, khi bạn không xác định được RecordSet cụ thể, Fields và Field tỏ ra rất hữu ích. Ví dụ dưới đây minh hoạ một thủ tục trình bày dữ liệu của RecordSet theo dạng bảng

#### ⁴ Ví dụ

```
<응
Sub ShowTable(RS)
                             'RS la mot RecordSet
If RS.BOF And RS.EOF Then Exit Sub
Response.Write "<TABLE>"
'Dong dau tien trinh bay ten cac cot du lieu
Response.Write "<TR>"
For each f in RS.Fields
   Response.Write "<TH>f.Name</TH>"
Next
Response.Write "</TR>"
RS.MoveFirst
Do While (Not RS.EOF)
   Response.Write "<TR>"
   For each f in RS.Fields
      Response.Write "<TD> &nbsp;" & f.value & "</TD>"
      'Tranh gia tri NULL
   Next
   Response.Write "</TR>"
Response.Write "</TABLE>"
End Sub
```

Bạn có thể tạo một thủ tục "chuyên nghiệp" hơn để sử dụng cho chính mình bao gồm các tham số truyền vào để thay đổi cách trình bày Table như Border, Cell space, Cell padding, BackGround color, ForeColor,... 2 màu BackGround thay đổi luân phiên giữa các dòng trong bảng, danh sách các cột muốn trình bày, độ rộng mỗi cột ước lượng dựa vào chiều dài trường dữ liệu,... Khi viết thủ tục này bạn sẽ thấy không chỉ phải kết hợp ASP linh hoạt với HTML mà cũng sẽ sử dụng rất nhiều thuộc tính của tập hợp Fields và đối tượng Field.



#### IV.6. Hiển thị dữ liệu trong RecordSet

Tương tự với cách tạo ComboBox ở trên bạn có thể trình bày dữ liệu trong ListBox hay trong Table.

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<!--#include file="ADOVBS.inc"-->
<%Dim conn
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "DSN=example"
'Mo recordset
Set rs = Server.CreateObject("ADODB.Recordset")
rs.CursorLocation = adUseClient
Dim SQL
SQL = "SELECT * FROM example"
rs.Open SQL, Conn
'Bat dau tu mau tin dau tien
rs.MoveFirst%>
<HTML>
<HEAD><TITLE>Hien thi du lieu dang bang</TITLE></HEAD>
<BODY>
<TABLE BORDER=1 WIDTH=600>
<TH><TD>NAME</TD></TH>
<TH><TD>EMAIL</TD></TH>
<TH><TD>PHONE</TD></TH>
</TR>
<%Do While Not rs.eof And MauTin < rs.RecordCount%>
<TD><%= rs("Name") %></TD>
<TD><%= rs("Email") %></TD>
<TD><%= rs("Phone") %></TD>
</TR>
<%rs.MoveNext
Loop %>
```



```
</TABLE></BODY></HTML>
```

Trong ví dụ trên, chúng tôi cố gắng trình bày code của một file ASP hoàn chỉnh. Bạn sẽ chú ý thấy chỉ thị #INCLUDE ADOVBS.INC. Đây là file chứa các hằng số của ADO cần dùng như adUseClient.

#### Hiển thị RecordSet có nhiều mẩu tin trên nhiều trang web

```
Khi dữ liệu người dùng yêu cầu gồm nhiều mẫu tin, việc hiển
thị dữ liệu theo dạng bảng và phân trang là một cách được dùng
phổ biến để nhanh chóng đáp ứng yêu cầu của người dùng.
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<!--#include file="ADOVBS.inc"-->
Const RecTrang = 2 'So mau tin se hien thi moi trang
'Xac din trang can hien thi
Dim TrangHH=request.querystring("TrangHH")
    TrangHH= "" then
     TrangHH = 1
End If
Dim conn
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "DSN=example"
'Mo recordset
Dim rs
Set rs = Server.CreateObject("ADODB.Recordset")
rs.CursorLocation = adUseClient 'Dat vi tri cua cursor
'Kich thuoc vung nho tam bang so mau tin
rs.CacheSize = RecTrang
Dim SQL
SQL = "SELECT * FROM example"
rs.Open SQL, Conn
'Bat dau tu mau tin dau tien
rs.MoveFirst
rs.PageSize = RecTrang 'Dat kich thuoc moi trang bang so mau
tin mot trang
'Tinh tong so trang can hien thi
Dim SoTrang
SoTrang = rs.PageMauTin
rs.AbsolutePage = TrangHH 'Toi trang can hien thi du lieu
Dim MauTin
%>
<HTMT<sub>1</sub>>
<HEAD><TITLE>Hien thi phan trang</TITLE></HEAD>
<BODY>
<TABLE BORDER=1 WIDTH=600>
<TR>
<TH>NAME</TH>
<TH>EMAIL</TH>
<TH>PHONE</TH>
</TR>
<응
MauTin = 0
Do While Not rs.eof And MauTin < rs.PageSize
응>
```



```
<TD><%= rs("Name") %></TD>
<TD><%= rs("Email") %></TD>
<TD><%= rs("Phone") %></TD>
</TR>
<응
MauTin = MauTin + 1
rs.MoveNext
Loop
response.write "</TABLE>"
'Hien thi trang hien hanh/Tong so trang
response.write("Trang " & TrangHH & "/" & SoTrang)
'Hien thi nut TRUOC va SAU
If TrangHH > 1 then
     'Nut TRUOC
     response.write("<INPUT TYPE=BUTTON VALUE='<<TRUOC'")</pre>
     response.write("ONCLICK=""document.location.href=
                     'display_data.asp?TrangHH=")
     response.write( TrangHH - 1 & "';"">")
End If
If CInt(TrangHH) <> CInt(SoTrang) then
     'Nut SAU
     response.write("  <INPUT TYPE=BUTTON
VALUE='SAU>>'")
    response.write("ONCLICK=""document.location.href=
                     'display_data.asp?TrangHH=")
     response.write( TrangHH + 1 & "';"">")
End If
응>
</BODY>
</HTML>
```



# Bài 8

# SỬ DỤNG ADODB: CẬP NHẬT DỮ LIỆU

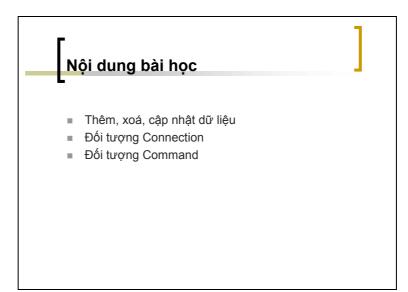
# Tóm tắt

Lý thuyết 8 tiết - Thực hành 12 tiết

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài giảng tập trung vào việc sử dụng các đối tượng Connection và Command để thực hiện các câu SQL cập nhật dữ liệu. Kiến thức quan trọng nhất cần trình bày với học viên là việc sử dụng các hidden field để chứa các giá trị khóa nhằm xác định đúng dữ liệu cần cập nhật. Các kiến thức liên quan đế cú pháp của câu lệnh SQL giáo viên có thể yêu cầu học viên tự ôn tập.  Sau khi hoàn tất bài học này, học viên có thể:  Xây dựng các câu SQL để cập nhật dữ liệu dựa trên các điều kiện về khóa chính lấy từ các hidden field của form  Sử dụng  Connection hay  Command để thực hiện các câu lệnh SQL  Phân biệt sự khác nhau trong cách sử dụng Command và Connection để thực hiện các câu SQL.	<ul> <li>I. Giới thiệu bài học</li> <li>II. Thêm, xoá, cập nhật dữ liệu</li> <li>III. Đối tượng Connection</li> <li>IV. Đối tượng Command</li> </ul>	8.1 – 8.6	



# I. Giới thiệu bài học



Việc cập nhật dữ liệu trong ứng dụng Web thường được tiến hành qua 3 bước: hiển thị dữ liệu, chờ người dùng chọn dữ liệu để cập nhật và cập nhật dữ liệu vào CSDL. Do phải đi qua 3 bước, RecordSet không còn là đối tượng thích hợp để thêm, xoá hay cập nhật dữ liệu vào CSDL, thay vào đó, bạn thường dùng cáp câu truy vấn SQL để thực hiện việc cập nhật.

Bài này hướng dẫn bạn cách xây dựng các câu SQL cập nhật dữ liệu và cách sử dụng đối tượng Connnection hay Command để thực hiện những câu SQL đó.



# II. Thêm, xoá, cập nhật dữ liệu

# Thêm, xoá, cập nhật dữ liệu

- Cập nhật dữ liệu trong ứng dụng web
- Xác định thông tin cần cập nhật
- Các câu truy vấn cập nhật dữ liệu
- Cập nhật dữ liệu

Để thực hiện được việc cập nhật dữ liệu, điểm quan trọng nhất là bạn phải xác định được dữ liệu nào người dùng muốn cập nhật và thông tin cập nhật cho dữ liệu đó là gì. Tiếp theo, bạn chỉ cần xây dựng câu SQL Insert/Update/Delete thích hợp để thực hiện việc cập nhật vào CSDL.



#### II.1. Cập nhật dữ liệu trong ứng dụng web

#### Cập nhật dữ liệu trong ứng dụng web

- Để tăng khả năng đáp ứng các yêu cầu từ nhiều Client và tiết kiệm tài nguyên cho Web server, các trang ASP không nên chiếm dụng nhiều bộ nhớ trong thời gian dài
- RecordSet được tạo ra trong trang Web không được duy trì liên tục giữa các lần tương tác của Web server và Client
- Để cập nhật dữ liệu vào CSDL, sử dụng các câu lênh SQL INSERT, UPDATE, DELETE

Nếu bạn đã từng lập trình với CSDL trên các ứng dụng Windows, có thể bạn sẽ cảm thấy khác biệt giữa hai cách lập trình ứng dụng Windows và Web. Vì ứng dụng chạy trên Web phải phục vụ nhiều kết nối, Web server sử dụng giao thức HTTP phi trạng thái và không duy trì kết nối với Client. Tương tự như vậy, Server không duy trì kết nối tới CSDL được tạo trong một trang ASP nào đó sau khi nó đã xử lý xong.

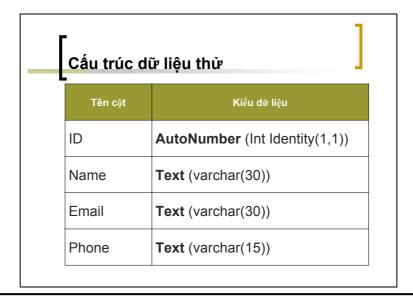
Giải thích cơ chế này khá đơn giản. Nếu ứng dụng của bạn gồm có nhiều trang ASP cần xử lý dữ liệu, mỗi trang cần có những RecordSet riêng biệt. Tại một thời điểm, có rất nhiều người truy cập vào ứng dụng, mỗi người có thể đang truy cập những trang khác nhau và mở các RecordSet khác nhau. Các RecordSet này ít nhiều đều chiếm tài nguyên bộ nhớ của Web server. Để tiết kiểm bộ nhớ và tăng cường tốc độ đáp ứng các yêu cầu của nhiều Client, sau khi xử lý xong một trang ASP, Web server bắt buộc phải huỷ RecordSet mà trang đó tạo ra.

RecordSet chỉ được giữ lại nếu bạn lưu nó vào trong Session. Tuy nhiên, việc này rõ ràng là không nên vì kích thước của RecordSet là quá cồng kềnh, nếu bạn biến nó thành một biến toàn cục thì sẽ rất phí bộ nhớ. Người dùng có thể chỉ lướt qua trang web của bạn và sau đó thì ... "lặn mất tăm". Trong khi đó, Web server "ngây thơ" của chúng ta thì vẫn cố gắng duy trì Session cho Client trong khoảng Session.Timeout.

Do RecordSet bị huỷ ngay sau khi hiển thị dữ liệu lên trang web, bạn không thể dùng nó để cập nhật dữ liệu. Thay vào đó, chúng ta sẽ sử dụng những câu lệnh SQL INSERT, UDPATAE, DELETE để cập nhật dữ liệu vào CSDL.



#### II.2. Cấu trúc dữ liệu thử



Thông thường, việc cập nhật dữ liệu được thực hiện trên các bảng khác nhau trong CSLD vì mỗi câu lệnh INSERT, UPDATE, DELETE chỉ thực hiện trên một bảng. Trong các ví dụ thêm, xoá, sửa dưới đây, chúng tôi sẽ dùng một bảng dữ liệu đơn giản tên là **EXAMPLE**.

Trong bảng trên, kiểu dữ liệu chính được trình bày dành cho Access. Kiểu dữ liệu trong dấu () chúng tôi ghi chú cho bạn nếu sử dụng SQL Server.

#### Khác biệt giữa T-SQL và ANSI SQL 92

Microsoft Access và Microsoft SQL Server sử dụng hai ngôn ngữ truy vấn SQL khác nhau. Mặc dù hầu hết các cú pháp là không thay đổi, SQL Server sử dụng nhiều hàm khác hẳn so với Access. Ví dụ, SQL Server sử dụng SubString trong khi Access dùng mid. SQL Server dùng hàm CONVERT trong khi Access dùng format, SQL Server sử dụng cấu trúc CASE WHEN THEN trong khi Access dùng iif...

Cơ chế thực hiện các câu SQL của SQL Server cũng có khác biệt so với Access. Ví dụ cập nhật sau không thể chạy trong Access, chỉ SQL Server mới thực hiện theo cơ chế tự nhiên như vậy:

#### **↑** SQL Server

Update PhieuThu

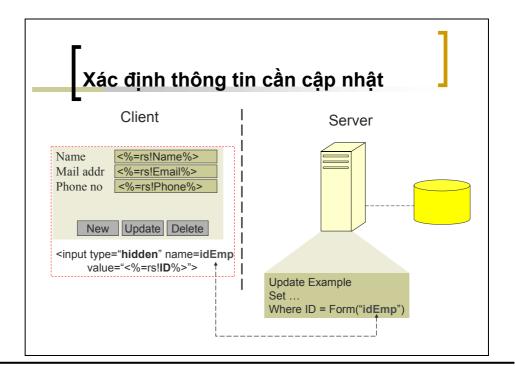
Set ThanhTien=(Select Sum(DonGia\*SoLuong) From ChiTietThu Where ChiTietThu.SoPT=PhieuThu.SoPT)

#### **⁴** MS Access

Update PhieuThu P Inner Join ChiTietThu C On P.SoPT=C.SoPT Set ThanhTien=ThanhTien + SoLuong\*DonGia



#### II.3. Xác định thông tin cần cập nhật



Quá trình cập nhật dữ liệu trong ứng dụng Web thường thông qua 3 bước:

- 1. Truy cập CSDL để lấy dữ liệu hiển thị trên trang web
- 2. Người dùng chọn dữ liệu cần cập nhật và thực hiện các thay đổi trên dữ liệu
- 3. Web Server nhận những thay đổi và cập nhật vào CSDL

Nói chung 3 bước này cũng tương đối giống một ứng dụng Windows thông thường, tuy nhiên đặc điểm của ứng dụng Web có điểm khác hẳn so với ứng dụng Windows trong đó, RecordSet được duy trì suốt quá trình cập nhật.

Với ứng dụng web, tại thời điểm chuyển từ bước 1 sang 2 và từ 2 sang 3 thì Web server hòan tòan không xác định được dữ liệu mà người dùng sẽ cập nhật là gì và cập nhật vào đâu.

Để xác định những thông tin sẽ cập nhật, các thông tin nhập liệu sẽ được đặt trong form kể cả trong trường hợp người dùng chỉ sử dụng một hành động click để cập nhật dữ liệu (thêm hay xóa dữ liệu chẳng hạn).

Để xác định vị trí của dữ liệu sẽ cập nhật, bạn phải dùng khóa chính tương ứng với dòng dữ liệu đó. Thông thường, khóa chính của các bảng không cần hiển thị trên trang web nên để nhúng field khóa chính vào form, bạn sẽ dùng một hidden field cho mỗi field tham gia vào bộ khóa.

Hidden field là một thành phần của form được tạo ra bằng tag <input> với type="hidden". Ví dụ sau tạo ra một hidden field trong form với giá trị định sẵn là giá trị của cột làm khóa chính:

```
⁴ Ví du
```

```
<form name="..." ...>
...
<input type="hidden" name="EmpID" Value="<%= rs!ID%>">
...
</form>
```



# II.4. Các câu truy vấn cập nhật dữ liệu

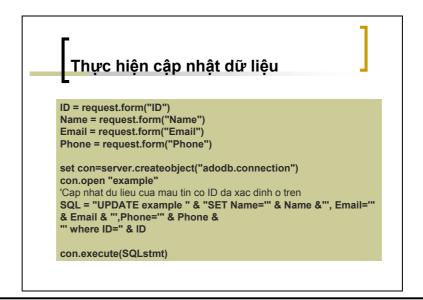
# Các câu truy vấn cập nhật dữ liệu

- Insert Into Table (col1, col2, ...) Values (value1, value2, ...)
- UPDATE table SET column='value' column2=value... where column='...' AND ...
- Delete \* From Table where column='...' And ...
- Delete \* From Table

Trên slide này, giáo viên ôn lại một số cú pháp của các câu truy vấn cập nhật dữ liệu cho học viên.



# II.5. Thực hiện cập nhật dữ liệu



Trên slide này, giáo viên giải thích ý nghĩa các dòng lệnh được thực hiện cho học viên.



# III. Đối tượng Connection

#### Đối tượng Connection

- Kết nối không tường minh được duy trì với CSDL trong khi server xử lý trang ASP
- Kết nối tường minh
  - o Có thể dùng cho nhiều RecordSet
  - Ngắt kết nối ngay khi đã lấy dữ liệu
- CN.Execute: Thực hiện câu SQL thay cho việc phải sử dụng RecordSet

Ở những ví dụ trước, bạn thấy rằng ta có thể truy cập dữ liệu từ CSDL mà không cần tạo một kết nối tường minh tới CSDL đó. Mỗi lần tạo một RecordSet, bạn sẽ đồng thời tạo một kết nối đến CSDL, kết nối đó chỉ ngắt khi bạn thực hiện hành động RecordSet.Close hoặc khi đối tượng RecordSet bị xoá bỏ.

Thông thường, trên một trang ASP, bạn sẽ phải sử dụng nhiều RecordSet khác nhau dẫn đến việc tạo nhiều kết nối với CSDL. Ngoài ra, ứng dụng Web của bạn còn có nhiều người sử dụng dẫn đến Web Server và Database Server sẽ phải tốn rất nhiều chi phí để quản lý hàng loạt kết nối không tường minh được tạo ra.

ADO cung cấp một đối tượng Connection cho phép bạn tạo một kết nối tường minh đến CSDL, sau đó, sử dụng một kết nối này trong nhiều lần truy cập CSDL khác nhau. Tạo kết nối tường minh đến CSDL cũng tăng tốc độ thực hiện chương trình đồng thời giảm bớt tài nguyên của Web server bị chiếm dụng.

#### Thực hiện câu truy vấn bằng Connection. Execute

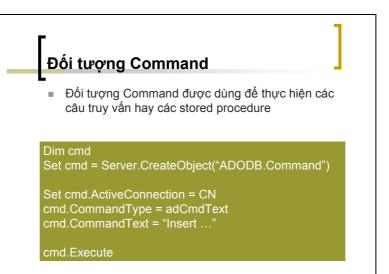
Thực hiện câu truy vấn là một trong những chức năng hay dùng nhất của đối tượng Connection. Bạn có thể sử dụng một đối tượng Command của ADODB để gọi thực hiện câu truy vấn hay thậm chí là một stored procedure nhưng với một câu truy vấn đơn giản, dùng Connection. Execute cũng giúp tăng tốc độ lập trình của bạn.

Để lập một câu lệnh truy vấn, thực hiện câu lệnh và xử lý lỗi, bạn có thể theo cấu trúc lập trình sau:

```
Dim sql
sql = "Câu lệnh SQL"
Con.Execute sql
If Con.Errors.Count > 0 Then
    Dim err
    Set err = Con.Errors(0)
    Response.Write err.Description
End if
```



# IV. Đối tượng Command



Đối tượng Command có thể nói là đối tượng ít được sử dụng và tìm hiểu nhất. Lý do là trong khi việc sử dụng RecordSet và Connection của ADODB gần như là thiết yếu, hầu hết các công việc mà Command làm được đều có thể thực hiện bằng đối tượng Connection. Command cũng không dễ sử dụng với đối tượng Parameter và do đó, những người mới làm quen với ADO thường bỏ qua đối tượng này hoặc chỉ thực sư tìm hiểu những qì cần thiết để giải quyết vấn đề mà ho gặp phải.

Tuy nhiên, Command không phải là một đối tượng thừa của ADO hay được đưa ra cho phù hợp với cấu trúc của CSDL quan hệ thường sử dụng các câu truy vấn. Command thực sự hữu dụng và là công cụ đắc lực cho những lập trình viên chuyên nghiệp khi xét tới mặt kỹ thuật lập trình và tốc độ thực hiện của ứng dụng.

# Khi nào nên sử dụng Command

Có rất nhiều trường hợp mà bạn cần sử dụng đối tượng Command:

- Khi thực hiện các câu truy vấn nhằm thay đổi cấu trúc của CSDL như cấu trúc bảng hay định nghĩa của một stored procedure chẳng hạn.
- Khi bạn cập nhật dữ liệu của bảng (thêm, xoá, sửa) và truyền tham số vào chuỗi SQL hay stored procedure. Command đặc biệt hữu ích khi công việc cần thực hiện không chỉ một lần (sử dụng trong vòng lặp). Bạn có thể sử dụng thuộc tính Prepared để tăng tốc độ thực hiện.
- Khi bạn muốn truyền tham số vào các câu truy vấn SQL.
- Khi ban muốn sử dung các stored procedure.



#### Sử dụng đối tượng Command

Đối tượng Command được tạo lập tương tự như RecordSet hay Connection, cũng giống như RecordSet, Command cần kết nối với CSDL bằng chuỗi kết nối ConnectionString hoặc một kết nối sẵn có ActiveConnection.

```
Set Con=Server.CreateObject("ADODB.Connection")
Con.Open "DSN=..."
Set Cmd=Server.CreateObject("ADODB.Command")
Set Cmd.ActiveConnection=con
```

Thuộc tính CommandType chỉ định loại câu truy vấn chứa trong CommandText mà bạn muốn thi hành. Để tăng tốc độ thực hiện câu truy vấn, bạn nên đặt giá trị CommandType tương ứng với loại câu truy vấn bạn sử dụng. Các giá trị của CommandType bao gồm:

- adCmdText: câu truy vấn là một chuỗi SQL.
- adCmdTable: câu truy vấn là một bảng.
- adCmdStoredProc: câu truy vấn là một Stored Procedure.
- **adCmdUnknown**: câu truy vấn chưa xác đinh được. (Giá tri này là mặc đinh)

Sau khi xác định loại câu truy vấn, nội dung của câu truy vấn được đặt trong thuộc tính **CommandText**. Bạn có thể thay đổi giá trị **CommandTimeout** xác định khoảng thời gian tối đa mà câu truy vấn có thể thực hiện trước khi bị ngắt và trả về lỗi. Đôi khi bạn phải đặt giá trị CommandTimeout lớn hơn trị mặc định là 30 giây nếu ứng dụng Web của bản phải xử lý quá nhiều kết nối cùng một lúc hay khi bạn dự đoán câu truy vấn sẽ cần khoảng thời gian dài để hoàn tất.

Như đã nói, một trong những đặc điểm để sử dụng Command là vấn đề tốc độ, thuộc tính **Prepared** là một giá trị Boolean (Đúng/Sai) để yêu cầu Database Server có hay không sử dụng bộ nhớ để lưu trữ nội dung câu truy vấn cho các lần thực hiện sau đó nhanh hơn. Nếu câu truy vấn của bản chỉ thực hiện một lần, đặt Prepared=True chỉ làm giảm tốc độ thực hiện và chiếm vùng nhớ mà thôi. Ngược lại, nếu bạn cần thực hiện câu truy vấn nhiều lần, nên đặt Prepared=True. Thuộc tính này chỉ cần đặt một lần mà không cần lặp lại trong mỗi lần thực hiện nếu không Database Server sẽ lặp lại các bước chuẩn bị trước mỗi lần thực hiện truy vấn và tốc độ sẽ bị giảm xuống.

Cuối cùng, để yêu cầu thực hiện câu truy vấn, bạn dùng hành động Command.Excute.

```
Cmd.CommandType=adCmdText
Cmd.CommandText="Alter Table Titles Add LibraryID INT"
Cmd.ConnectionTimeOut=20
Cmd.Prepared=False 'Gia tri mac dinh, co the khong can
Cmd.Excute
```







# ĐỀ THI MẪU CUỐI MÔN

#### Đề thi :

# Lập trình web cơ bản

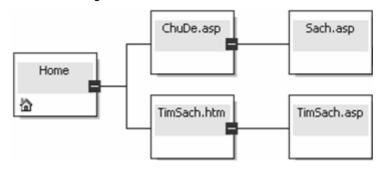
Thời gian : 120 phút Ngày thi : .../.../...

\*\*\* Các kết quả lưu đĩa làm việc để chấm điểm \*\*\*

Tạo thư mục SoMay\_TenHV để chứa toàn bộ các tập tin được tạo ra. Ví dụ: học viên Nguyễn Thái Hưng làm bài thi tại máy số 01 sẽ tạo thư mục 01\_NTHUNG.

#### Câu 1 TỔ CHỨC WEBSITE

Tạo website với sơ đồ tổ chức các trang web như sau:



Caâu 1 Hình 1.1 – Sơ đồ liên kết các trang web

Trang chủ có cấu trúc frameset như mô tả ở câu 2:

#### Câu 2

Thư viện Đại học Khoa Học Tự Nhiên				
Các chủ đề	Các chủ đề			
<u>Tìm sách</u>	<u>Tổng quát</u>	Tin học & CNTT	Triết học	
Mượn sách	<u>Tôn giáo</u>	Khoa hoc xã hôi	Ngoai ngữ	
	Toán học	<u>Vật lý</u>	Hoá học	
	<u>Sinh vật</u>			

Caâu 2 Hình 1.2 – tổ chức frameset của trang chủ

Các liên kết trong frame bên trái sẽ được hiển thị trong frame chính:

2..1 Các chủ để: Liên kết tới trang *ChuDe.asp*2..2 Tìm sách: Liên kết tới trang *TimSach.htm* 



#### Câu 3 XÂY DỰNG CƠ SỞ DỮ LIỆU

Tạo CSDL Access có tên là **ThuVien.mdb** với các bảng và dữ liệu như sau:

#### SACH - Sách

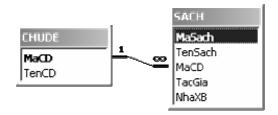
Field Name	Field Type	Field Size	Description
<u>MaSach</u>	Autonumber		Mã sách
TenSach	Text	30	Tên sách
MaCD	Text	3	Mã chủ đề
TacGia	Text	50	Tác giả
NhaXB	Text	50	Nhà xuất bản

#### CHUDE - Chủ đề

Field Name	Field Type	Field Size	Description
<u>MaCD</u>	Text	3	Mã chủ đề
TenCD	Text	30	Tên chủ đề

 $\underline{\text{Chú }\acute{y}}\text{: }$  Các field in đậm, nghiêng và gạch dưới là khoá chính của bảng.

#### Sơ đồ quan hệ giữa các bảng



#### Dữ liệu thử

SACH				
MaSach	TenSach	MaCD	TacGia	NhaXB
1	Toán cao cấp A1	510	Nguyễn Tiến Hưng	Giáo dục
2	Toán cao cấp A2	510	Trịnh Thanh Đèo	Giáo dục
3	Toán cao cấp A3	510	Cao Ngọc Lan	Giáo dục
4	Cấu trúc máy tính	001	Trần Thanh Hải	Kỹ thuật
5	Vật lý 10	530	Nguyễn Võ Hùng Thăng	Hà nội
6	Hoá học 11	540	Nguyễn Thi Minh Châu	Giáo dục
7	Hoá học 10	540	Nguyễn Thi Minh Châu	Giáo dục
8	Lập trình Web	001	Lê Văn Hưu	Kỹ thuật
9	Cấu trúc dữ liệu	001	Trần Thanh Hải	Trẻ
10	Văn phạm Anh ngữ thực hành	400	Phạm Đức Hạnh	Trẻ

CHUDE			
MaCD	TenCD		
000	Tổng quát		
001	Tin học & CNTT		
100	Triết học		
200	Tôn giáo		
300	Khoa học xã hội		
400	Ngoại ngữ		
510	Toán học		
530	Vật lý		
540	Hoá học		
560	Sinh vật		



#### Câu 4 THỰC HIỆN CÁC CHỨC NĂNG

#### Trang ChuDe.asp

Liệt kê tất các các chủ đề sách có trong CSDL theo cách trình bày dạng bảng, mỗi dòng gồm ba chủ đề.

Úng với mỗi chủ đề được liệt kê là một mối liên kết mà khi người dùng click sẽ mở trang *Sach.asp* liệt kê các sách thuộc chủ đề đó.

#### **Trang Sach.asp**

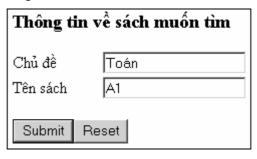
Xử lý các thông tin gửi về từ trang *ChuDe.asp* để liệt kê các sách thuộc chủ đề đó như sau:

Các sách trong chủ đề Hoá học				
Tên sách	Tác giả	NXB		
Hoá học 10	Nguyễn Thị Minh Châu	Giáo dục		
Hoá học 11	Nguyễn Thị Minh Châu	Giáo dục		

Caâu 3 Hình 3.1 – Sách theo chủ đề được chọn từ trang ChuDe.asp (Hoá học)

#### Trang TimSach.htm

Trang Tìm sách cho phép người dùng đưa ra các điều kiện để tìm các cuốn sách mình muốn.



Caâu 4 Hình 3.2 – Trang tìm sách

Các thông tin tìm kiếm đều ở dạng gần đúng, khi người dùng click vào nút Submit, trang *TimSach.asp* sẽ hiển thi các sách thoả điều kiên tìm kiếm

#### Trang TimSach.asp

Xử lý các thông tin gửi về từ trang *TimSach.htm* để liệt kê những sách thoả điều kiện tìm kiếm.



Caâu 5 Hình 3.3 – Kiết quả tìm kiếm sách

