

Bài 1

Tổng quan về Thiết kế và lập trình Web

Mục đích

- Cung cấp cho Sinh viên kiến thức về cơ chế hoạt động của trang web
 - Khái niệm siêu văn bản (hypertext)
 - Máy phục vụ, trình duyệt web, giao thức hoạt động

Nội dung

- 1. Khái niệm về siêu văn bản
- 2. Các thành phần của Web
 - 2.1. Máy phục vụ, máy khách
 - 2.2. WebServer, WebBrowser
 - 2.3. Website, WebPage
 - 2.4. URL
 - 2.5. HTTP
- 3. Phân loại trang Web
- 4. Một số điều cần chú ý trong phát triển Web

0. Lịch sử phát triển của WWW

- Năm 1978, Ward Christensen và Randy Suess đã xây dựng một hệ thống liên lạc đơn giản giữa 2 máy tính thông qua đường dây điện thoại... CBBS (tên viết tắt của Computerized Bulletin Board System - Hệ thống bảng tin trên nền máy tính) Đây là những bước đi đầu tiên để bắt đầu một kỷ nguyên thông tin mới – kỷ nguyên của World Wide Web
- Các hệ thống CBBS trở nên rất thông dụng vào thập niên 80 và đầu những năm 90 của thế kỷ trước. Có thể khẳng định rằng CBBS chính là tiền thân của mạng World Wide Web hiện nay

Cơ chế hoạt động của WWW

■ Cơ chế hoạt động

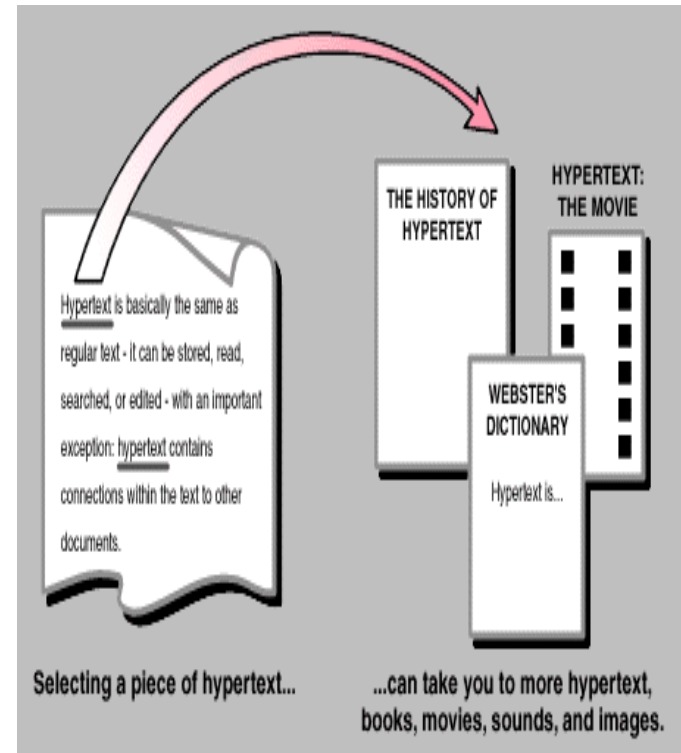
– WWW hoạt động dựa trên 3 cơ chế:

- Giao thức HTTP (Hyper Text Transfer Protocol): Dùng để truy cập tài nguyên trên web.
- Địa chỉ URL (Uniform Resource Locator): Nhận dạng các trang và tài nguyên trên web
- Ngôn ngữ đánh dấu siêu văn bản HTML: Tạo các tài liệu có thể truy cập trên web



1. Khái niệm về siêu văn bản

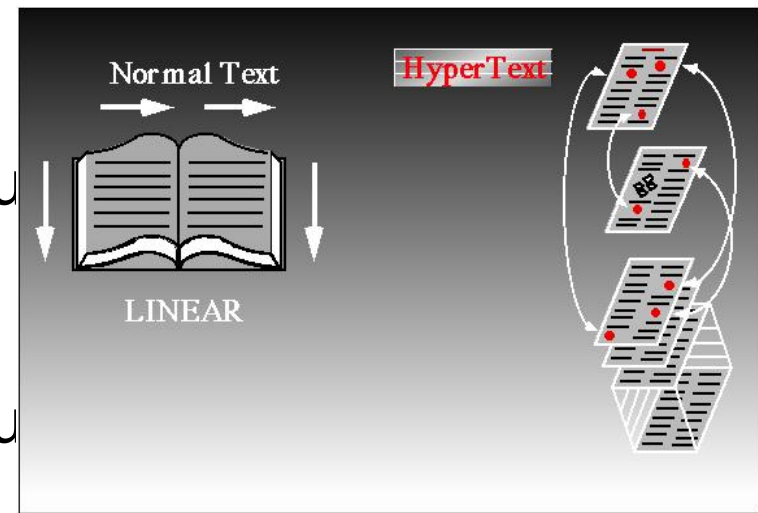
- Hypertext? Hyperlink? Hypermedia?
- **Hypertext:** là văn bản (text) trên máy tính có sự *liên kết* đến một văn bản khác và có thể *truy cập trực tiếp* thông qua click chuột, nhấn phím.
- **Hyperlink:** là một liên kết từ một file hypertext đến một file khác
- **Hypermedia:** là sự mở rộng của hypertext, trong đó đối tượng có thể là văn bản, đồ họa, âm thanh, video...



H1. Hypertext

1. Khái niệm về siêu văn bản

- Một số ngôn ngữ siêu văn bản tiêu biểu
 - HTML (Hypertext Markup Language) - Ngôn ngữ đánh dấu siêu văn bản
 - XML (Extensible Markup Language) - Ngôn ngữ đánh dấu có thể mở rộng

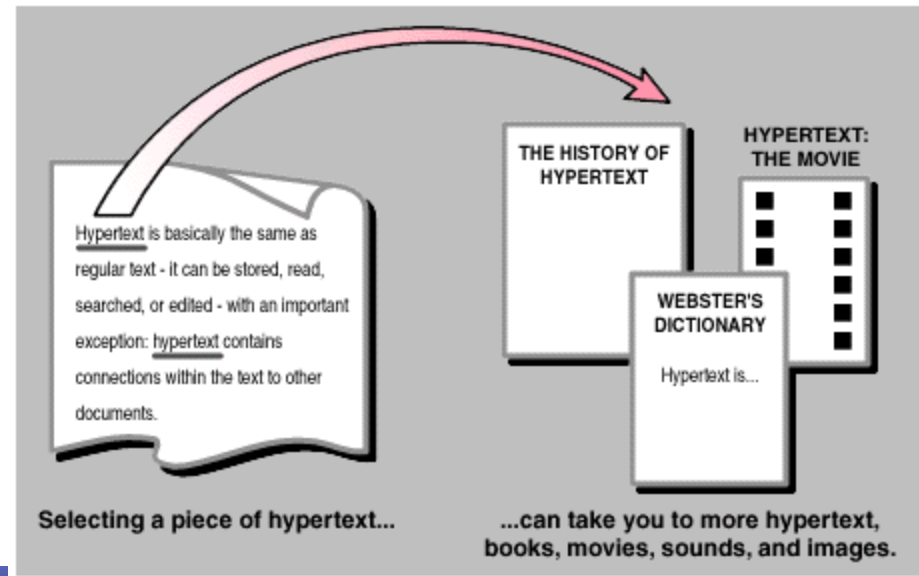


H1. Hypertext

1. Khái niệm về siêu văn bản

■ Ứng dụng

- Các trình ứng dụng dùng siêu văn bản rất hữu ích trong trường hợp phải làm việc với số lượng văn bản lớn...như các bộ từ điển bách khoa và các bộ sách nhiều tập..



2. Các thành phần của Web

- Web (World Wide Web)?

Web là hệ thống các tài liệu hypertext liên kết với nhau thông qua mạng Internet

- Các thành phần của Web

- Web Page
- Web Site
- Uniform Resource Locator
- Web Server
- Web Browser



2.1. Máy phục vụ, máy khách

Máy khách (Client)

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



2.1. Máy phục vụ, máy khách

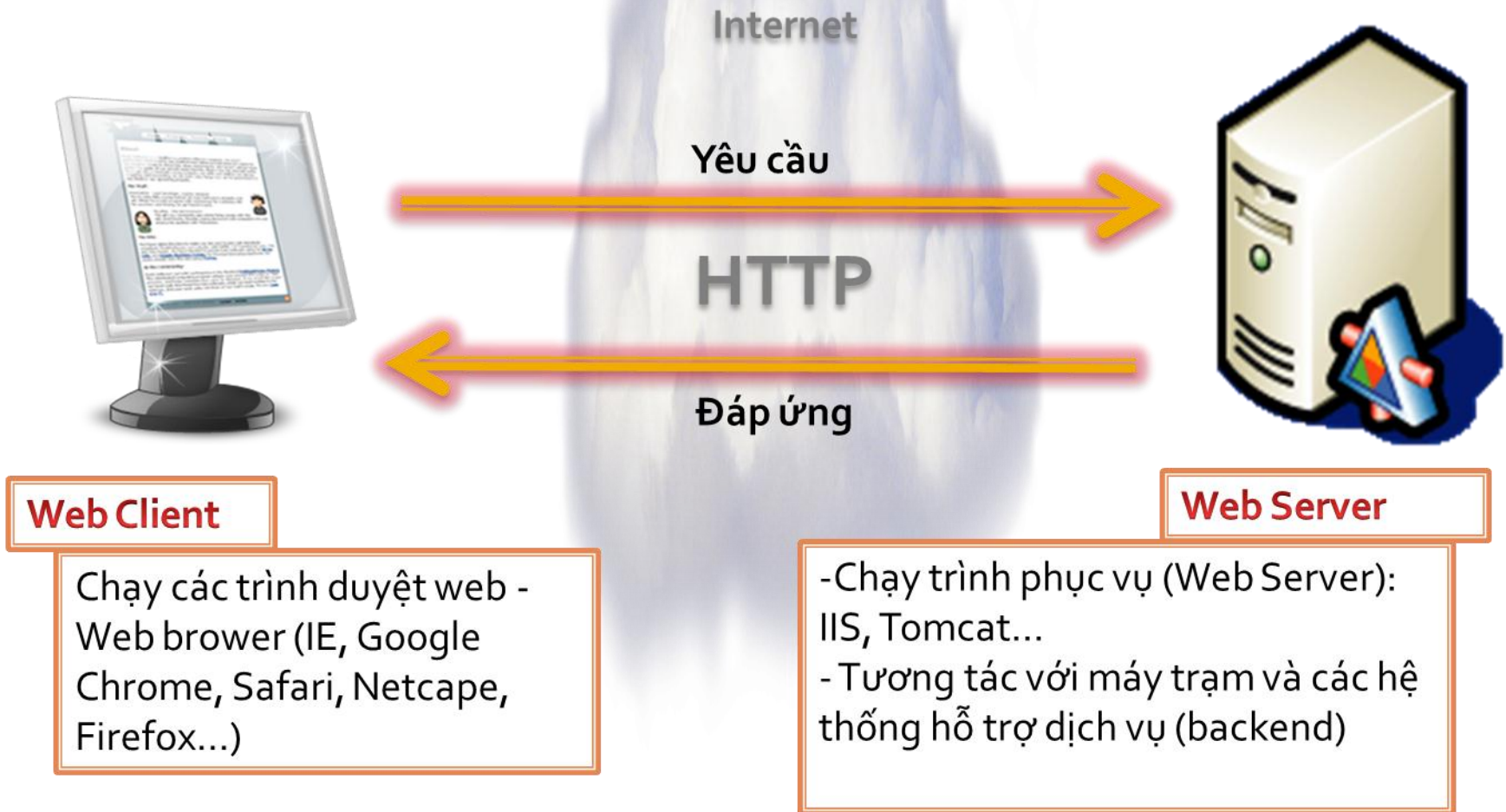
Máy phục vụ (Server)

- Là máy chuyên cung cấp các dịch vụ và tài nguyên cho các máy tính khác
- Thường cài các phần mềm chuyên dụng
- Một máy phục vụ có thể dùng cho một hay nhiều mục đích.
 - *File server*
 - *Application server*
 - *Mail server*
 - ***Web server***



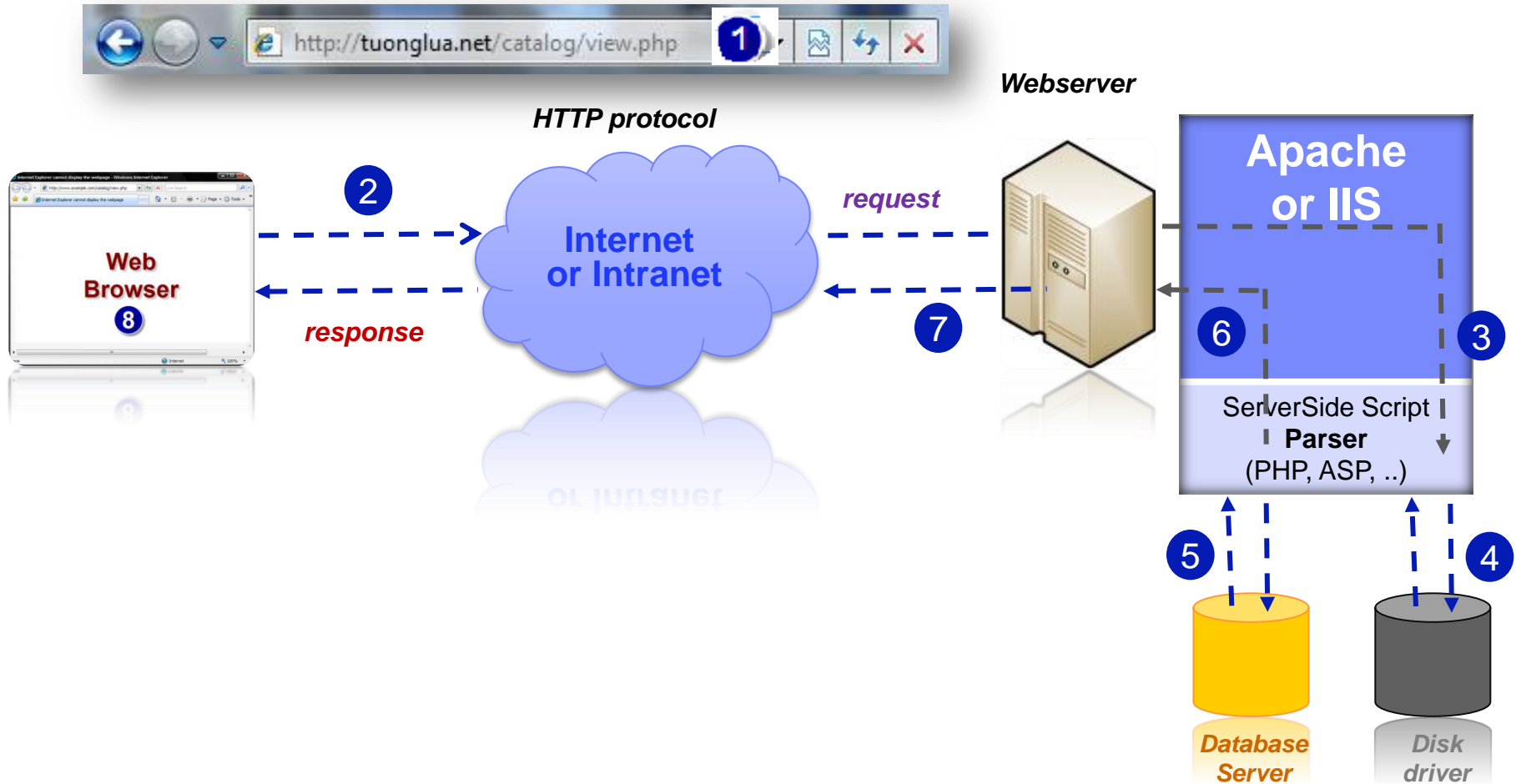
2.2. Web Server, Web Browser

▪ Kiến trúc ứng dụng Web



2.2. Web Server, Web Browser

- Kiến trúc ứng dụng Web



2.2. Web Server, Web Browser

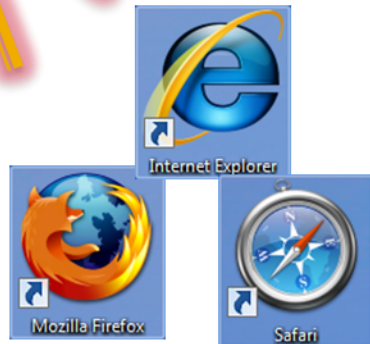
Web Browser

Web Server



Trang web dưới dạng (X)HTML, kịch bản JavaScript, CSS....

Yêu cầu



Web Browser



Xuất kết quả ra màn hình

Web browser là giao diện người sử dụng phía máy trạm dùng để:

- Gửi yêu cầu và nhận nội dung trang web từ Web server
- Thông dịch trang HTML, CSS và thực thi các kịch bản JavaScript nhận được
- Xuất kết quả ra màn hình



2.2. Web Server, Web Browser

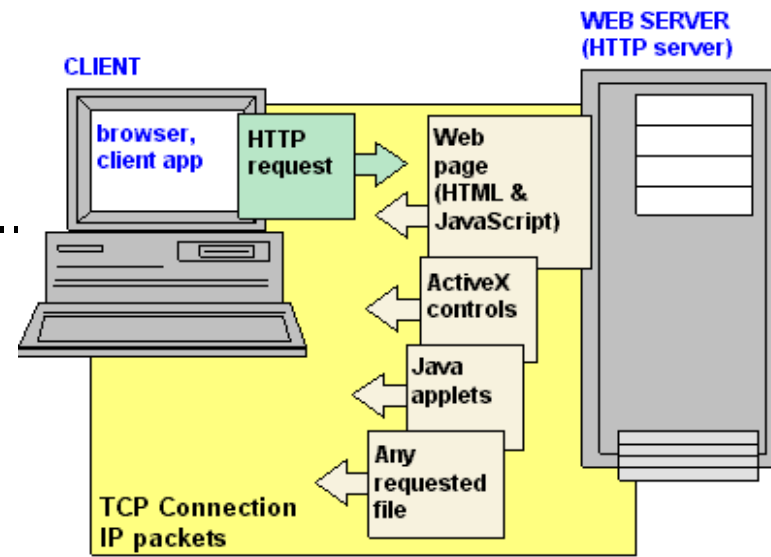
■ Web Server

- Là thành phần chính của hệ thống dịch vụ
- Nhận yêu cầu từ Web Clients qua Web Browsers
- Sinh và gửi nội dung trang web dưới dạng HTML, CSS và các đoạn mã kịch bản JavaScripts
- Tương tác với hệ thống hỗ trợ dịch vụ (backend system)
- Thường là các hệ quản trị cơ sở dữ liệu

2.2. Web Server, Web Browser

■ Web Server

- Mỗi Web Server phục vụ một số kiểu file chuyên biệt chẳng hạn như IIS của Microsoft dành cho *.asp, *.aspx...;
- Apache dành cho *.php...;
- Tomcat server dành cho *.jsp...



2.3. WebPage, WebSite

- **WebPage:**
 - Là một trang Web
 - Ngôn ngữ viết: PHP, ASP, ASP.Net... nhưng kết quả trả về cho Client là HTML
- **Website:**
 - Là tập hợp các WebPage có nội dung thống nhất



2.3. WebPage, Website

- Phân cấp trang Web



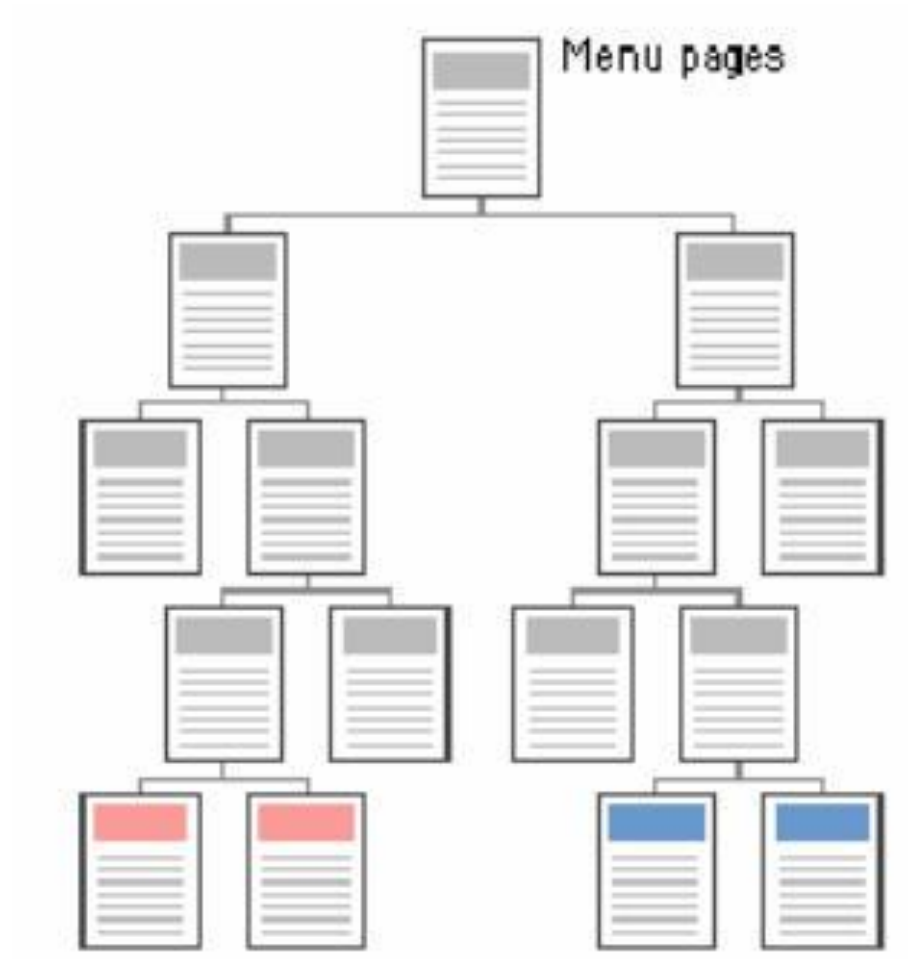
2.3. WebPage, Website

- Phân cấp quá “nông”



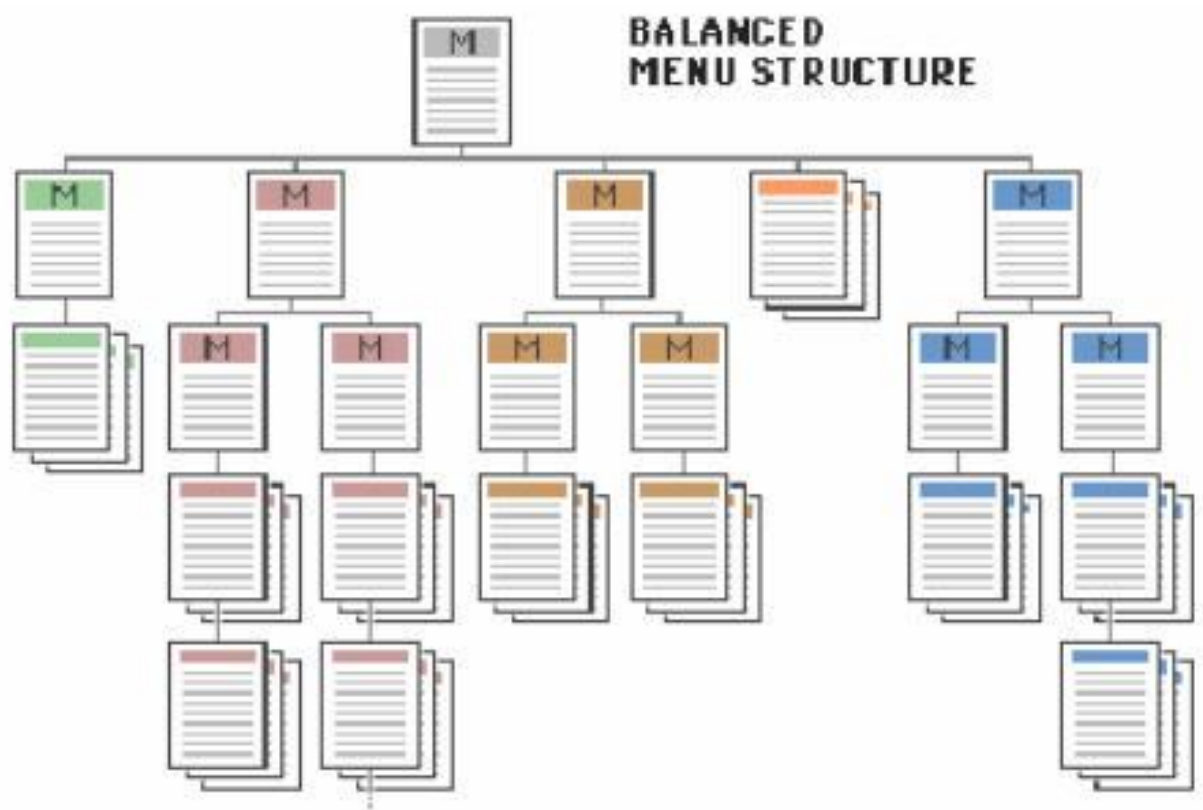
2.3. WebPage, Website

- Phân cấp quá “sâu”



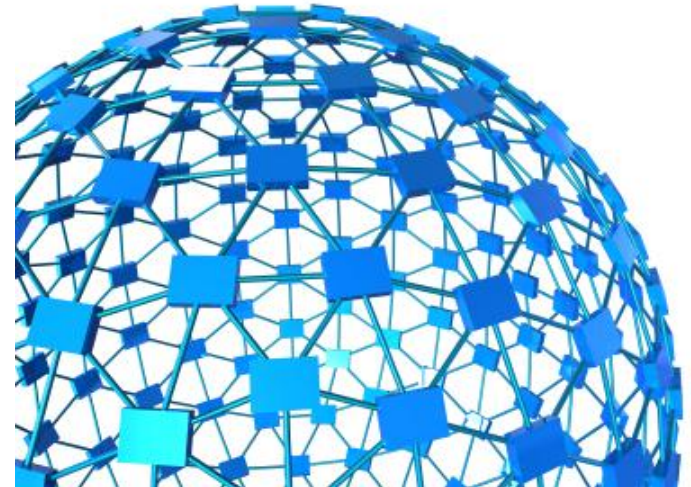
2.3. WebPage, Website

- Phân cấp hợp lý



2.4. URL (Uniform Resource Locator)

- Địa chỉ IP (IP Address)
 - Xác định một máy tính trong mạng dựa trên giao thức TCP/IP. Hai máy tính trong mạng có 2 địa chỉ IP khác nhau
 - Có dạng x.y.z.t ($0 \leq x, y, z, t \leq 255$)
 - Ví dụ: google Việt Nam?
74.125.127.106
 - Địa chỉ Loop Back: 127.0.0.1



2.4. URL (Uniform Resource Locator)

- Tên miền (Domain name)
 - Tại sao lại cần?
 - Là tên gắn liền với một địa chỉ IP
 - Ở dạng văn bản, thân thiện với người dùng
 - Đánh số cấp từ bên phải, bắt đầu từ 1
 - Ex: www.soict.hut.edu.vn
 - Localhost: 127.0.0.1



2.4. URL (Uniform Resource Locator)

- Cổng dịch vụ (Service Port)
 - Tại sao lại cần?
 - Một Server có thể cung cấp nhiều dịch vụ => cần sử dụng cổng để xác định dịch vụ cung cấp
 - Mỗi dịch vụ thường chiếm những cổng mặc định
 - HTTP: 80, FTP: 21, SMTP: 25, POP3: 110



.name
.jp .org .com .net .cc .CN
.info .ca .biz
.us .co.uk

2.4. URL (Uniform Resource Locator)

- Là chuỗi định vị tài nguyên trên Internet

Ví dụ: `http://is.hut.vn:8080/hpsg/index.js`



- Cấu trúc

giao thức://địa chỉ máy:cổng/đường dẫn đến tài nguyên

- Nhiều tham số có giá trị mặc định => có thể bỏ qua

2.5. Giao thức HTTP

■ HTTP - (HyperText Transfer Protocol)

- Là giao thức được dùng để truyền tải dữ liệu dạng HTML, XML trên môi trường mạng (World Wide Web)
- Là giao thức không trạng thái: 1 giao tác chỉ gồm 1 yêu cầu và 1 đáp ứng yêu cầu đó.
- Khi một trình duyệt (client) kết nối tới một web server nó sẽ gửi một HTTP Request tới web server
- Web Server sau khi nhận và xử lý yêu cầu sẽ gửi một HTTP Response – đáp ứng lại cho Client

2.5. Giao thức HTTP

▪ HTTP Request

- Một HTTP Request được gửi từ Client đến Web Server là một tệp có định dạng:
 - Dòng đầu tiên được gọi là đầu yêu cầu (**request header**) chứa ba thông số:
 - Phương thức yêu cầu (request method)
 - URL
 - Phiên bản HTTP được sử dụng
 - Các dòng tiếp theo chứa thông tin về các kiểu tệp, tập ký tự được chấp nhận, phiên bản trình duyệt, hệ điều hành sử dụng trên client, ...

2.5. Giao thức HTTP

- **HTTP Request:** Ví dụ: Một HTTP Request được gửi từ Client:

Phương thức: GET
URL: /index.html
Phiên bản HTTP: 1.1

Các thông tin mô tả khác

Request from Client Browser

```
GET /index.html HTTP/1.1  
Host: www.example.com  
Accept-Language: en
```

2.5. Giao thức HTTP

▪ HTTP Request

Dòng yêu cầu (request line):

`<method> <uri> <version>`

- `<version>` là phiên bản HTTP của yêu cầu (HTTP/1.0 or HTTP/1.1)
- `<uri>` thường là URL trên server
- `<method>` có thể nhận các giá trị GET, POST, OPTIONS, HEAD, PUT, DELETE, hoặc TRACE.

2.5. Giao thức HTTP

- **HTTP Request: Các phương thức HTTP**
 - **GET: Truy vấn thông tin tĩnh hoặc động**
 - Các tham số cho nội dung động nằm trong URI
 - **POST: Truy vấn thông tin động**
 - Các tham số cho nội dung động nằm trong request body
 - **OPTIONS: Lấy các thuộc tính của server hoặc tệp tin**
 - **HEAD: Tương tự như GET nhưng không có dữ liệu trong response body**
 - **PUT: Ghi một tệp tin lên server**
 - **DELETE: Xoá một tệp tin trên server**
 - **TRACE: lặp lại request trong response body**
 - Hữu dụng cho việc gỡ lỗi (debugging).

2.5. Giao thức HTTP

▪ HTTP Request: *Phương thức GET*

- Là phương thức thường xuyên được sử dụng khi trình duyệt gửi yêu cầu.
- Nếu Client không chỉ rõ phương thức được sử dụng thì mặc định được hiểu là sử dụng phương thức GET
- Chúng ta sử dụng phương thức GET khi yêu cầu một trang web (web page)

2.5. Giao thức HTTP

▪ HTTP Request: Phương thức GET

- Khi sử dụng phương thức GET:
 - Các tham số và giá trị tham số (nếu có) được nối vào với URL
 - Chuỗi tham số này cũng được hiển thị trên address bar của trình duyệt
 - Chiều dài của URL là có hạn do vậy chuỗi tham số cũng bị giới hạn.
- Ví dụ:

Tham số = giá trị



<http://www.soict.hust.edu.vn:73/Khoa.aspx?makhoa=1>

2.5. Giao thức HTTP

▪ HTTP Request: *Phương thức POST*

- Là phương thức cũng được sử dụng khi trình duyệt gửi yêu cầu.
- Chúng ta sử dụng phương thức POST khi cần gửi dữ liệu để xử lý
- Khi sử dụng phương thức POST:
 - Các cặp tham số/giá trị được nối vào HTTP request và được che dấu.
 - Không giới hạn số lượng tham số
 - Các tham số không được hiển thị trên Address bar của trình duyệt.

2.5. Giao thức HTTP

▪ HTTP Response

- . Dữ liệu do server gửi về cho client được định dạng bởi HTTP Response
- Một HTTP Response bao gồm
 - *Dòng trạng thái (status line)*: Giao thức được dùng, mã trạng thái và giá trị trạng thái
 - *Đầu đáp ứng (response header)*: Chứa chuỗi các cặp tên/giá trị,
 - *Dữ liệu thực sự*: Trang HTML.

2.5. Giao thức HTTP

▪ HTTP Response

- Ví dụ: một HTTP Response được gửi từ web server

Phiên bản HTTP: 1.1
Mã trạng thái: 200
Mô tả trạng thái: OK

Response Headers: mô tả các thông tin về Web Server và Web Page

Nội dung trang HTML được gửi về Browser

Response from Web Server

```
HTTP/1.1 200 OK
Date: Mon, 23 Oct 2006 12:05:20 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Content-Type: text/html; charset=utf-8
Content-Length: 194

<html>
  <head>
    <title>Welcome</title>
  </head>
  <body>
    <h1>Welcome to Example.com</h1>
```

2.5. Giao thức HTTP

▪ HTTP Response

Dòng phản hồi (response line):

<version> <status code> <status msg>

- **<version>** là phiên bản HTTP của phản hồi.
- **<status code>** mã trạng thái (dạng số).
- **<status msg>** thông điệp tương ứng với mã trạng thái.
 - 200 OK
 - 403 Forbidden
 - 404 Not found

Response headers: <header name>: <header data>

- Cung cấp các thông tin bổ sung về phản hồi
- **Content-Type:** MIME type của nội dung trong response body.
- **Content-Length:** chiều dài của nội dung response body.

2.5. Giao thức HTTP

- HTTP status codes: 1xx (Phản hồi tạm thời) - 2xx (Truy cập thành công)

Mã	Mô tả
100 : (Continue)	Người truy cập cần tiếp tục với yêu cầu truy cập.
101 : (Switching protocols)	Khách truy cập đã yêu cầu máy chủ chuyển đổi giao thức truy cập và máy chủ phản hồi rằng đã nhận được yêu cầu và xác nhận chuyển đổi giao thức truy cập.
200 : (Successful)	Các máy chủ xử lý yêu cầu thành công.
202 : (Accepted)	Thông báo rằng nó đã nhận được phần đầu của một yêu cầu truy cập và đang chờ đợi nhận được phần còn lại.
204 : (No content)	Các máy chủ xử lý yêu cầu thành công, nhưng không phản hồi bất kỳ nội dung gì tới người dùng.

2.5. Giao thức HTTP

- HTTP status codes: 3xx (truy cập bị chuyển hướng) - 4xx
(Phát sinh lỗi trong yêu cầu truy cập)

Mã	Mô tả
400 : (Bad request)	Máy chủ không hiểu được cú pháp của yêu cầu.
401 : (Not authorized)	(Không được uỷ quyền) Yêu cầu truy cập cần được chứng thực.
403 : (Forbidden)	Máy chủ từ chối các yêu cầu truy cập.
404 : (Not found)	Máy chủ không thể tìm thấy trang yêu cầu.
408 : (Request timeout)	Máy chủ đã hết thời gian để chờ nhận yêu cầu truy cập.

2.5. Giao thức HTTP

■ HTTP status codes: 5xx (Lỗi server)

Mã	Mô tả
500 : (Internal server error)	Máy chủ đã gặp lỗi và không thể hoàn tất yêu cầu.
502 : (Bad gateway)	Nhận được phản hồi không hợp lệ.
503 : (Service unavailable)	Máy chủ hiện tại không thể thực hiện yêu cầu (vì bị quá tải hoặc đang phải bảo trì).
504 : (Gateway timeout)	Hết thời gian.
505 : (HTTP version not supported)	Các máy chủ không hỗ trợ phiên bản giao thức HTTP được sử dụng trong các yêu cầu.

3. Phân loại trang Web

- Web tĩnh
 - Sử dụng HTML
 - Tương tác yếu
- Web động
 - Kết hợp mã HTML và mã lệnh
 - Tương tác mạnh
 - Mã lệnh được thực thi trên Server, trả về HTML cho Client

3. Phân loại trang Web

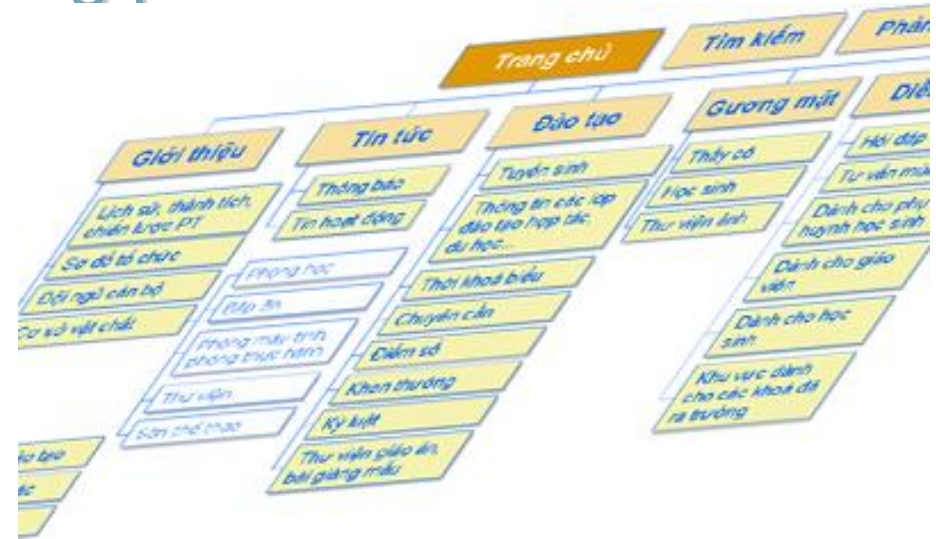
- So sánh
- Web tĩnh
 - Dễ phát triển
 - Nội dung thường cố định trên Browser
- Web động
 - Khó phát triển hơn
 - Có thể thay đổi nội dung trên Browser
 - Các ngôn ngữ sử dụng đa dạng: PHP, ASP, ASP.NET..

4. Một số điều cần chú ý trong phát triển Website

- Phân tích thiết kế HTTT
 - Nhận yêu cầu và tiến hành khảo sát
 - Đặc tả
 - Phân tích
 - Thiết kế
 - Lập trình
 - Kiểm thử

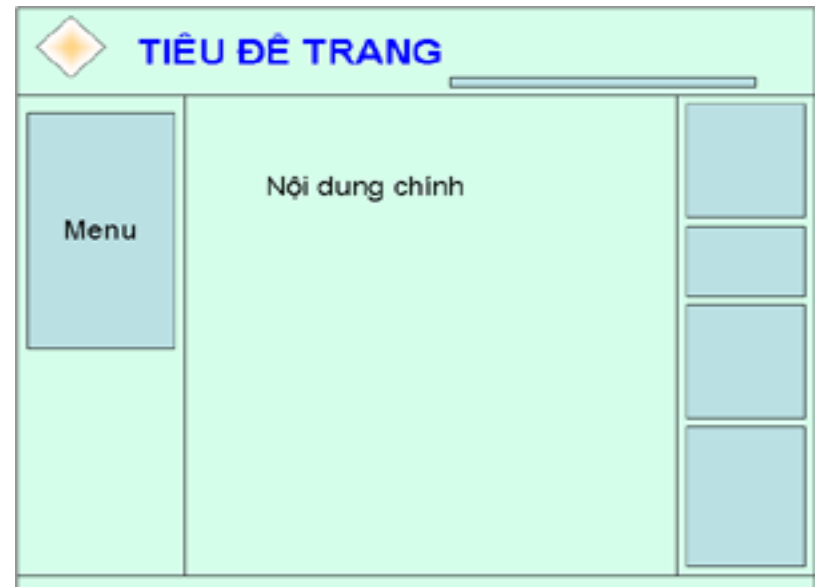
4. Một số điều cần chú ý trong phát triển Website

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



4. Một số điều cần chú ý trong phát triển Website

- **Thiết kế**
 - Sơ đồ cấu trúc website
 - Giao diện
 - Tĩnh hay động
 - CSDL
 - Nội dung từng trang
 - Liên kết giữa các trang
- **Xây dựng**
 - Cấu trúc thư mục
 - Các modul dùng chung
 - ...



4. Một số điều cần chú ý trong phát triển Website

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



4. Các tiêu chí thiết kế trang Web

- 1. Hãy đơn giản trong thiết kế
- 2. Nên tạo các đường link rõ ràng
- 3. Định hướng minh bạch
- 4. Đưa ra hướng dẫn cho người sử dụng (NSD)
- 5. Cung cấp cho NSD những thứ họ cần
- 6. Hạn chế tối đa thời gian download
- 7. Tránh những đăng ký không cần thiết
- 8. Sử dụng màu sắc hài hòa
- 9. Font chữ sử dụng

1. Hãy đơn giản trong thiết kế

**Bạn sử dụng dễ dàng
website của tôi?**

2. Tạo các đường link rõ ràng

Đây là cái gì?

3. Định hướng minh bạch

Tôi đang ở đâu?

4. Đưa ra hướng dẫn cho NSD

**Tôi không hiểu tôi
phải làm như thế
nào?**

5. Cung cấp cho NSD những thứ họ cần

**Người sử dụng muốn
điều gì?**

6. Hạn chế tối đa thời gian download

Bạn có thể chờ tôi bao lâu?

7. Tránh những mẫu đăng ký không cần thiết

Bạn có thích đăng kí không?

8. Sử dụng màu sắc hài hòa

**Bạn có phải là một
họa sĩ?**

9. Font chữ sử dụng

Đó có phải là một font
chữ đặc biệt?

10. Others

- Thêm khoảng trống
- Thêm người dùng

11. Principles of Beautiful web design

- 1. Bố trí và kết hợp (Layout & Composition)
- 2. Màu (Color)
- 3. Kết cấu (Texture)
- 4. Tạo chữ (Typography)
- 5. Hình ảnh (Imagery)