



Bài 1

FORM, JS & CSS

iNET Academy

[.danang@inet.vn](mailto:danang@inet.vn)

[.hanoi@niit.vn](mailto:hanoi@niit.vn)

[.hcm@niit.vn](mailto:hcm@niit.vn)





NỘI DUNG:

- 1. FORM**
- 2. Script**
- 3. CSS**
- 4. Website design guide**
- 5. HTML 5**





FORM



1. FORM





FORM » Definition

- ❖ Form là một thành phần quan trọng của HTML; thu thập nhiều loại dữ liệu khác nhau của người dùng, bắt đầu và kết thúc bởi cặp thẻ form.
- ❖ Thẻ `<form>`:

```
<form method = <GET/POST> action = "URL"> ...  
</form>
```

 - ❖ Các **trường** thu thập thông tin từ người dùng đặt giữa thẻ mở form và đóng form;
 - ❖ Giá trị method bao gồm GET và POST định nghĩa cách thức truyền dữ liệu
 - ❖ Action: nơi gửi dữ liệu đến khi form được submit
 - ❖ Phía máy chủ cần có **URL** nhận dữ liệu và xử lý dữ liệu đó





FORM » Sample

- ❖ Ví dụ về form: trang đăng ký của Facebook

Đăng Ký
Hoàn toàn miễn phí, và sẽ luôn như thế.

Tên:

Họ:

Email:

Nhập lại địa chỉ email:

Mật Khẩu Mới:

Giới Tính:

Ngày Sinh: Ngày: Tháng: Năm:

Tại sao tôi cần phải cung cấp thông tin này?

Tạo Trang cho người nổi tiếng, ban nhạc hoặc kinh doanh.





FORM » Textarea

```
<textarea Name = "name" Rows = number  
          Cols = number Wrap = value>
```

Nội dung hiển thị trong ô text

```
</textarea>
```

- ❖ Cho phép người dùng nhập nhiều dòng văn bản vào Form với số dòng và số cột cần hiển thị.
- ❖ Text: Hướng dẫn người sử dụng nhập dữ liệu.
- ❖ Value: OFF (giá trị mặc định) nếu không dùng Wrap.





FORM » Textarea » Sample

```
<form action="">
```

```
<textarea rows="5" cols="40">
```

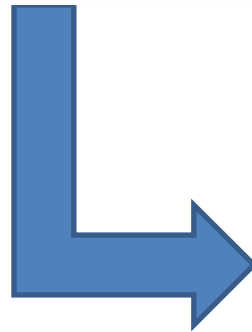
Cơ hội nghề nghiệp cho sinh viên NIIT

Mr. Nguyễn Trọng Thơ - CEO

Công ty Cổ phần iNET.

```
</textarea>
```

```
</form>
```



Your Result:

Ví dụ sử dụng textarea

Cơ hội nghề nghiệp cho sinh viên NIIT
Mr. Nguyễn Trọng Thơ - CEO
Công ty Cổ phần iNET.





FORM » input text

`<Input Type = "text" Name = "tên trường" Size =number MaxLength= number>`

- ❖ Với **type=text**, ta có ô nhập văn bản cho phép người dùng nhập một đoạn văn bản có chiều dài Size và chiều dài tối đa cho đoạn văn bản là MaxLength (Size < MaxLength)





FORM » input password

```
<Input Type = "password"  
Name = "tên trường"  
Size = number  
MaxLength= number />
```

- ❖ Tương tự type = text. Dùng để nhập loại ký tự mật khẩu, khi nhập dữ liệu vào thì các ký tự được dấu đi và thay vào đó là những ký tự "*" hay "x" tùy thuộc vào trình duyệt.





FORM » input CheckBox

```
<input type="checkbox" name="tên dữ liệu" value="Giá trị 1" /> Hiện thị 1<br />  
<input type="checkbox" name="tên dữ liệu" value="Giá trị 2" /> Hiện thị 2<br />
```

- ❖ Tùy chọn này được dùng khi có nhiều giá trị cho một tùy chọn.
- ❖ Muốn xác định trạng thái mặc định của checkBox là đánh dấu hoặc không đánh dấu thì thêm một trong hai giá trị: **Checked** hoặc **Unchecked**





FORM » input Radio

```
<Input Type = "radio" Name = "tên dữ liệu" Value = "Giá trị 1"> String 1<br />  
<Input Type = "radio" Name = "tên dữ liệu" Value = "Giá trị 2"> String 2<br />  
<Input Type = "radio" Name = "tên dữ liệu" Value = "Giá trị 3"> String 3<br />
```

- ❖ Cho phép người dùng lựa chọn trong các tùy chọn được định trước.
- ❖ Thuộc tính **Name** phải giống nhau và thuộc tính **Value** phải khác nhau





FORM » input Submit

```
<input type="submit" value="tên nút bấm">
```

❖ **Hiển thị nút bấm**

- ✓ Dùng để chuyển dữ liệu trên Form mà người sử dụng đã nhập sang một trang mới.
- ✓ String: Là dòng chữ ghi trên nút, nếu không có Value thì mặc định là Submit ghi trên nút





FORM » input Reset

```
<input type="reset" value="tên nút bấm">
```

❖ **Hiển thị nút bấm xóa dữ liệu trong form**

- ✓ Dùng để xóa dữ liệu trên Form mà người sử dụng đã nhập, khởi động lại cho các phần tử trên Form.
- ✓ **String**: Là dòng chữ ghi trên nút, nếu không có **Value** thì mặc định là **Reset** ghi trên nút





FORM » input Hidden

```
<input type="hidden" name="tên dữ liệu" value="Giá trị" />
```

- ❖ Được thiết kế để truyền (ngầm) giá trị đến Web Server và Script. Giá trị truyền thường là một từ khoá, giá trị kiểm tra hay một chuỗi bất kỳ





FORM » input Select

```
<select name="tên dữ liệu">  
  <option value="Giá trị 1">Giá trị 1</option>  
  <option value="Giá trị 2">Giá trị 2</option>  
  <option value="Giá trị n">Giá trị n</option>  
</select>
```

- ❖ **<Option>**: Định nghĩa một phần tử trong danh sách.
- ❖ **Multiple**: Cho phép người dùng chọn cùng một lúc nhiều giá trị.





FORM » input sample

```
<form action="form_action.asp" method="get">  
  Tài khoản:<input type="text" name="username" /> <br />  
  Mật khẩu:<input type="text" name="passwd" /> <br />  
  <input type="checkbox" name="remember">Nhớ mật khẩu<br />  
  Bạn là:<br />  
  <input type="radio" name="usertype" value="longterm">  
    Sinh viên dài hạn<br />  
  <input type="radio" name="usertype" value="shortterm">  
    Sinh viên ngắn hạn<br />  
  <input type="radio" name="usertype" value="iNET">  
    Nhân viên iNET<br />  
  <input type="submit" value="đăng nhập">  
</form>
```



Your Result:

Tài khoản:

Mật khẩu:

☐ Nhớ mật khẩu

Bạn là:

- ☒ Sinh viên khóa dài hạn
☐ Sinh viên khóa ngắn hạn
☐ Nhân viên iNET

Click nút đăng nhập thì dữ liệu được nhập ở form sẽ được gửi đến trang "form_action.asp" trên server.





SCRIPT

```
JS
function form(form){
  if(notEmpty(form.nm)
  {
    return true;
  }
  return false;
}
```

2. SCRIPT





2. SCRIPT

- ❖ Thẻ **<script>** được sử dụng để định nghĩa client-side script; đó là đoạn mã lệnh được chạy tại phía người sử dụng để tác động lên nội dung của trang HTML.
- ❖ Có thể sử dụng VBscript, JavaScript
- ❖ Mục đích để: xử lý hình ảnh, kiểm tra dữ liệu nhập trong form, thay đổi thông tin trang HTML tự động.
- ❖ Script được gọi trong phần body; trên các sự kiện của form như: OnClick, OnFocus, OnChange, ...
- ❖ Chi tiết về ngôn ngữ Script sẽ được trình bày ở các khóa học lập trình web.

```
<script type="text/javascript">  
    document.write("Hello World!")  
</script>
```





SCRIPT » clock sample

```
<html>
<head>
<script language="javascript">
    var dateform;
    speed=1000;
    len=28;
    tid = 0;
    function dodate(){
        f.date.value=new Date();
        tid=window.setTimeout("dodate()",speed);
    }
    function start(x){
        f=x;
        tid=window.setTimeout("dodate()",speed);
    }
    function cleartid() {
        window.clearTimeout(tid);
    }
}
```



```
function Clock(){
    document.write('<FORM name=dateform>');
    document.write('<input name=date size=');
    document.write(len);
    document.write(' value="Demo Script"></FORM>');
    start(document.dateform);
}
</script>
</head>

<body>
    <script language="javascript">
        Clock();
    </script>
</body>

</html>
```



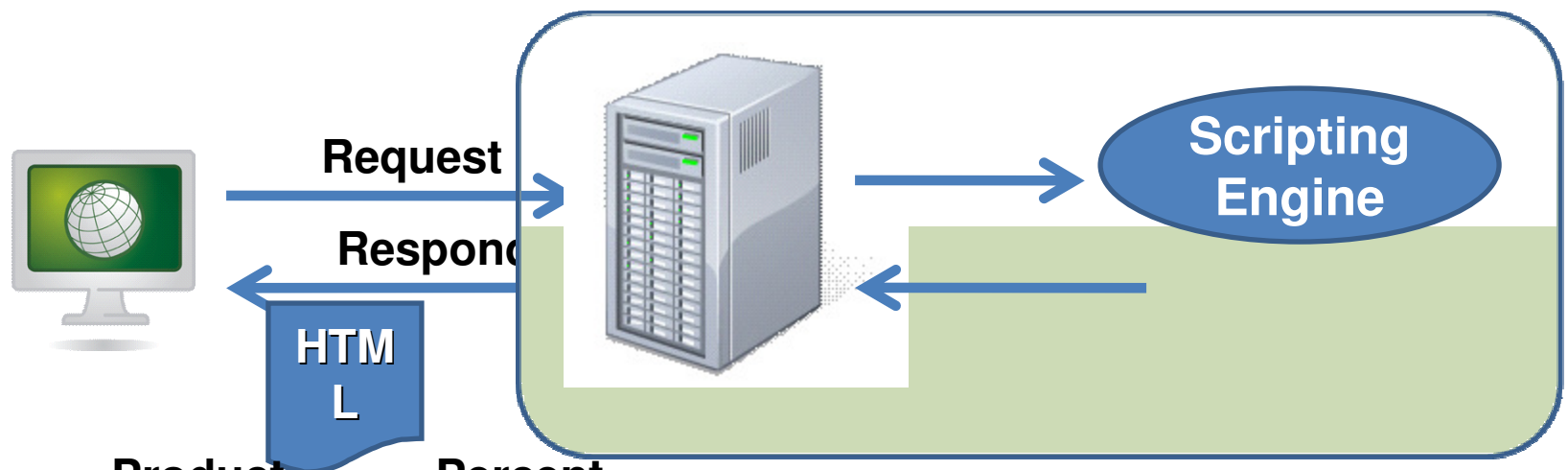
Wed Dec 15 2010 02:42:24 GMT+





SCRIPT » Server-side scripting

- ❖ Server-side scripting là công nghệ web cho phép việc thực thi các script trực tiếp trên máy chủ để trả kết quả là html về cho người dùng



Vendor	Product	Percent
Apache	Apache	59.36%
Microsoft	IIS	22.70%
Igor Sysoev	nginx	6.04%
Google	GWS	5.94%
lighttpd	lighttpd	0.83%



SCRIPT » Server-side scripting

- ❖ Các ngôn ngữ Script phía server phổ biến:
 - ✓ PHP
 - ✓ ASP, ASP .NET
 - ✓ JSP (java) và JavaScript
- ❖ Việc chạy Script trực tiếp trên server giúp cho các ứng dụng web trong suốt với người dùng; cung cấp trang web động mạnh mẽ và hiệu quả.

```
<html>
  <body bgcolor="#FFFFFF">
    <asp:label id="HelloWorld" runat="server" />
  </body>
</html>
```

```
<html>
  <body>
    <?php
      echo "Hi, I'm a PHP script!";
    ?>
  </body>
</html>
```





3. Cascading Style Sheets





CSS » definition

- ❖ Ngôn ngữ định kiểu theo cấu trúc được dùng để miêu tả **cách trình bày** các tài liệu viết bằng ngôn ngữ HTML và XHTML.
- ❖ CSS định nghĩa các kiểu định dạng của trang web: font, màu, cỡ chữ, border, margin, ...
- ❖ Được lưu thành file .css hoặc tích hợp luôn vào file html.

`Đoạn text cần in đậm, gạch chân, màu đỏ`





❖ Lợi ích của CSS:

- Thẩm mỹ
- Giữ tính thống nhất cho trang HTML.
- Định dạng một số tính chất thông thường cùng một lúc cho tất cả các đối tượng trên trang
- Tiết kiệm thời gian
- Khi thay đổi định dạng chỉ cần thay đổi CSS, các trang khác sẽ tự động cập nhật sự thay đổi đó
- Có thể dùng các CSS cùng với JavaScript để tạo các hiệu ứng đặc biệt

❖ Bất lợi của CSS:

- Không một trình duyệt nào chấp nhận nó hoàn toàn
- Phải mất thời gian để học cách sử dụng





- ❖ CSS Selector là linh hồn của CSS; Selector có quan hệ trực tiếp đến HTML tags.

SELECTOR { PROPERTY: VALUE }

- ❖ Selector name là các HTML tags: <p>, , ...
- ❖ Property: là các thuộc tính kiểu dáng.
- ❖ Value: là giá trị của property.





CSS » Inline CSS

- ❖ Inline CSS là cách dùng CSS bằng cách áp dụng trực tiếp trên một đối tượng nhất định bằng thuộc tính **style**

```
<TagName style="property1:value1;property2: value2; ...">  
    Nội dung văn bản muốn định dạng  
</TagName>
```

```
<span style="font-weight:bold;text-decoration:underline;color:#FF0000;">  
Đoạn text cần in đậm, gạch chân, màu đỏ</span>
```





CSS » Internal CSS

- ❖ Là bảng mẫu thích hợp cho trang riêng lẻ.
 - ✓ Cách tạo: Tạo bảng CSS *trên phần đầu trang* trong cặp tag `<head>`
 - ✓ Sử dụng: Trong phần body, nội dung nào muốn sử dụng định dạng theo bảng mẫu trên thì đặt trong tag được định nghĩa trong phần head

```
<html>
<head>
<style type="text/css">
    h1,h2 { color: limegreen;
           font-family: arial }
</style>
</head>
<body>
    <h1>this is the H1 element</h1>
    <h2>this is the H2 element</h2>
    <h3>this is the H3 element with
        its default style as displayed in
        the browser</h3>
</body>
</html>
```





CSS » External CSS

- ❖ Đây là loại thông dụng nhất, đúng với mục đích ra đời của CSS.
- ❖ Là một bảng kiểu được lưu trữ thành một file bên ngoài và được liên kết với trang HTML.
- ❖ Bảng kiểu này sẽ được áp dụng và ảnh hưởng cho tất cả các trang của một website

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```
</head>
```

```
<body>
```

```
    <h2>this is the H2 element</h2>
```

```
</body>
```





CSS » External CSS » Create

❖ Cách tạo:

- ✓ Tạo một tập tin văn bản mới
- ✓ Nhập tên các tag muốn định dạng thuộc tính theo mẫu:
`TagName{property1: value1; property2:value2;...}`
- ✓ Lưu tập tin với định dạng Text Only và có phần mở rộng `.css`
- ✓ Để dùng các style trong CSS thì ta phải link file css vào file html:
`<link rel="stylesheet" type="text/css" href="link to CSS">`

style.css

```
body {  
  background: #ffffff;  
  font-family: Verdana;  
  color: #ff0000;  
}  
  
h2 {color:blue; font-style:italic}  
  
p{text-align:justify; text-indent:8pt;  
font:10pt/15pt "Verdana"}
```

**.html*

```
<html>  
<head>  
<link rel="stylesheet" type="text/css" href="style.css">  
</head>  
<body>  
  <h2>this is the H2 element</h2>  
  <p>this is the P element</p>  
</body>  
</html>
```





CSS » Priority order

- ❖ Trong trường hợp dùng nhiều loại CSS cùng lúc, các CSS ở mức thấp (trong cùng) sẽ được ưu tiên hơn cả

(ưu tiên cao) Inline → Internal → External (ít ưu tiên)

- ✓ Nên dùng External CSS cho toàn bộ Website
- ✓ Đối với các trang web cần style riêng thì sử dụng Internal CSS
- ✓ Đối với các tags cần style khác với style chung thì dùng Inline CSS





CSS » Class selector

- ❖ Sử dụng Class giúp ta có thể áp dụng nhiều style khác nhau cho 1 loại thẻ
- ❖ Cú pháp:

```
Selector.ClassName{property1: value1; property2:value2;...}
```

```
p{ color: red; font-size: 20px; }
```

```
p.first{ color: blue; }
```

```
p.second{ font-size: 12px; }
```

- ❖ Khi đó trong thẻ cần dùng style ta thêm thuộc tính Class="ClassName"

```
p.first{ color: blue; font:10pt/15pt "Verdana" }
```

```
p.second{ color: red; font:20pt/15pt "Tahoma" }
```

```
<p class="second">this is the P element</p>
```





CSS » ID selector

- ❖ Dùng áp dụng cho một phần tử riêng biệt trên trang Web (phần tử nào ít dùng mà cần thay đổi style thì áp ID)
- ❖ Cú pháp:

`TagName#IDName{property1: value1; property2:value2;...}`

- ❖ Trong tag Body nhập cú pháp:

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

- ❖ Nói chung tương tự Class, tuy nhiên nếu chỉ sử dụng 1 lần trong văn bản thì dùng ID, còn nếu dùng nhiều lần thì nên dùng class





CSS » ID/Class Sample

```
p{ color: red; font-size: 20px; }  
p.first{ color: blue; }  
p.second{ font-size: 12px; }  
p#special{font-style:italic; }
```

<p>this is the P element</p>

<p class="first">this is the P.First element</p>

<p class="second">this is the P.Second element</p>

<p id="special">this is the P#Special element</p>



this is the P element

this is the P.First element

this is the P.Second element

this is the P#Special element





CSS » Property

- ❖ Background
- ❖ Font
- ❖ Text
- ❖ Box





CSS » Property » Background

- ❖ background-color: định nghĩa màu nền của Selector
- ❖ background-image: dùng hình ảnh để làm nền
- ❖ background-repeat: lặp hình ảnh
- ❖ background-attachment: cố định hình nền hoặc scroll
- ❖ background-position: đặt vị trí của hình nền

```
h1{  
    background-color: rgb( 149, 206, 145);  
    color: red;  
}  
p {  
    background-image: url(http://www.inet.vn/smallbg.jpg);  
    background-position: 20px 10px;  
    background-repeat: repeat / repeat-x / repeat-y / no-repeat;  
    background-attachment: fixed / scroll;  
}
```





CSS » Property » Font

- ❖ color: định nghĩa màu font chữ
- ❖ font-family: tên font chữ
- ❖ font-size: kích cỡ của font theo: %, px, tên kích cỡ
- ❖ font-style: italic, normal, oblique
- ❖ font weight: mức độ đậm theo thang 100 hoặc tên
- ❖ font-variant: normal, small-caps, inherit

```
p.smallfont {  
    color: red / #1078E1 / rgb( r, g, b );  
    font-family: serif;  
    font-size: 150% / 20px / large;  
    font-style: italic / normal / oblique;  
    font-weight: 200 / bold;  
    font-variant: normal / small-caps / inherit;  
}
```





CSS » Property » Text

- ❖ text-decoration: thêm đường ngang (trên, dưới, giữa)
- ❖ text-indent: khoảng trống từ mép trái đến ký tự đầu
- ❖ text-align: canh dòng text
- ❖ text-transform: chuyển đổi chữ hoa, chữ thường
- ❖ white-space: cho phép text được wrap hay không
- ❖ word-spacing: khoảng cách giữa các từ tính bằng pixel
- ❖ letter-spacing: khoảng cách giữa các ký tự tính bằng pixel

```
p.textformat{  
    text-decoration: line-through / overline / underline / none;  
    text-indent: 20px / 30%;  
    text-align: right / left / center / justify;  
    text-transform: capitalize / uppercase / lowercase;  
    white-space: nowrap;  
    word-spacing: 10px;  
    letter-spacing: 3px;  
}
```





CSS » Property » Box (padding)

❖ padding: là khoảng cách từ border đến content

✓ Padding 1 giá trị:

▪ Padding: <k/c trên dưới trái phải> px;

✓ Padding 2 giá trị:

▪ padding: <k/c trên dưới>px <k/c trái phải>px;

✓ Padding 3 giá trị:

▪ padding : <trên>px <trái và phải>px <dưới>px;

✓ Padding 4 giá trị:

▪ padding: <trên>px <phải>px <dưới>px <trái>px;

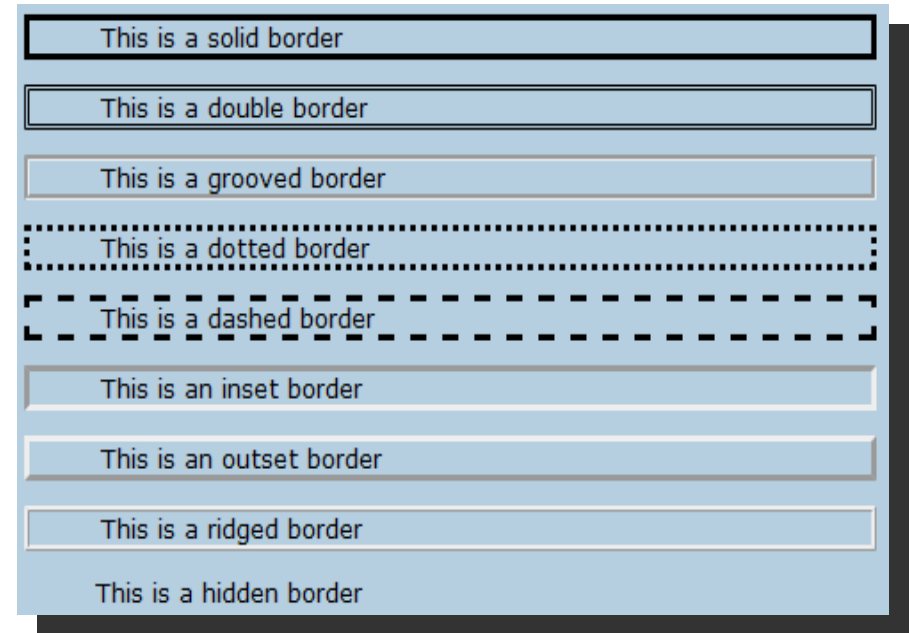




CSS » Property » Box (border)

❖ Border là đường viền xung quanh đối tượng, giữa padding và margin.

- ✓ border-style: kiểu border
- ✓ border-width: độ dày border
- ✓ border-color: màu border
- ✓ *Có thể ghép cùng 1 dòng*



```
p.customborder{  
  border-style: solid, double, groove, dotted, dashed, inset, outset, ridge, hidden;  
  border-width: medium, thick, 7px;  
  border-color: blue, #123456, rgb( 100, 100, 255);  
}  
p.allinoneborder{ border: 20px outset blue ;}
```





CSS » Property » Box (margin)

❖ Khoảng trống bên ngoài Border; cú pháp giống padding

✓ Margin 1 giá trị:

▪ margin: <k/c trên dưới trái phải> px;

✓ Margin 2 giá trị:

▪ margin: <k/c trên dưới>px <k/c trái phải>px;

✓ Margin 3 giá trị:

▪ margin: <trên>px <trái và phải>px <dưới>px;

✓ Margin 4 giá trị:

▪ margin: <trên>px <phải>px <dưới>px <trái>px;





❖ list-style-type

- ✓ Danh sách không thứ tự: square, circle, disc (default), none
- ✓ Danh sách có thứ tự: upper-alpha, lower-alpha, upper-roman, lower-roman, decimal (default), none

❖ list-style-image: dùng hình làm list

❖ list-style-position: vị trí đặt biểu tượng

```
ol { list-style-type: upper-alpha, lower-alpha, upper-roman, lower-roman, decimal (default), none; }  
ul { list-style-type: square, circle, disc (default), none;  
ul { list-style-image: url("listArrow.gif"); }  
ul { list-style-position: inside, outside; }  
ul { list-style: upper-roman inside url("http://www.niit.vn/bullet.gif"); }
```



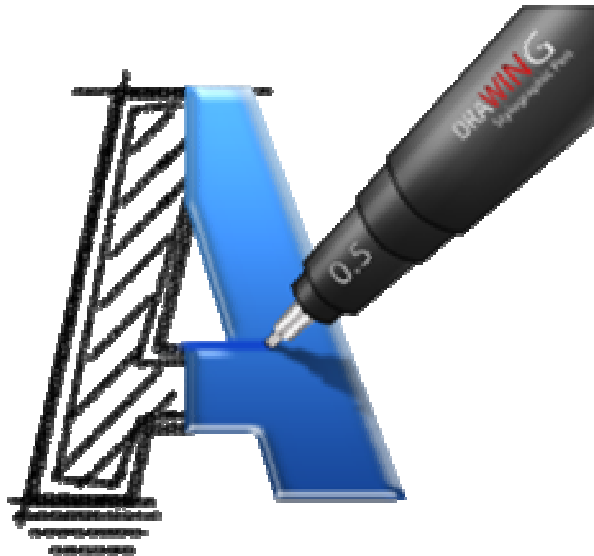


CSS » anchor/link

- ❖ Trạng thái của anchor/link:
 - ✓ link – Trạng thái chưa được click, không có chuột đang trở vào
 - ✓ visited – Trạng thái đã được click trước đó, không có chuột đang trở vào
 - ✓ hover – Trạng thái đang có chuột trở vào
 - ✓ active – trạng thái khi vừa click xong
- ❖ Anchor/link không phải là 1 Selector; trong CSS được coi là 1 lớp giả (pseudo-classes)
 - ✓ a:(tên trạng thái) { attribute: value; }

```
a:link { color: red; text-decoration: none; }  
a:visited { color: red; text-decoration: none; }  
a:hover { color: blue; }
```





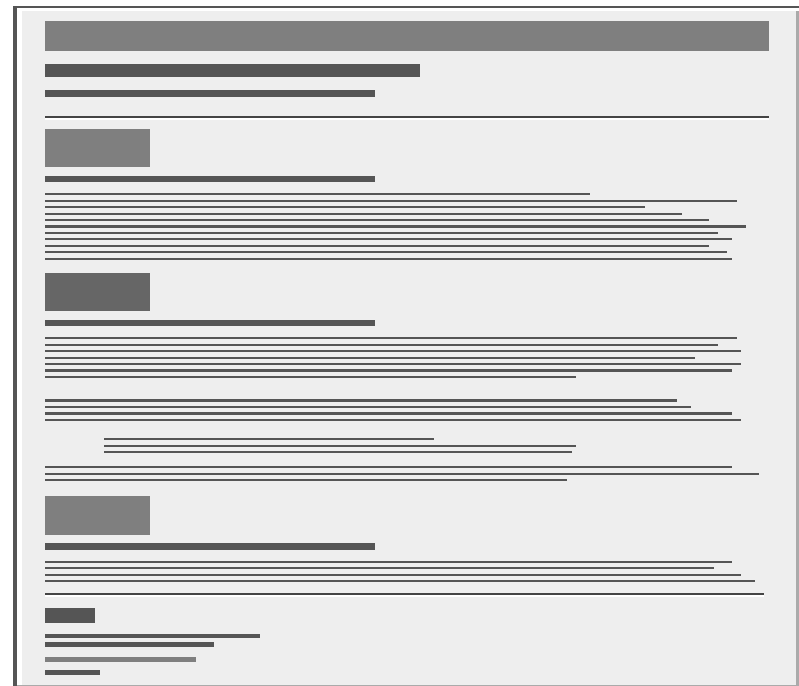
4. Một số lời khuyên thiết kế website





Design Guide (1)

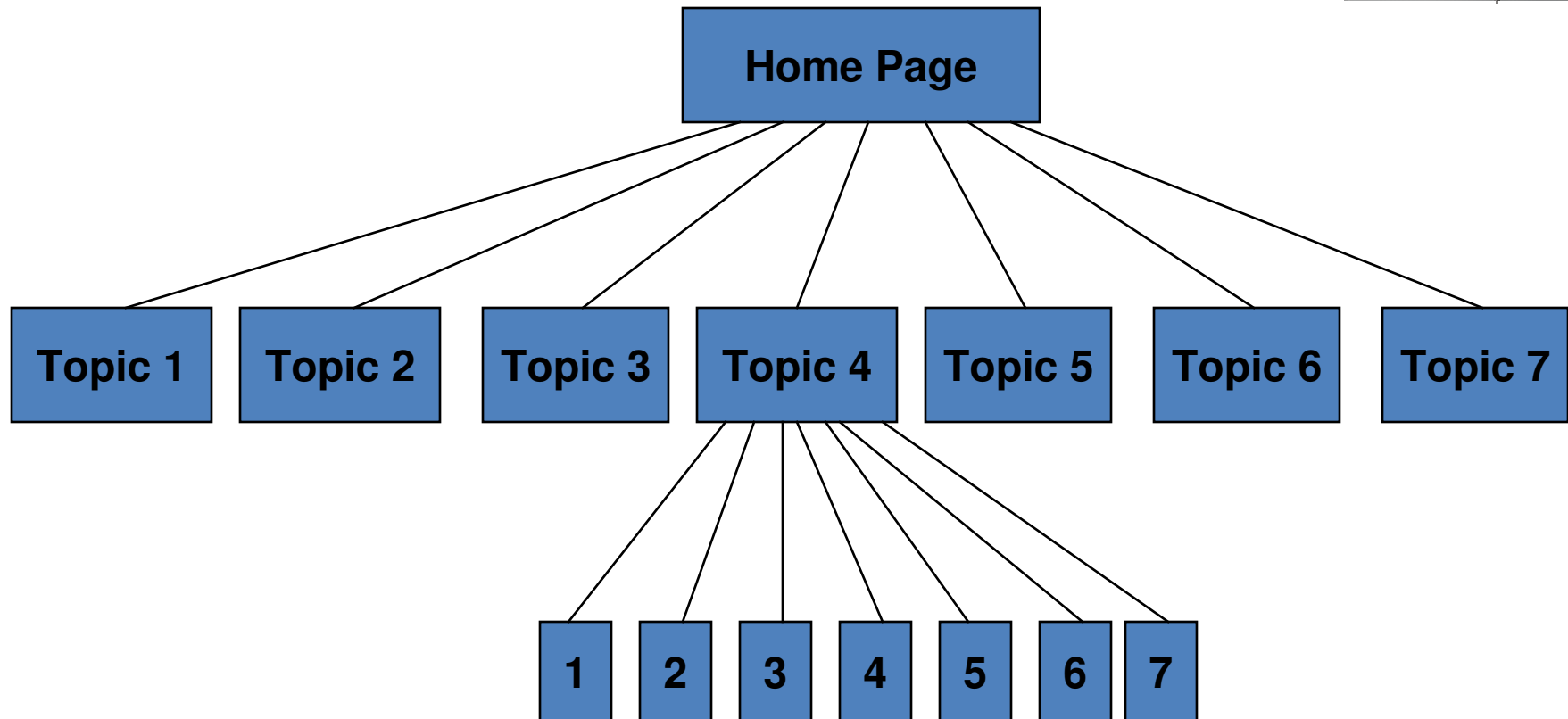
- ❖ Cố gắng giữ website gọn; không quá nhiều level; người dùng click 3 lần là có thể đến được trang web cần tìm.
- ❖ Bố trí paragraph/hình ảnh sao cho mắt di chuyển tối thiểu





Design Guide (2)

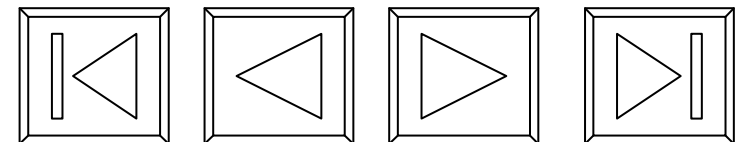
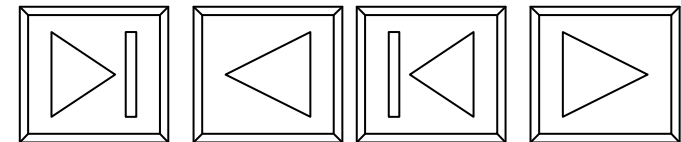
- ❖ Số lượng topic không nên quá lớn
- ❖ Cần có Sitemap, và chức năng tìm kiếm





Design Guide (3)

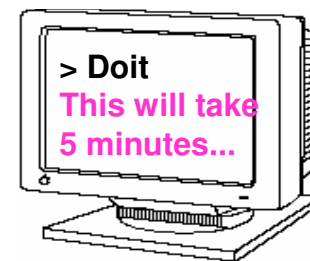
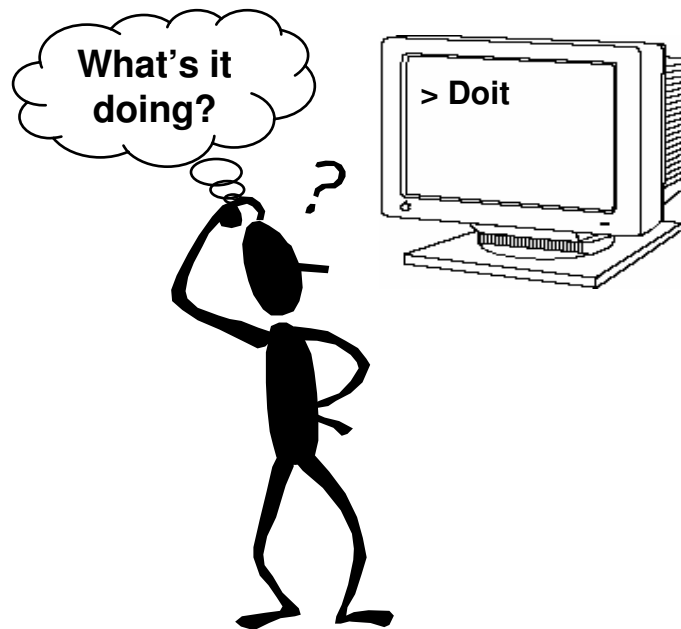
- ❖ Đặt tên hoặc sử dụng hình ảnh nên thông dụng, rõ ràng; vị trí phù hợp với thực tế (ví dụ phần help, about thường nằm bên phải, Home nằm bên trái cùng, trong 2 hình dưới hình nào dễ hiểu hơn)
- ❖ Các mục thông dụng của website nên dùng các quy tắc đặt tên:
 - ✓ Home / Main/ Trang chủ
 - ✓ Search / Find/ Tìm kiếm
 - ✓ Contact Us/ Liên hệ
 - ✓ About Us/ Về ...
 - ✓ Help / FAQ/ Câu hỏi thường gặp
 - ✓ News / What's New/ Tin tức





Design Guide (4)

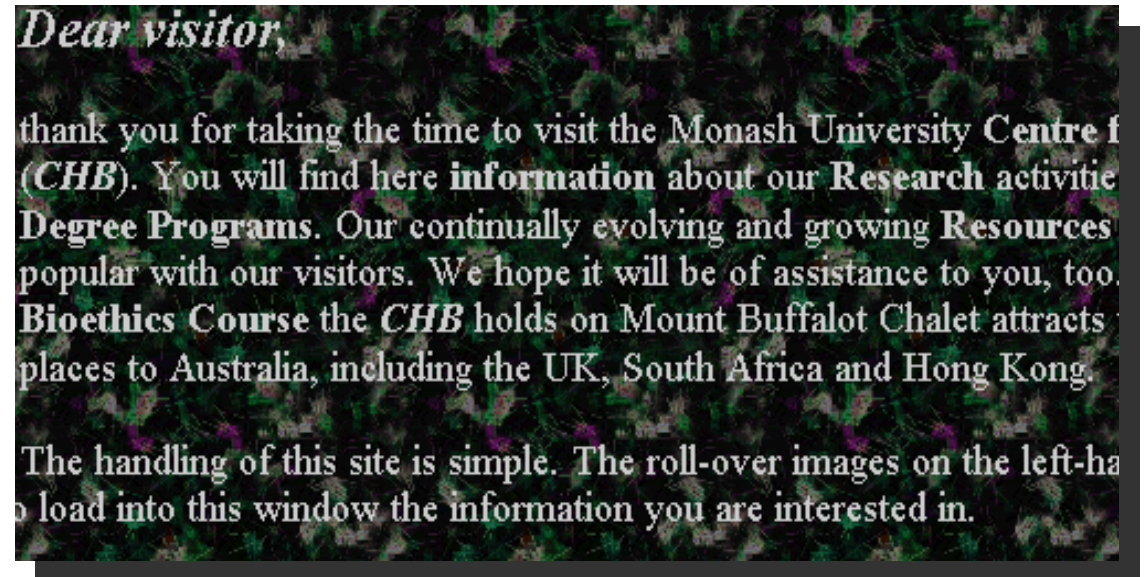
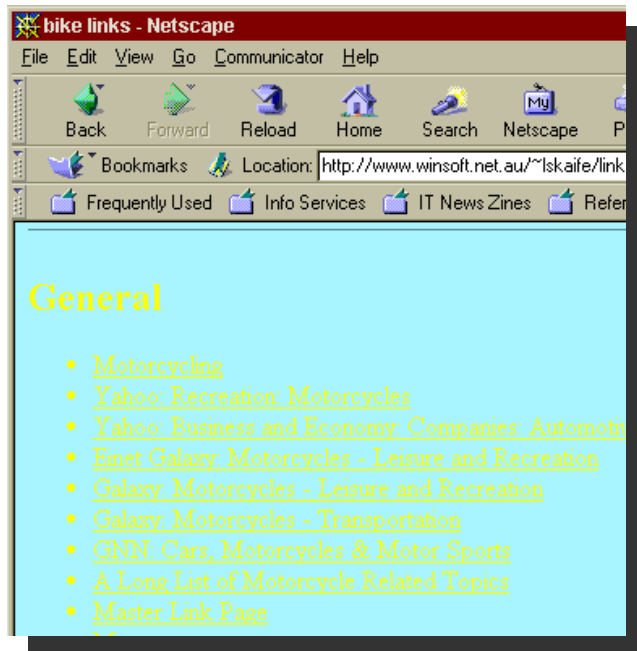
- ❖ Đề ý tính tương thích của trình duyệt;
- ❖ Thống nhất ngôn ngữ, hiệu ứng, hình ảnh, cách thức nhập dữ liệu trong toàn bộ website.
- ❖ Cung cấp thông tin tương tác cho user





Design Guide (5)

- ❖ Thông tin quan trọng cần được thấy ngay từ cái nhìn đầu tiên (không nhất thiết phải trình bày ở trang chủ).
- ❖ Sử dụng Landing page để hướng người dùng đến thông tin quan trọng.
- ❖ Nên canh lề trái text, phối hợp màu hợp lý.





Design Guide (6)

- ❖ Multimedia có thể cung cấp thông tin nhanh chóng, nâng cao chất lượng nội dung, thích ứng với người dùng khiếm thị;
- ❖ Nhưng Multimedia cần nhiều thời gian để tải về, có thể không tương thích với trình duyệt, đôi lúc đòi hỏi phải có plug-in.
- ❖ Windows title dễ hiểu cho người dùng và dễ hiểu cho Search Engine.
- ❖ Cuối cùng: **đừng đổi định dạng của link**





FAQ

