

UBND TỈNH BÀ RỊA – VŨNG TÀU  
**TRƯỜNG CAO ĐẲNG KỸ THUẬT CÔNG NGHỆ**



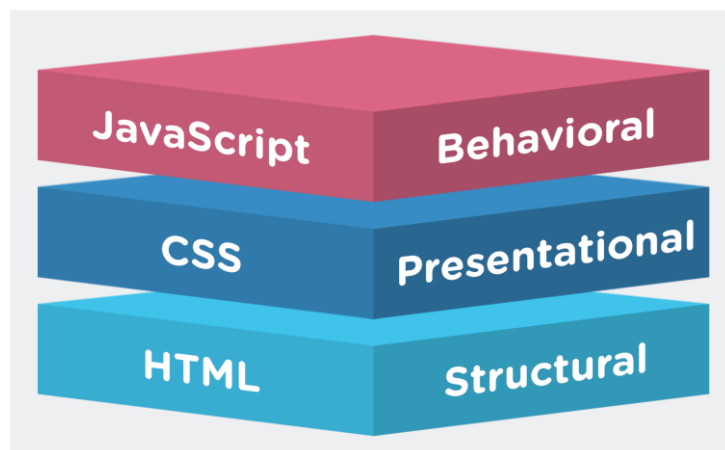
## **GIÁO TRÌNH**

**MÔ ĐUN THIẾT KẾ WEB**

**NGHỀ: QUẢN TRỊ MẠNG**

**TRÌNH ĐỘ: CAO ĐẲNG**

*(Ban hành kèm theo Quyết định số: ...../QĐ-CDKTCN, ngày ... tháng ... năm 20..... của Hiệu trưởng Trường Cao đẳng Kỹ thuật Công nghệ BR-VT)*



**BÀ RỊA – VŨNG TÀU, NĂM 2020**



## **TUYÊN BỐ BẢN QUYỀN**

Nhằm đáp ứng nhu cầu học tập và nghiên cứu cho giảng viên và sinh viên ngành Công nghệ Thông tin trong trường Cao đẳng Kỹ thuật Công nghệ Bà Rịa – Vũng Tàu, chúng tôi đã thực hiện biên soạn tài liệu Thiết kế web này.

Tài liệu được biên soạn thuộc loại giáo trình phục vụ giảng dạy và học tập, lưu hành nội bộ trong Nhà trường nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.



## LỜI GIỚI THIỆU

Giáo trình “Thiết kế web” được biên soạn dựa trên khung chương trình đào tạo Cao đẳng nghề Công nghệ Thông tin đã được Trường Cao đẳng Kỹ thuật Công nghệ Bà Rịa – Vũng Tàu phê duyệt.

Tác giả đã nghiên cứu một số tài liệu, công nghệ hiện đại kết hợp với kinh nghiệm làm việc thực tế để viết nên giáo trình này. Nội dung được tác giả trình bày cô đọng, dễ hiểu kèm theo các bước hướng dẫn thực hành chi tiết theo nguyên tắc quan tâm đến kết quả đầu ra, khả năng tự học và kỹ năng cần thiết để HSSV có thể hoàn thành thiết kế và lập trình những xử lý tương tác cơ bản trên trang web với sự kết hợp của HTML5 và CSS3, lập trình những xử lý cơ bản trên trang web bằng Javascript, thiết kế được giao diện trang web chuyên nghiệp với jQuery, ... tạo và quản lý nội dung của website đáp ứng nhu cầu thực tế của doanh nghiệp.

Nội dung giáo trình được chia thành 8 bài, trong đó:

Bài 1: Môi trường tạo trang web tĩnh DreamWeaver, thẻ HTML

Bài 2: Tạo trang web bằng HTML5

Bài 3: Hoàn chỉnh giao diện trang web với CSS

Bài 4: Tùy biến giao diện web với CSS3

Bài 5: Xử lý tương tác với Javascript

Bài 6: Một số kỹ thuật thiết kế nâng cao

Bài 7: Tạo giao diện chuyên nghiệp bằng jQuery

Bài 8: Publish website

Trong quá trình biên soạn, chắc chắn giáo trình còn nhiều thiếu sót. Tác giả rất mong nhận được ý kiến đóng góp của quý thầy/cô và các em học sinh, sinh viên để tiếp tục hoàn thiện hơn.

Xin chân thành cảm ơn quý đồng nghiệp, bạn bè đã có những ý kiến đóng góp trong quá trình biên soạn giáo trình này.

Bà Rịa – Vũng Tàu, ngày ..... tháng ..... năm .....

Tham gia biên soạn

1. Nguyễn Phạm Ái Hương – Chủ biên



# MỤC LỤC

LỜI GIỚI THIỆU.....	1
MỤC LỤC.....	3
BÀI 1: MÔI TRƯỜNG TẠO TRANG WEB TÍNH DREAMWAVER, THẺ HTML.....	11
1. Giới thiệu.....	<b>Error! Bookmark not defined.</b>
2. Quản lý các site .....	<b>Error! Bookmark not defined.2</b>
3. Quản lý nội dung trong site.....	<b>Error! Bookmark not defined.3</b>
4. Chọn vùng nhìn .....	<b>Error! Bookmark not defined.4</b>
5. Thẻ HTML cơ bản.....	<b>Error! Bookmark not defined.</b>
CÂU HỎI, BÀI TẬP .....	<b>Error! Bookmark not defined.9</b>
BÀI 2: TẠO TRANG WEB BẰNG HTML5 .....	21
1. Sử dụng các thẻ ngữ nghĩa .....	21
2. Sử dụng các thẻ mới của HTML5 trong Form.....	22
3. Sử dụng HTML5 GraphicsSVG .....	24
4. Sử dụng các HTML Media.....	26
CÂU HỎI, BÀI TẬP .....	<b>Error! Bookmark not defined.7</b>
BÀI 3: HOÀN CHỈNH GIAO DIỆN TRANG WEB VỚI CSS.....	288
1. Giới thiệu CSS và các loại CSS .....	<b>Error! Bookmark not defined.8</b>
2. Tạo và sử dụng CSS .....	<b>Error! Bookmark not defined.9</b>
3. Tạo định dạng chung cho trang web .....	30
4. Tạo giao diện trang web canh giữa trình duyệt.....	30
5. Tạo hiệu ứng đổi định dạng cho menu.....	31
6. Tạo khung viền bo tròn cho nội dung .....	<b>Error! Bookmark not defined.2</b>
CÂU HỎI, BÀI TẬP .....	<b>Error! Bookmark not defined.2</b>
BÀI 4: TÙY BIẾN GIAO DIỆN WEB VỚI CSS3 .....	28
1. Rounded Corners.....	<b>Error! Bookmark not defined.</b>

2. Border Images .....	<b>Error! Bookmark not defined.</b>	4
3. Backgrounds .....	<b>Error! Bookmark not defined.</b>	5
4. Colors .....	<b>Error! Bookmark not defined.</b>	7
5. Gradients .....	<b>Error! Bookmark not defined.</b>	8
6. Shadows.....	<b>Error! Bookmark not defined.</b>	9
7. Text.....		41
8. Fonts .....		42
9. 2D Transforms.....		42
10. 3D Transforms.....		44
11. Transitions .....		46
12. Animations .....		47
13. Box Sizing .....		47
14. User Interface .....		47
CÂU HỎI, BÀI TẬP .....		48
BÀI 5: XỬ LÝ TƯƠNG TÁC VỚI JAVASCRIPTS .....		50
1. Tổng quan Javascript.....		50
2. Sử dụng cấu trúc điều khiển .....		55
3. Sử dụng mảng và hàm .....		57
CÂU HỎI, BÀI TẬP .....		61
BÀI 6: MỘT SỐ KỸ THUẬT THIẾT KẾ NÂNG CAO.....		5162
1. Template trong Dreamwaver.....		62
2. Navigation Menu .....		66
3. Đối tượng nâng cao trong Javascript.....		67
CÂU HỎI, BÀI TẬP .....		73
BÀI 7: TẠO GIAO DIỆN CHUYÊN NGHIỆP BẰNG JQUERY .....		74
1. Tổng quan JQuery .....		74
2. Chọn element.....		75
3. Thay đổi nội dung, thuộc tính .....		77



4. Xử lý sự kiện.....	80
5. Tạo hiệu ứng và hoạt ảnh.....	83
CÂU HỎI, BÀI TẬP.....	85
BÀI 8: PUBLISH WEBSITE.....	86
1. Đăng ký web hosting miễn phí .....	<b>Error! Bookmark not defined.</b> 86
2. Publish web lên web hosting miễn phí.....	86
3. Publish web lên local IIS.....	87
CÂU HỎI, BÀI TẬP.....	89

# GIÁO TRÌNH MÔ ĐUN

**Tên mô đun: Thiết kế web**

**Mã mô đun: MĐ18**

**Vị trí, tính chất, ý nghĩa và vai trò của mô đun:**

- Vị trí: được bố trí sau khi học xong các môn cơ sở và Đồ họa ứng dụng
- Tính chất: Là module chuyên ngành của nghề công nghệ thông tin, cung cấp kiến thức và kỹ năng của một công việc trong quy trình xây dựng ứng dụng web
- Ý nghĩa và vai trò của mô đun: cung cấp cho người học kiến thức và kỹ năng để thiết kế và lập trình những xử lý tương tác cơ bản trên trang web với sự kết hợp của HTML5 và CSS3, lập trình những xử lý cơ bản trên trang web bằng Javascript, thiết kế được giao diện trang web chuyên nghiệp với jQuery, ... tạo và quản lý nội dung của website đáp ứng nhu cầu thực tế của doanh nghiệp.

**Mục tiêu của mô đun:**

Về kiến thức:

- Biết tạo và quản lý nội dung của website trong phần mềm DreamWeaver.
- Biết công dụng các thẻ HTML, HTML5, định dạng của CSS và CSS3
- Có kiến thức cơ bản về JavaScript.
- Biết công dụng của JQuery
- Biết quy trình publish website lên internet.

Về kỹ năng:

- Chia bố cục, thiết kế được các trang web bằng các thẻ HTML, HTML5, và định dạng của CSS và CSS3.
- Lập trình những xử lý tương tác cơ bản trên trang web bằng Javascript.
- Sử dụng được Template, Navigation và menu trong Dreamweaver.
- Thiết kế được giao diện trang web chuyên nghiệp với jQuery.
- Đăng ký được web hosting miễn phí
- Publish được website lên Web Werver

Về năng lực tự chủ và trách nhiệm:

- Có tinh thần trách nhiệm, ý thức tổ chức kỷ luật, tác phong công nghiệp, tinh thần hợp tác trong công việc

- Có tính chủ động, độc lập trong công việc, tự học cập nhật kiến thức, nâng cao trình độ chuyên môn.
- Có khả năng tổ chức và điều hành một nhóm, đánh giá được các thành viên trong nhóm.
- Rèn luyện tính cẩn thận, kiên trì, sáng tạo, độc lập và hoạt động nhóm.
- Bảo đảm an toàn và vệ sinh cho người và thiết bị trong phòng máy.

**Nội dung của mô đun:**

Số TT	Tên các bài trong mô đun	Thời gian (giờ)			
		Tổng số	Lý thuyết	Thực hành, thí nghiệm, thảo luận, bài tập	Kiểm tra
1	<b>Bài 1: Môi trường tạo trang web tĩnh DreamWeaver, thẻ HTML</b>	<b>4</b>	<b>1</b>	<b>3</b>	<b>0</b>
	1. Giới thiệu	0.4	0.1	0.3	0
	2. Quản lý các site	0.7	0.2	0.5	0
	3. Quản lý nội dung trong site	0.6	0.1	0.5	0
	4. Chọn vùng nhìn	0.3	0.1	0.2	0
	5. Thẻ HTML cơ bản	2	0.5	1.5	0
2	<b>Bài 2: Tạo trang web bằng HTML5</b>	<b>15</b>	<b>5</b>	<b>9</b>	<b>1</b>
	1. Giới thiệu HTML5	0.7	0.2	0.5	0
	2. Sử dụng các thẻ ngữ nghĩa	1.3	0.3	1	0
	3. Sử dụng các thẻ mới của HTML5 trong Form	3.5	1.0	2.5	0
	4. Sử dụng HTML Graphics SVG	4.5	2.0	2.5	0
	5. Sử dụng các HTML Media	4	1.5	2.5	0
	Kiểm tra	1	0	0	1
3	<b>Bài 3: Hoàn chỉnh giao diện trang web với CSS</b>	<b>13</b>	<b>4</b>	<b>8</b>	<b>1</b>
	1. Giới thiệu CSS và các loại CSS	0.6	0.1	0.5	0
	2. Tạo và sử dụng CSS	0.8	0.3	0.5	0
	3. Tạo định dạng chung cho trang web	1.6	0.6	1	0

	4. Tạo giao diện trang web canh giữa trình duyệt	1.5	0.5	1	0
	5. Tạo hiệu ứng đổi định dạng cho menu	4.5	1.5	3	0
	6. Tạo khung viền bo tròn cho nội dung	3	1.0	2	0
	Kiểm tra	1	0	0	1
<b>4</b>	<b>Bài 4: Tùy biến giao diện web với CSS3</b>	<b>14</b>	<b>4</b>	<b>8</b>	<b>2</b>
	1. Rounded Corners	0.6	0.2	0.4	0
	2. Border Images	0.6	0.2	0.4	0
	3. Backgrounds	0.6	0.2	0.4	0
	4. Colors	0.6	0.2	0.4	0
	5. Gradients	0.9	0.3	0.6	0
	6. Shadows	0.9	0.3	0.6	0
	7. Text	0.6	0.2	0.4	0
	8. Fonts	0.6	0.2	0.4	0
	9. 2D Transforms	0.9	0.3	0.6	0
	10. 3D Transforms	0.9	0.3	0.6	0
	11. Transitions	0.9	0.3	0.6	0
	12. Animations	0.9	0.3	0.6	0
	13. Box Sizing	1.5	0.5	1	0
	14. User Interface	1.5	0.5	1	0
	Kiểm tra	2	0	0	2
<b>5</b>	<b>Bài 5: Xử lý tương tác với Javascript</b>	<b>12</b>	<b>2</b>	<b>8</b>	<b>2</b>
	1. Tổng quan Javascript	0.8	0.2	0.6	0
	2. Sử dụng cấu trúc điều khiển	6.4	1	5.4	0
	3. Sử dụng mảng và hàm	2.8	0.8	2	0
	Kiểm tra	2	0	0	2
<b>6</b>	<b>Bài 6: Một số kỹ thuật thiết kế nâng cao</b>	<b>10</b>	<b>2</b>	<b>6</b>	<b>2</b>
	1. Template trong Dreamweaver	2	0.5	1.5	0
	2. Navigation và Menu	2	0.5	1.5	0
	3. Đối tượng nâng cao trong Javascript	4	1.0	3	0
	Kiểm tra	2	0	0	2
<b>7</b>	<b>Bài 7: Tạo giao diện chuyên nghiệp bằng</b>	<b>18</b>	<b>3</b>	<b>13</b>	<b>2</b>

	<b>jQuery</b>				
	1. Tổng quan jQuery	1.2	0.2	1	0
	2. Chọn element	1.2	0.2	1	0
	3. Lọc chọn element	1.2	0.2	1	0
	4. Thay đổi nội dung, thuộc tính	2.4	0.4	2	0
	5. Xử lý sự kiện	5	1	4	0
	6. Tạo hiệu ứng và hoạt ảnh	5	1	4	0
	Kiểm tra	2	0	0	2
<b>8</b>	<b>Bài 8: Publish website</b>	<b>4</b>	<b>1</b>	<b>3</b>	<b>0</b>
	1. Đăng ký web hosting miễn phí	1.5	0.5	1	0
	2. Publish web lên web hosting miễn phí	1.5	0.3	1	0
	3. Publish web lên local IIS	1.5	0.2	1	0
	<b>Cộng</b>	<b>90</b>	<b>22</b>	<b>58</b>	<b>10</b>



# BÀI 1: MÔI TRƯỜNG TẠO TRANG WEB TĨNH DREAMWEAVER

Mã bài: 18.1

## Giới thiệu:

Adobe DreamWeaver là ứng dụng thiết kế web được sử dụng phổ biến nhất hiện nay. Adobe DreamWeaver hỗ trợ các chức năng mạnh mẽ cho cả người thiết kế và lập trình web.

## Mục tiêu:

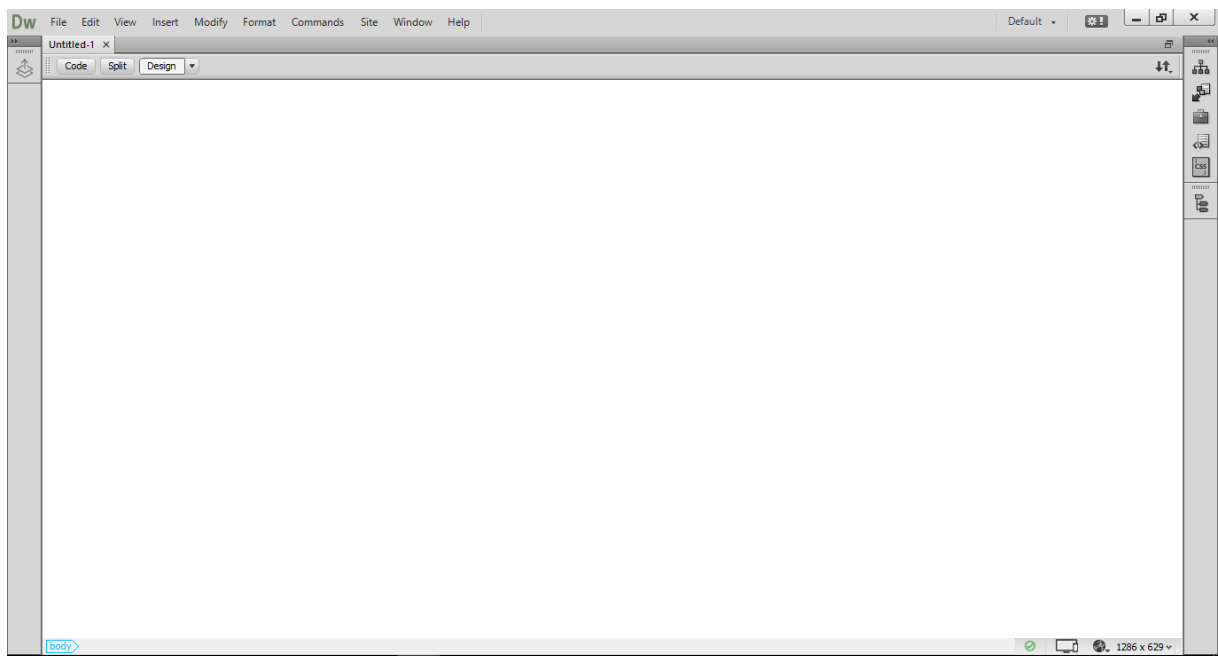
- Biết công dụng của từng thành phần trong DreamWeaver
- Biết tạo và quản lý site
- Biết quản lý nội dung trong site
- Quản lý cẩn thận nội dung trong site, tránh xóa nhầm nội dung, site

## Nội dung chính:

### 1. Giới thiệu

#### 1.1. Khởi động

- Click đôi chuột lên biểu tượng Adobe DreamWeaver trên desktop
- Vào Start → Programs → Adobe DreamWeaver




Hình 1.1. Giao diện Adobe DreamWeaver

## 1.2. Quản lý các palette

- Vào menu Window → Chọn/bỏ chọn để hiện/ẩn palette.

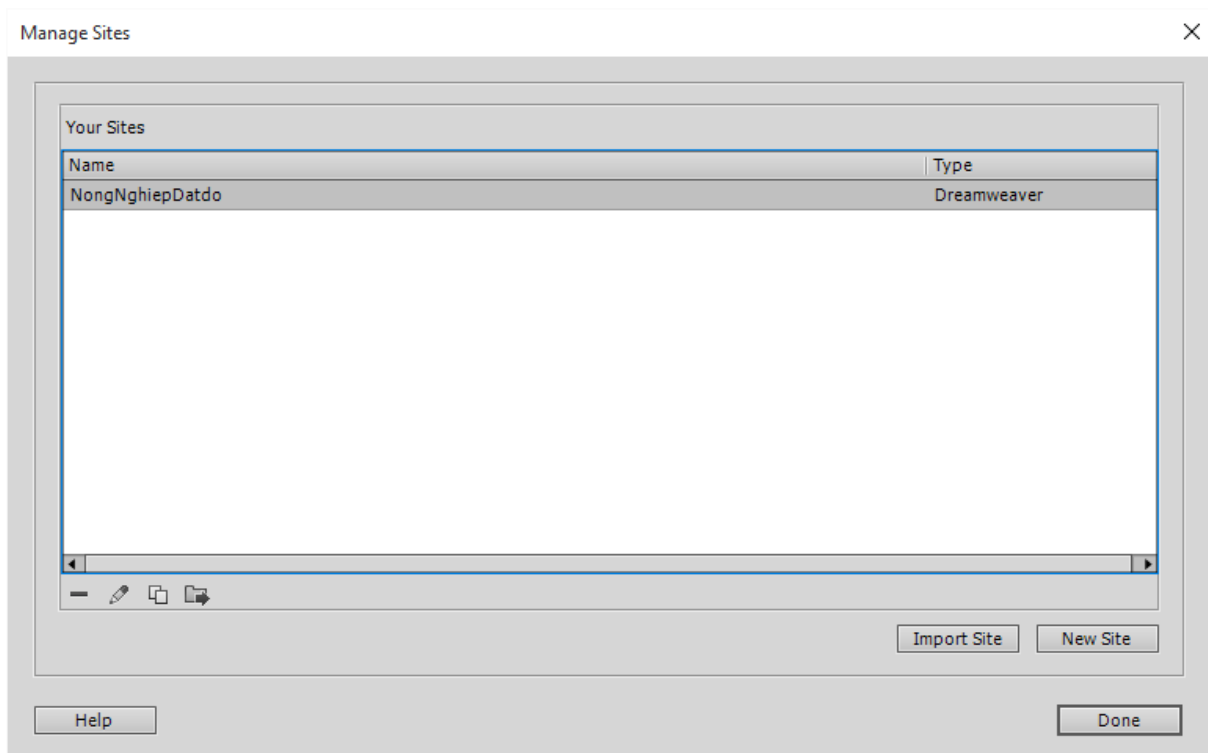


Hình 1.2. Vị trí hiển thị của các palette

- Muốn mở rộng palette nào thì click chuột lên biểu tượng của palette đó
- Click chuột lên biểu tượng  để mở rộng tất cả palette

## 2. Quản lý các site

Vào menu Site → Manage Sites. Xuất hiện hộp thoại quản lý các site.



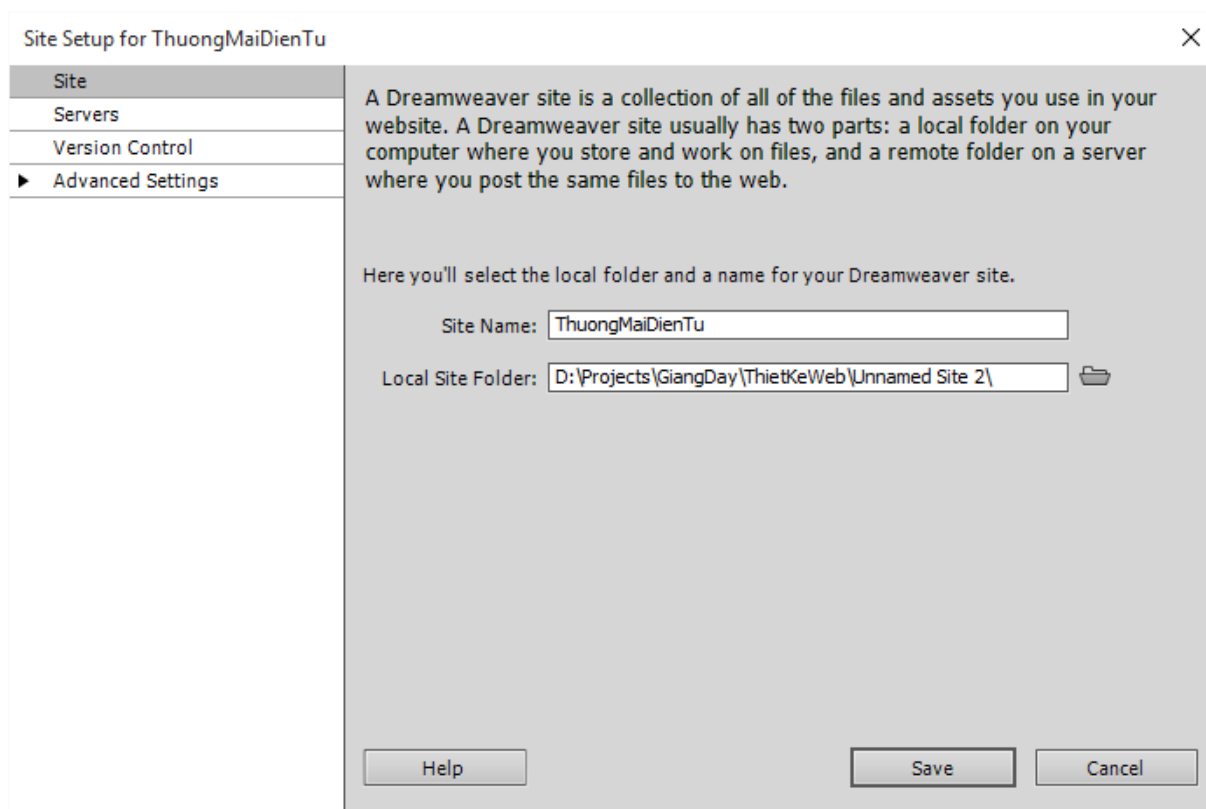
Hình 1.3. Hộp thoại Manage Sites

### 2.1. Tạo site

- Click chuột lên nút New Site



- Đặt tên cho site, chọn ổ đĩa, thư mục chứa site rồi chọn Save trong hộp thoại Site Setup



Hình 1.4. Hộp thoại Site Setup

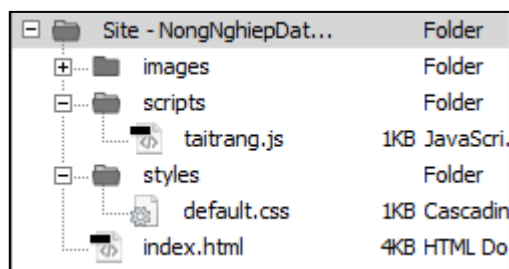
## 2.2. Xóa site

- Trong hộp thoại Manage Site, chọn site cần xóa trong danh sách
- Click chuột lên biểu tượng Delete the current selected sites (—)

## 3. Quản lý nội dung trong site

### 3.1. Cấu trúc site

Mở rộng palette Files. Cấu trúc site cần tạo sẽ có nội dung như sau:



Hình 1.5. Cấu trúc site

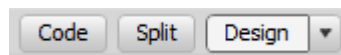
### 3.2. Các loại tập tin trong site

- .htm, .html: trang web, tài liệu html
- .jpg, .png, .gif: tập tin hình ảnh, dùng để chèn, định dạng nền cho trang web, thường được đặt trong thư mục images
- .css: tập tin định dạng cho trang web, thường được đặt trong thư mục styles
- .js: tập tin javascript, chứa các đoạn mã thực thi trên trình duyệt, thường được đặt trong thư mục scripts

### 3.3. Quản lý thư mục, tập tin

- Yêu cầu: tạo 03 thư mục images, scripts, styles và tập tin index.html trong site
- Các bước thực hiện
  - Click chuột phải lên đối tượng chứa trong palette Files, chọn New Folder
  - Đặt tên cho folder cần tạo
  - Click chuột phải lên site, chọn New File, đặt tên cho tập tin cần tạo

## 4. Chọn vùng nhìn làm việc



*Hình 1.6. Các tùy chọn vùng nhìn làm việc*

- Click đôi chuột lên trang web cần thiết kế
- Click chuột lên biểu tượng vùng nhìn cần chọn
  - Code: thiết kế trang bằng các thẻ HTML
  - Design: thiết kế trang bằng cách kéo thả các element vào trang web
  - Split: tách cửa sổ tài liệu thành 2 vùng nhìn Code và Design

## 5. Thẻ HTML cơ bản

### 5.1. Tìm hiểu cấu trúc tài liệu HTML

- Hầu hết các element bắt đầu bằng thẻ mở và kết thúc bằng thẻ đóng. Ví dụ:  
<body> ... </body>

- Nội dung được đặt giữa thẻ mở và thẻ đóng. Ví dụ: <p>Đây là đoạn văn bản</p>
- Một số thẻ đặc biệt vừa mở và đóng
  - o <br />: ngắt xuống dòng
  - o <hr />: tạo đường kẻ nằm ngang
  - o <img ... />: chèn ảnh (sẽ được tìm hiểu chi tiết ở phần sau)
- Thuộc tính của element được đặt trong thẻ mở.
  - o Ví dụ: 
  - o src: là thuộc tính
  - o images/i1.jpg: giá trị của thuộc tính src.
  - o Giá trị của thuộc tính luôn được đặt trong cặp dấu nháy kép ""
- Element nào được mở trước thì phải đóng sau

## 5.2. Thiết kế trang web bằng các thẻ HTML cơ bản

### 5.2.1. META, LINK, STYLE, SCRIPT

#### - META

- o Tạo từ khóa tìm kiếm

```
<meta name="keywords" content="thiết kế web, web design" />
```

- o Tạo nội dung mô tả cho trang

```
<meta name="description" content="Website hướng dẫn thiết kế web" />
```

### 5.2.2. TABLE, TR, TD, TH

- TABLE: Tạo bảng
- TR: Tạo dòng trong bản
- TD: Tạo ô trong dòng
- TH: Tương tự TD nhưng là ô tiêu đề, nội dung được tự động tô đậm và canh giữa ô

Ví dụ:

```

<table>
  <tr>
    <th>STT</th>
    <th>Họ tên</th>
    <th>Điểm</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Nguyễn Văn A</td>
    <td>6</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Trần Văn B</td>
    <td>7</td>
  </tr>
</table>

```

STT	Họ tên	Điểm
1	Nguyễn Văn A	6
2	Trần Văn B	7

### 5.2.3. DIV, P, BR, HR

- DIV: Chia vùng riêng, chiếm hết chiều ngang của vùng chứa

Ví dụ:

```

<div>header</div>
<div>Nội dung</div>
<div>Footer</div>

```

- P: Tạo đoạn văn bản

Ví dụ:

```

<div>header</div>
<div>Nội dung</div>
<div>
    <p>Bản quyền thuộc về BRTVC</p>
    <p>Địa chỉ: Đất Đỏ, BR-VT</p>
</div>

```

- BR: Ngắt xuống dòng trong đoạn văn bản

```

<p>
Tại hội nghị lần 2 để đánh giá những kết quả năm 2015 và bàn một số nhiệm vụ, phương hướng lớn trong năm 2016, Bí thư Thành ủy Đà Nẵng Nguyễn Xuân Anh đã đề cập nhiều đến việc TP sẽ tiến hành phân cấp, phân quyền mạnh mẽ trong thời gian đến.<br />Theo ông Xuân Anh, Thường trực Thành ủy và Ban Thường vụ Thành ủy Đà Nẵng đã có chủ trương thời gian tới sẽ tiến hành phân cấp mạnh. Cơ chế tập quyền, tập trung quyền lực đã lỗi thời, cần phải tiến hành phân cấp mạnh. Phân cấp đi đôi với trách nhiệm.
</p>

```

```

Tại hội nghị lần 2 để đánh giá những kết quả năm 2015 và bàn một số nhiệm vụ, phương hướng lớn trong năm 2016, Bí thư Thành ủy Đà Nẵng Nguyễn Xuân Anh đã đề cập nhiều đến việc TP sẽ tiến hành phân cấp, phân quyền mạnh mẽ trong thời gian đến.
Theo ông Xuân Anh, Thường trực Thành ủy và Ban Thường vụ Thành ủy Đà Nẵng đã có chủ trương thời gian tới sẽ tiến hành phân cấp mạnh. Cơ chế tập quyền, tập trung quyền lực đã lỗi thời, cần phải tiến hành phân cấp mạnh. Phân cấp đi đôi với trách nhiệm.

```

- HR: Tạo đường kẻ ngang

Ví dụ: <hr />

#### 5.2.4. IMG, A

- IMG: Chèn ảnh vào trang web

Ví dụ:

```

```

- A: Tạo liên kết đến trang web

Ví dụ:

```
<a href="http://brtvc.edu.vn">BRTVC</a>
<a href="lienhe.html">Liên hệ</a>
```

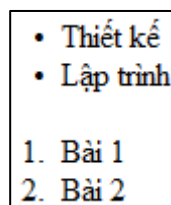
#### 5.2.5. OL, UL, LI

- OL: Tạo danh sách đánh số thứ tự
- UL: Tạo danh sách không đánh số thứ tự

- LI: Tạo mục trong danh sách

Ví dụ:

```
<ul>
  <li>Thiết kế</li>
  <li>Lập trình</li>
</ul>
<ol>
  <li>Bài 1</li>
  <li>Bài 2</li>
</ol>
```



### 5.2.6. FORM, INPUT, SELECT, OPTION

- FORM: Tạo form cho người truy cập nhập thông tin vào trang web
- INPUT: Tạo các điều khiển cho phép người dùng nhập/chọn dữ liệu, đặt trong FORM
- SELECT: Tạo danh sách chọn
- OPTION: Tạo mục trong danh sách chọn

Ví dụ:

```
<form id="form1" name="form1" method="post">
<label for="textfield">Họ tên:</label>
  <input type="text" name="textfield" id="textfield">
  <input type="radio" name="radio" id="radionam" value="radio">
<label for="radionam">Nam</label>
<input type="radio" name="radio" id="radionu" value="radio">
<label for="radionu">Nữ</label>

<label for="select">Nơi sinh:</label>
<select name="select" id="select">
  <option value="brt">Ba Ria - Vung Tau</option>
  <option value="hcm">Tp. Ho Chi Minh</option>
</select>
</form>
```

# CÂU HỎI, BÀI TẬP

## 1.1. Thiết kế một trang web cơ bản theo mẫu: tạo layout cho TrangChu.html



## 2.2. Tạo form và các điều khiển trong form

### PHIẾU ĐĂNG KÝ

Họ và tên:  (\*)

Địa chỉ:

Email:  (\*)

Giới tính:  Nam  Nữ

Tuổi:  (\*)

Sở thích:  Thể thao  
 Văn học  
 Công nghệ thông tin  
 Lịch sử  
 Game

Chuyên ngành:  ▼

Gửi hình:

Yêu cầu:

### YÊU CẦU KIỂM TRA, ĐÁNH GIÁ KẾT QUẢ HỌC TẬP

Đúng cấu trúc tài liệu HTML

Đúng quy tắc: mở trước, đóng sau

Giao diện đăng ký phải được đặt trong form



## BÀI 2: TẠO TRANG WEB BẰNG HTML5

Mã bài: 18.2

### Giới thiệu:

HTML5 là phiên bản mới nhất của HTML, hỗ trợ các element ngữ nghĩa rõ ràng. Với HTML5, các trang web được thiết kế với số lượng element ít hơn, rõ ràng hơn, chèn multimedia đơn giản hơn.

### Mục tiêu:

- Biết công dụng các tag mới trong HTML5
- Bố cục, thiết kế được trang web bằng HTML5
- Chăm thận, an toàn

### Nội dung chính:

#### 1. Sử dụng các thẻ ngữ nghĩa

- section: định nghĩa một vùng của tài liệu

Ví dụ:

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

- article: định nghĩa một bài viết (bài trên forum, blog, báo)

Ví dụ:

```
<article>
  <h1>What Does WWF Do?</h1>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

- nav, menu: định nghĩa vùng chứa navigation (thường gọi là menu) của trang

Ví dụ:

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

- header: định nghĩa vùng header (banner) của trang/bài viết

Ví dụ:

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

- footer: định nghĩa vùng cuối trang (thông tin liên hệ, bản quyền)

Ví dụ:

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">
  someone@example.com</a>.</p>
</footer>
```

## 2. Sử dụng các thẻ mới của HTML5 trong FORM

- datalist, keygen, output
- Các type mới của input trong FORM

- color: chọn màu

```
<form>
  Select your favorite color:
  <input type="color" name="favcolor">
</form>
```

Select your favorite color:

- date: chọn ngày

```

<form>
  Enter a date before 1980-01-01:
  <input type="date" name="bday" max="1979-12-31"><br>
  Enter a date after 2000-01-01:
  <input type="date" name="bday" min="2000-01-02"><br>
</form>

```

Enter a date before 1980-01-01:	<input type="text" value="mm/dd/yyyy"/>
Enter a date after 2000-01-01:	<input type="text" value="mm/dd/yyyy"/>

- o time: chọn giờ

```

<form>
  Select a time:
  <input type="time" name="usr_time">
</form>

```

Select a time:	<input type="text" value="--:-- --"/>
----------------	---------------------------------------

- o email: nhập email

```

<form>
  E-mail:
  <input type="email" name="email">
</form>

```

- o month: chọn tháng, năm

```

<form>
  Birthday (month and year):
  <input type="month" name="bdaymonth">
</form>

```

Birthday (month and year):	<input type="text" value="----- --"/>
----------------------------	---------------------------------------

- o week: chọn tuần trong năm

```

<form>
  Select a week:
  <input type="week" name="week_year">
</form>

```

Select a week:	<input type="text" value="Week --, ----"/>
----------------	--

- o number: chọn số

```
<form>
  Quantity:
  <input type="number" name="points" min="0" max="100" step="10" value="30">
</form>
```

- range: chọn giá trị trong vùng giới hạn

```
<form>
  <input type="range" name="points" min="0" max="10">
</form>
```



### 3. Sử dụng HTML Graphics SVG

#### 3.1. SVG là gì?

SVG là viết tắt của *Scalable Vector Graphics*, là một định dạng hình ảnh (tương tự như JPG, PNG,... mà chúng ta vẫn thường dùng).

SVG được sử dụng để vẽ đồ họa 2D và các ứng dụng đồ họa trên website, phần lớn hữu ích cho các sơ đồ kiểu vecto như các biểu đồ Pie, các đồ thị hai chiều trong hệ tọa độ X, Y

SVG là chuẩn chính thức của tổ chức web thế giới W3C.

#### Ưu điểm

- Hình ảnh SVG có thể phóng to nhưng không vỡ ảnh.
- Hình ảnh SVG có thể được in với chất lượng cao ở bất kỳ độ phân giải nào.
- Hình ảnh SVG có thể được tạo và chỉnh sửa bằng javascript.
- Hình ảnh SVG có thể tìm kiếm, đánh chỉ mục, nén,...
- Hình ảnh SVG có thể chuyển động sử dụng các thành phần animation đã xây dựng sẵn.
- Hình ảnh SVG có thể chứa liên kết đến bất kỳ tài liệu nào khác.
- SVG hầu như được hỗ trợ trên tất cả các trình duyệt.

Phần tử `<svg>` trong HTML là một vùng chứa để vẽ đồ họa.

SVG có một vài phương thức để vẽ như: đường thẳng, hình hộp, hình tròn, văn bản và hình ảnh đồ họa..

### 3.2. Vẽ đường tròn SVG

Để vẽ đường tròn ta dùng thẻ `<circle>`

```
<!DOCTYPE html>
<html>
<body>

<h2>Vẽ đường tròn trong SVG HTML5</h2>
<svg width="300" height="300">
  <circle cx="200" cy="100" r="80"
    stroke="lightcoral" stroke-width="4" fill="moccasin" />
</svg>

</body>
</html>
```

Các thuộc tính trong `<circle>` là:

- cx: vị trí tâm, tính từ mép trái của SVG.
- cy: vị trí tâm, tính từ mép trên.
- r: bán kính.
- fill: xác định màu tô.
- stroke: đường biên.

### 3.3. Vẽ hình ellipse SVG

Để vẽ hình ellipse ta sử dụng thẻ `<ellipse>`

```
<!DOCTYPE html>
<html>
<body>

<h2>Vẽ hình ellipse trong SVG HTML5</h2>
<svg width="300" height="300">
  <ellipse cx="150" cy="70" rx="100" ry="50"
    stroke="darkslateblue" stroke-width="4" fill="lightblue" />
</svg>

</body>
</html>
```

Các tham số tương tự khi vẽ hình tròn, thêm hai thuộc tính:

rx: bán kính từ tâm đến mép bên trái.

ry: bán kính từ tâm đến mép phía trên.

## 4. Sử dụng HTML Media

Các đặc trưng HTML5, bao gồm sự hỗ trợ audio và video tự nhiên mà không cần Flash.

Thẻ HTML5 <audio> và <video> làm nó đơn giản để thêm đa phương tiện tới một Website. Bạn cần thiết lập thuộc tính src để nhận diện nguồn phương tiện và bao gồm một thuộc tính control để người sử dụng có thể chơi và dừng đa phương tiện.

### 4.1. Nhúng video

Một phần tử video cho phép nhiều phần tử source và trình duyệt sẽ sử dụng định dạng được nhận ra đầu tiên:

```
<!DOCTYPE HTML>
<html>
<body>
  <video width="300" height="200" controls autoplay>
    <source src="http://www.vietjack.com/html5/foo.mp4" type="video/ogg" />
    <source src="http://www.vietjack.com/html5/foo.mp4" type="video/mp4" />
    Your browser does not support the <video> element.
  </video>
</body>
</html>
```

Ví dụ: play một video đã cho

```
<!DOCTYPE HTML>
<head>
<script type="text/javascript">
function PlayVideo(){
  var v = document.getElementsByTagName("video")[0];
  v.play();
}
</script>
</head>
<html>
<body>
  <form>
  <video width="300" height="200" src="http://www.vietjack.com/html5/foo.mp4">
    Your browser does not support the <video> element.
  </video>
  <input type="button" onclick="PlayVideo();" value="Play"/>
  </form>
</body>
</html>
```

## 4.2. Nhúng audio

Các định dạng audio được sử dụng phổ biến nhất là **ogg**, **mp3** và **wav**.

Một phần tử audio cho phép nhiều phần tử source và trình duyệt sẽ sử dụng định dạng mà nó nhận ra đầu tiên:

```
<!DOCTYPE HTML>
<html>
<body>
  <audio controls autoplay>
    <source src="http://www.vietjack.com/html5/audio.ogg" type="audio/ogg" />
    <source src="http://www.vietjack.com/html5/audio.wav" type="audio/wav" />
    Your browser does not support the <audio> element.
  </audio>
</body>
</html>
```

## CÂU HỎI, BÀI TẬP

Tạo trang web theo mẫu của bài 1 với cách sử dụng tối đa thẻ HTML5

# BÀI 3: HOÀN CHỈNH GIAO DIỆN TRANG WEB VỚI CSS

## Mã bài: 18.3

### Giới thiệu:

Bất kỳ trang web nào, ngoài hiển thị nội dung, nó cũng phải được định dạng, trang trí để thu hút và tiện lợi cho người xem. Không ngoại trừ có nhiều nội dung được định dạng giống nhau. Với CSS, người thiết kế web chỉ cần viết kịch bản định dạng một lần và sử dụng cho nhiều nội dung.

### Mục tiêu:

- Biết công dụng của CSS, các loại CSS
- Thiết kế được trang web với HTML, HTML5, và CSS
- Chăm thận, an toàn

### Nội dung chính:

## 1. Giới thiệu CSS và các loại CSS

### 1.1. Giới thiệu

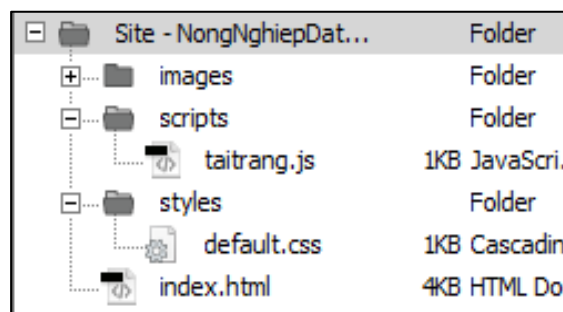
CSS (Cascading Style Sheet) là tài liệu định nghĩa các quy tắc định dạng cho các element trong trang web.

### 1.2. External

Quy tắc định dạng đặt trong một tập tin .css và được sử dụng trong trang thông qua thẻ `<link ... />`

Ví dụ:

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



Hình 3.1. Cấu trúc site



### 1.3. Internal

Quy tắc định dạng được đặt ngay trong trang web, đặt trong thẻ `<style>`  
... `</style>`

Ví dụ:

```
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
<style type="text/css">
body{
    font-family:Helvetica, Arial, sans-serif;
    font-size:13px;
}
</style>
</head>
```

### 1.4. Thuộc tính style

Quy tắc định dạng cũng có thể đặt trong thuộc tính style của thẻ mở.

Ví dụ:

```
<div style="font-weight:bold;">header</div>
<div>Nội dung</div>
```

## 2. Tạo và sử dụng CSS

- ID Selector: Dùng dấu # đặt trước tên quy tắc. Quy tắc định dạng sẽ áp dụng trên các element có thuộc tính id phù hợp.

Ví dụ:

Nội dung HTML

```
<p id="d1">Đoạn văn bản 1</p>
<p id="d2">Đoạn văn bản 2</p>
```

Nội dung CSS

```
#d1{
    color:#8F0002;
}
```

- Class Selector: dùng dấu chấm (.) đặt trước tên quy tắc. Quy tắc định dạng sẽ áp dụng trên các element có thuộc tính class phù hợp.

Ví dụ:

Nội dung HTML

```
<p>Đoạn văn bản</p>
<p class="c2">Đoạn văn bản n</p>
```

Nội dung CSS

```
.c2{
    color:#0A8E00;
}
```

- Tạo một số định dạng thông dụng
  - font-family, color: font và màu chữ
  - border[-top/right/bottom/left]: đường viền xung
  - padding[-top/right/bottom/left]: khoảng cách giữa đường viền với nội dung bên trong
  - margin[-top/right/bottom/left]: khoảng cách giữa vùng chứa với đường viền
  - background-color, background-image: màu nền, ảnh nền

### 3. Tạo định dạng chung cho trang web

```
body{
    font-family:Helvetica, Arial, sans-serif;
    font-size:13px;
}
a{
    color:#AA0002;
    text-decoration:none;
}
p{
    margin:5px 0px 0px 0px;
    padding:0px 0px 0px 0px;
}
th, td{
    vertical-align:top;
}
```

### 4. Tạo giao diện trang web canh giữa trình duyệt

Hầu hết các màn hình LCD phổ biến trên thị trường ngày nay là màn ảnh rộng. Tỷ lệ màn hình rộng này có thể gây ảnh hưởng tới các nhà thiết kế web và cách họ trình bày trang Web.

Với một đoạn mã Cascading Style Sheet (CSS) ngắn gọn, bạn có thể điều chỉnh thiết kế trang web vào vị trí trung tâm trên bất kỳ màn hình nào cho dù màn hình đó theo tỷ lệ nằm ngang hay thẳng đứng.

Tạo file .css như sau: để điều chỉnh sao cho các trình duyệt chạy như nhau và bỏ đi một số hiển thị không cần thiết (bỏ list-style cho <ol> và <ul>, bỏ border cho <img />)

```
* {
  margin: 0;
  padding: 0;
}

ol, ul { list-style: none; }

img { border: none; vertical-align: top; }

div#layout {
  border: 1px solid #a3a2a3;
  margin: 20px auto;
  width: 900px;
}
```

## 5. Tạo hiệu ứng đổi định dạng cho menu

```
<div id="gNav">
<ul>
<li><a href="#">TRANG CHỦ</a></li>
<li><a href="#">GIỚI THIỆU</a></li>
<li><a href="#">SẢN PHẨM</a></li>
<li><a href="#">DỊCH VỤ</a></li>
<li class="last"><a href="#">LIÊN HỆ</a></li>
</ul>
```

Viết CSS cho <div id="gNav">



```
div#gNav ul li {
  border-right: solid 1px #fff;
  display: inline;
  padding: 0 10px;
}
```

```
div#gNav ul li.last {
  border-right: none;
  padding-right: 0;
}
```

Thêm hiệu ứng cho gNav khi hover (di chuyển chuột đổi hiệu ứng)

```
div#gNav ul li a:hover {  
    color: yellow;  
}
```

## 6. Tạo khung viền bo tròn cho nội dung

Bốn giá trị – bán kính đường viền: 15px 50px 30px 5px; (giá trị đầu tiên áp dụng cho góc trên cùng bên trái, giá trị thứ hai áp dụng cho góc trên cùng bên phải, giá trị thứ ba áp dụng cho góc dưới cùng bên phải và giá trị thứ tư áp dụng cho góc dưới cùng bên trái):

Ba giá trị – bán kính đường viền: 15px 50px 30px; (giá trị đầu tiên áp dụng cho góc trên cùng bên trái, giá trị thứ hai áp dụng cho các góc trên cùng bên phải và dưới cùng bên trái và giá trị thứ ba áp dụng cho góc dưới cùng bên phải):

Hai giá trị – bán kính đường viền: 15px 50px; (giá trị đầu tiên áp dụng cho các góc trên cùng bên trái và dưới cùng bên phải và giá trị thứ hai áp dụng cho các góc trên cùng bên phải và dưới cùng bên trái):

Một giá trị – bán kính đường viền: 15px; (giá trị áp dụng cho tất cả bốn góc, được làm tròn như nhau):

```
1 #rcorners1 {  
2     border-radius: 15px 50px 30px 5px;  
3     background: #73AD21;  
4     padding: 20px;  
5     width: 200px;  
6     height: 150px;  
7 }  
8  
9 #rcorners2 {  
10    border-radius: 15px 50px 30px;  
11    background: #73AD21;  
12    padding: 20px;  
13    width: 200px;  
14    height: 150px;  
15 }  
16
```

```
17 #rcorners3 {  
18    border-radius: 15px 50px;  
19    background: #73AD21;  
20    padding: 20px;  
21    width: 200px;  
22    height: 150px;  
23 }  
24  
25 #rcorners4 {  
26    border-radius: 15px;  
27    background: #73AD21;  
28    padding: 20px;  
29    width: 200px;  
30    height: 150px;  
31 }
```

## CÂU HỎI, BÀI TẬP

Viết css cho bài tập của bài học số 2 của trang web mẫu:

3.1 Viết css cho phần header

3.2 Viết css cho button của phần header

3.3 Viết css cho các đường thẳng đứng trong danh sách link của phần header

## BÀI 4: TÙY BIẾN GIAO DIỆN VỚI CSS3

### Mã bài: 18.4

#### Giới thiệu:

CSS3 là tiêu chuẩn mới nhất của CSS, hoàn toàn tương thích với các phiên bản trước của CSS. Với CSS3, chúng ta có thể định dạng trang web đa màu sắc sinh động, tạo kịch bản hiệu ứng hoạt hình cho trang web mà không cần phải biết bất kỳ ngôn ngữ lập trình nào.

#### Mục tiêu:

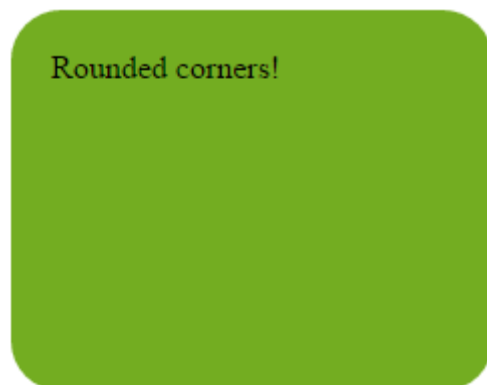
- Biết các thông tin định dạng của CSS3
- Thiết kế được giao diện trang web với HTML, HTML5 và CSS3
- Chăm thận, an toàn

#### Nội dung chính:

##### 1. Rounded Corners

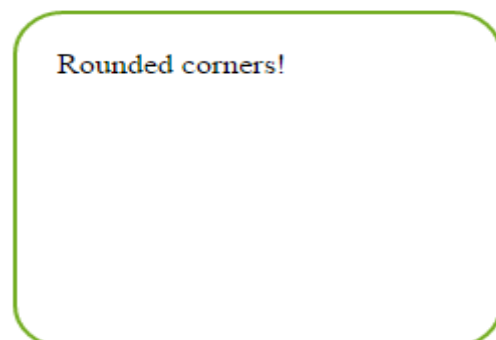
Tạo khung bo tròn các góc

```
#rcorners1 {  
    border-radius: 25px;  
    background: #73AD21;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}
```



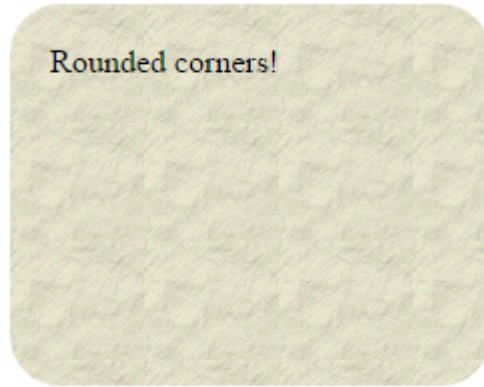
Hình 4.1. Kết quả rcorner1

```
#rcorners2 {  
    border-radius: 25px;  
    border: 2px solid #73AD21;  
    padding: 20px;  
    width: 200px;  
    height: 150px;  
}
```



Hình 4.2. Kết quả rcorner2

```
#rcorners3 {  
  border-radius: 25px;  
  background: url(paper.gif);  
  background-position: left top;  
  background-repeat: repeat;  
  padding: 20px;  
  width: 200px;  
  height: 150px;  
}
```



Hình 4.3. Kết quả rcorner3

## 2. Border Images

Dùng ảnh làm đường viền

Chuẩn bị ảnh border.png



Hình 4.4. Ảnh border.png

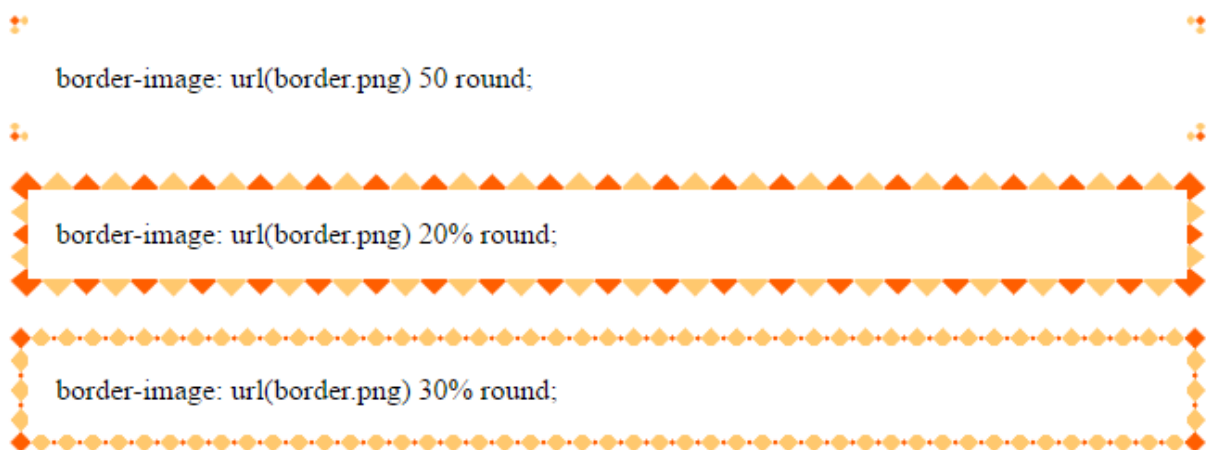
```

#borderimg1 {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 50 round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 50 round; /* Opera 11-12.1 */
    border-image: url(border.png) 50 round;
}

#borderimg2 {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 20% round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 20% round; /* Opera 11-12.1 */
    border-image: url(border.png) 20% round;
}

#borderimg3 {
    border: 10px solid transparent;
    padding: 15px;
    -webkit-border-image: url(border.png) 30% round; /* Safari 3.1-5 */
    -o-border-image: url(border.png) 30% round; /* Opera 11-12.1 */
    border-image: url(border.png) 30% round;
}

```



Hình 4.5. Kết quả của ảnh làm kẻ khung (border images)

### 3. Backgrounds

Ví dụ 1: Dùng nhiều ảnh làm nền

```

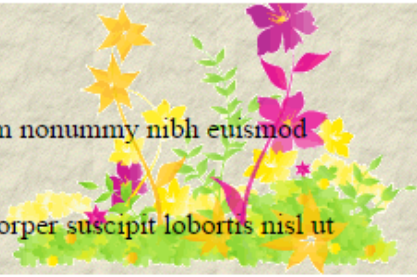
#example1 {
    background-image: url(img_flwr.gif), url(paper.gif);
    background-position: right bottom, left top;
    background-repeat: no-repeat, repeat;
}

```

# Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

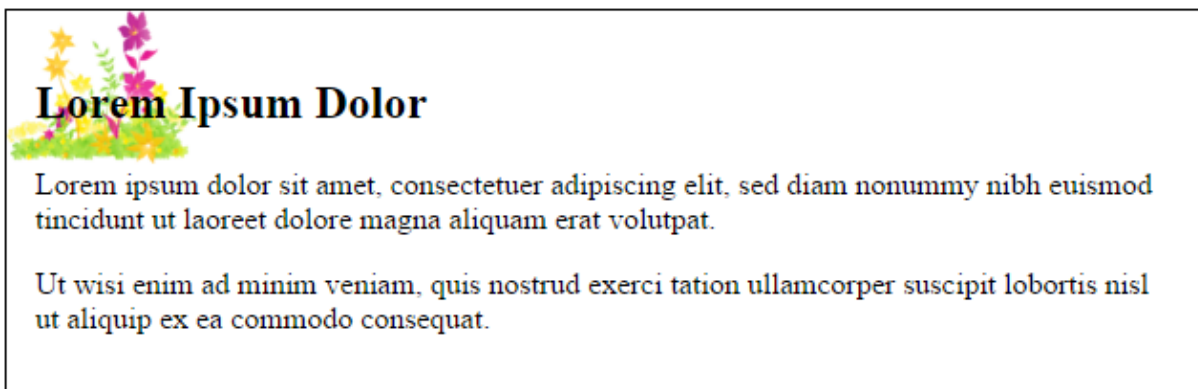
Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



Hình 4.6. Kết quả của ví dụ 1

Ví dụ 2: Thiết lập kích thước ảnh nền

```
#div1 {  
    background: url(img_flower.jpg);  
    background-size: 100px 80px;  
    background-repeat: no-repeat;  
}
```



Hình 4.7. Kết quả của ví dụ 2

Ví dụ 3: Sử dụng giá trị contain và cover để thiết lập kích thước ảnh nền

```
#div1 {  
    background: url(img_flower.jpg);  
    background-size: contain;  
    background-repeat: no-repeat;  
}  
#div2 {  
    background: url(img_flower.jpg);  
    background-size: cover;  
    background-repeat: no-repeat;  
}
```



Hình 4.8. Kết quả của div1





Hình 4.9. Kết quả của div2

Ví dụ 4: Thiết lập kích thước cho nhiều ảnh nền

```
#example1 {
    background: url(img_flwr.gif) left top no-repeat,
    url(img_flwr.gif) right bottom no-repeat,
    url(paper.gif) left top repeat;
    background-size: 50px, 130px, auto;
}
```



Hình 4.10. Kết quả của ví dụ 4

## 4. Colors

– Opacity

```
#p1 {background-color:rgb(255,0,0);opacity:0.6;}
#p2 {background-color:rgb(0,255,0);opacity:0.6;}
#p3 {background-color:rgb(0,0,255);opacity:0.6;}
```



– Hệ màu RGBA (Red-Green-Blue-Alpha)

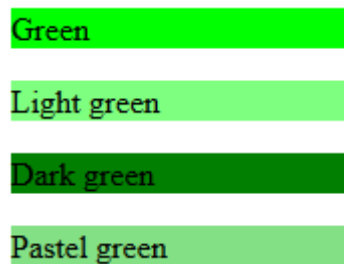
Là sự kết hợp giữa RGB và Opacity

```
#p1 {background-color: rgba(255, 0, 0, 0.3);}
#p2 {background-color: rgba(0, 255, 0, 0.3);}
#p3 {background-color: rgba(0, 0, 255, 0.3);}
```



– Hệ màu HSL (Hue-Saturation-Lightness)

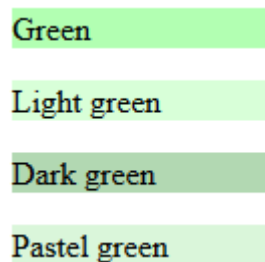
```
#p1 {background-color: hsl(120, 100%, 50%);}
#p2 {background-color: hsl(120, 100%, 75%);}
#p3 {background-color: hsl(120, 100%, 25%);}
#p4 {background-color: hsl(120, 60%, 70%);}
```



– Hệ màu HSLA (Hue-Saturation-Lightness-Alpha)

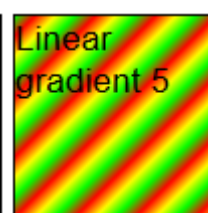
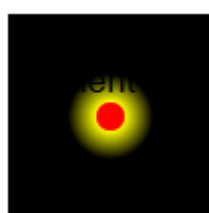
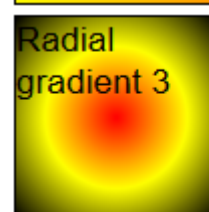
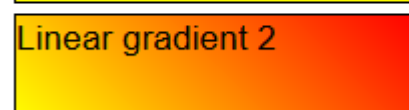
Là sự kết hợp giữa HSL và Opacity

```
#p1 {background-color: hsla(120, 100%, 50%, 0.3);}
#p2 {background-color: hsla(120, 100%, 75%, 0.3);}
#p3 {background-color: hsla(120, 100%, 25%, 0.3);}
#p4 {background-color: hsla(120, 60%, 70%, 0.3);}
```



## 5. Gradients

```
.grad1{
  width:200px;
  height:50px;
  background:linear-gradient(to bottom, #f00, #ff0);
}
.grad2{
  width:200px;
  height:50px;
  background:linear-gradient(225deg, #f00, #ff0);
}
.grad3{
  width:100px;
  height:100px;
  background:radial-gradient(#f00,#ff0,#000);
}
.grad4{
  width:100px;
  height:100px;
  background:radial-gradient(#f00 10%,#ff0 5%,#000 30%);
}
.grad5{
  width:100px;
  height:100px;
  background:repeating-linear-gradient(to bottom right,
  #0f0, #f00 5%, #ff0 10%, #0f0 15%);
}
```



## 6. Shadows

### – Text shadow

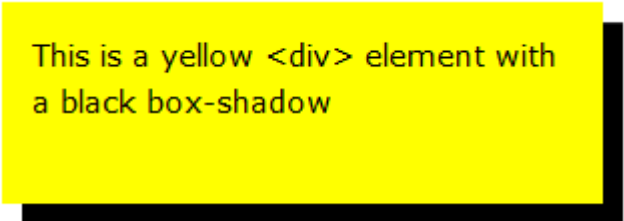
```
.sh1{
  font-size:40px;
  font-weight:bold;
  text-shadow: 2px 2px;
}
.sh2{
  font-size:40px;
  font-weight:bold;
  text-shadow: 2px 2px rgba(255,0,0,1.00);
}
.sh3{
  font-size:40px;
  font-weight:bold;
  color:rgba(255,0,0,1.00);
  text-shadow: 2px 2px 5px rgba(0,0,0,1.00);
}
```

**Text shadow 1**  
**Text shadow 2**  
**Text shadow 3**

### – Box shadow

Ví dụ 1:

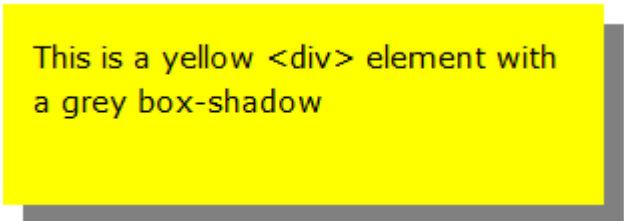
```
div {
  box-shadow: 10px 10px;
}
```



This is a yellow <div> element with a black box-shadow

Ví dụ 2:

```
div {
  box-shadow: 10px 10px grey;
}
```



This is a yellow <div> element with a grey box-shadow

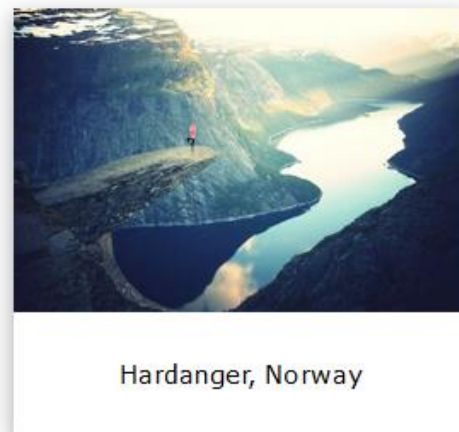
Ví dụ 3:

```
div {  
    box-shadow: 10px 10px 5px grey;  
}
```

This is a yellow <div> element with a blurred, grey box-shadow

Ví dụ 4:

```
div.card {  
    width: 250px;  
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);  
    text-align: center;  
}
```



Hình 4.11. Kết quả của ví dụ 4

## 7. Text

### – Text Overflow

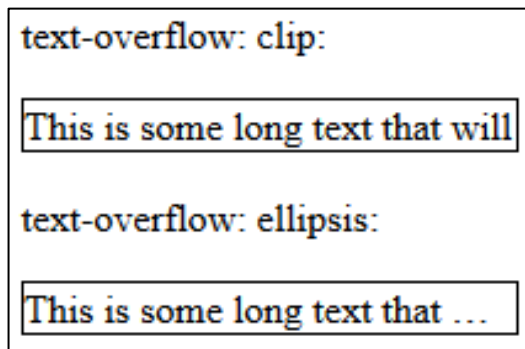
Xử lý hiển thị khi nội dung vượt quá kích thước vùng chứa

```

p.test1 {
  white-space: nowrap;
  width: 200px;
  border: 1px solid #000000;
  overflow: hidden;
  text-overflow: clip;
}

p.test2 {
  white-space: nowrap;
  width: 200px;
  border: 1px solid #000000;
  overflow: hidden;
  text-overflow: ellipsis;
}

```



Hình 4.12. Kết quả của Text Overflow

– Word Wrapping

Xử lý ngắt xuống dòng với những từ dài

Nội dung trong HTML

```

<p class="test"> This paragraph contains a
very long word:
thisisaveryveryveryveryveryverylongword. The
long word will break and wrap to the next
line.</p>

```

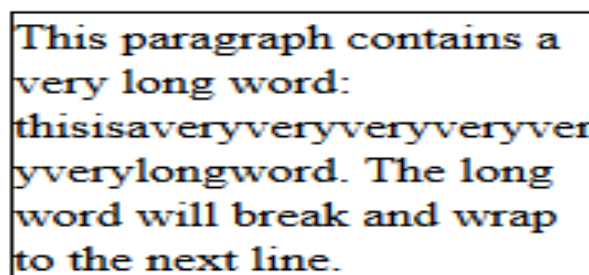
Nội dung CSS

```

p.test {
  width: 11em;
  border: 1px solid #000000;
  word-wrap: break-word;
}

```

Kết quả hiển thị



– Word Breaking: xử lý giữ hay ngắt những từ dài

```

p.test1 {
  width: 140px;
  border: 1px solid #000000;
  word-break: keep-all;
}

p.test2 {
  width: 140px;
  border: 1px solid #000000;
  word-break: break-all;
}

```

This paragraph contains some text. This line will-break-at-hyphens.

This paragraph contains some text. The lines will break at any character.

## 8. Fonts

```

@font-face {
  font-family: myFirstFont;
  src: url(sansation_light.woff);
}

.cf {
  font-family: myFirstFont;
}

```

With CSS3, websites can finally use fonts other than the pre-selected "web-safe" fonts.

## 9. 2D Transforms

### – Tịnh tiến (translate)

```

.transform1{
  width:200px;
  height:100px;
  background-color:#FF6366;
}
.transform1 header{
  background-color:#FFED00;
  width:100px;
  padding:10px 0px;
  font-weight:bold;
  text-align:center;
  transform:translate(50px,-20px);
}
.transform1 p{
  margin:0; padding:0;
}

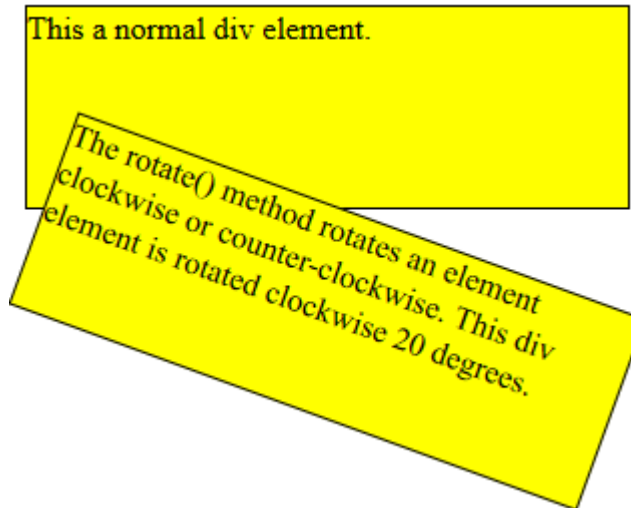
```

**TIÊU ĐỀ**

nội dung

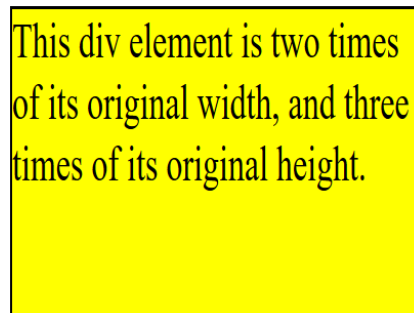
### – Xoay (rotate)

```
div {
  -ms-transform: rotate(20deg); /* IE 9 */
  -webkit-transform: rotate(20deg); /* Safari */
  transform: rotate(20deg);
}
```



– Kéo kích thước (scale)

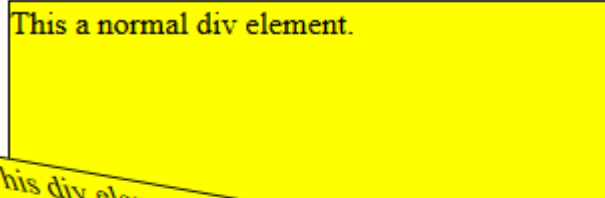
```
div {
  -ms-transform: scale(2,3); /* IE 9 */
  -webkit-transform: scale(2,3); /* Safari */
  transform: scale(2,3);
}
```



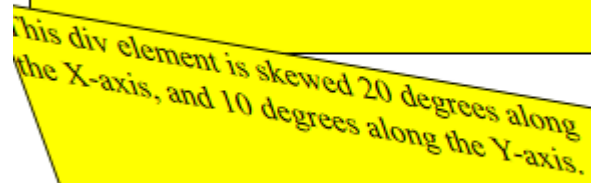
– Kéo xiên (skew)

```
div {
  -ms-transform: skew(20deg, 10deg); /* IE 9 */
  -webkit-transform: skew(20deg, 10deg); /* Safari */
  transform: skew(20deg, 10deg);
}
```





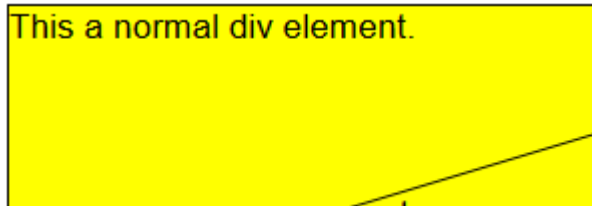
This a normal div element.



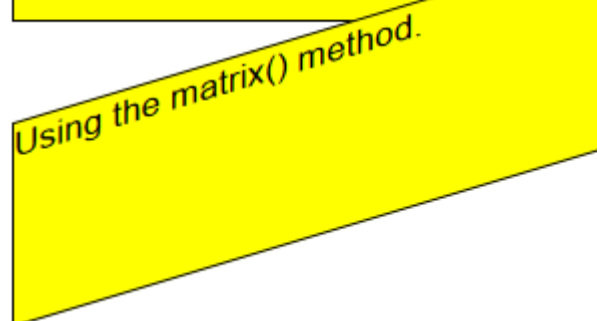
This div element is skewed 20 degrees along the X-axis, and 10 degrees along the Y-axis.

- Ma trận biến đổi (matrix): kết hợp Scale, Skew và Translate  
`matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())`

```
div {  
  width: 300px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
div#myDiv1 {  
  -ms-transform: matrix(1, -0.3, 0, 1, 0, 0);  
  -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0);  
  transform: matrix(1, -0.3, 0, 1, 0, 0);  
}
```



This a normal div element.



Using the matrix() method.

## 10. 3D Transforms

- Xoay xung quanh trục X (`rotateX`)

```
div {
  -webkit-transform: rotateX(150deg);
  transform: rotateX(150deg);
}
```

The rotateX() method rotates an element around its X-axis at a given degree. This div element is rotated 150 degrees.

– Xoay xung quanh trục Y (rotateY)

```
div {
  -webkit-transform: rotateY(130deg);
  transform: rotateY(130deg);
}
```

The rotateY() method rotates an element around its Y-axis at a given degree. This div element is rotated 130 degrees.

– Xoay xung quanh trục Z (rotateZ)

```
div {
  -webkit-transform: rotateZ(90deg);
  transform: rotateZ(90deg);
}
```

This a normal d

The rotateZ() method rotates an element around its Z-axis at a given degree. This div element is rotated 90 degrees.

Note: Internet Explorer (earlier versions) does not support the rotateZ() method.

## 11. Transitions

Ví dụ 1: Khi di chuyển chuột lên hình vuông thì chiều ngang sẽ thay đổi đến 300px trong vòng 2 giây.

```
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s;
    transition: width 2s;
}

div:hover {
    width: 300px;
}
```

Ví dụ 2: Chiều ngang thay đổi đến 300px trong 2 giây, chiều cao thay đổi đến 300px trong 4 giây.

```
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s, height 4s;
    transition: width 2s, height 4s;
}

div:hover {
    width: 300px;
    height: 300px;
}
```

Ví dụ 3: Chờ 1 giây, chiều ngang thay đổi đến 300px trong 3 giây tiếp theo.

```
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 3s;
    -webkit-transition-delay: 1s;
    transition: width 3s;
    transition-delay: 1s;
}

div:hover {
    width: 300px;
}
```

Ví dụ 4: Thay đổi chiều ngang, cao trong 2 giây, xoay trong 3 giây.

```

div {
  width: 100px;
  height: 100px;
  background: red;
  -webkit-transition: width 2s, height 2s, -webkit-transform 2s;
  transition: width 2s, height 2s, transform 3s;
}

div:hover {
  width: 300px;
  height: 300px;
  -webkit-transform: rotate(180deg); /* Safari */
  transform: rotate(180deg);
}

```

## 12. Animations

Ví dụ: Hình vuông sẽ chuyển từ màu đỏ sang màu vàng trong 2 giây

```

div{
  width: 100px;
  height: 100px;
  background-color: red;
}
div:hover {
  -webkit-animation-name: example; /* Chrome, Safari, Opera */
  -webkit-animation-duration: 2s; /* Chrome, Safari, Opera */
  animation-name: example;
  animation-duration: 2s;
}

/* Chrome, Safari, Opera */
@-webkit-keyframes example {
  background-color: yellow;
}

/* Standard syntax */
@keyframes example {
  from {}
  to {background-color: yellow;}
}

```

## 13. Box Sizing

Quy định, kích thước của element:

Rộng = width + padding + border

Cao = height + padding + border

→ Trở ngại cho người thiết kế web khi viết CSS

→ Thuộc tính box-sizing của CSS3 đảm bảo đúng padding, border mà vẫn đảm bảo kích thước đúng với giá trị của width và height.

```
.div1 {
  width: 100px;
  height: 50px;
  padding: 20px;
  border: 1px solid blue;
}
```

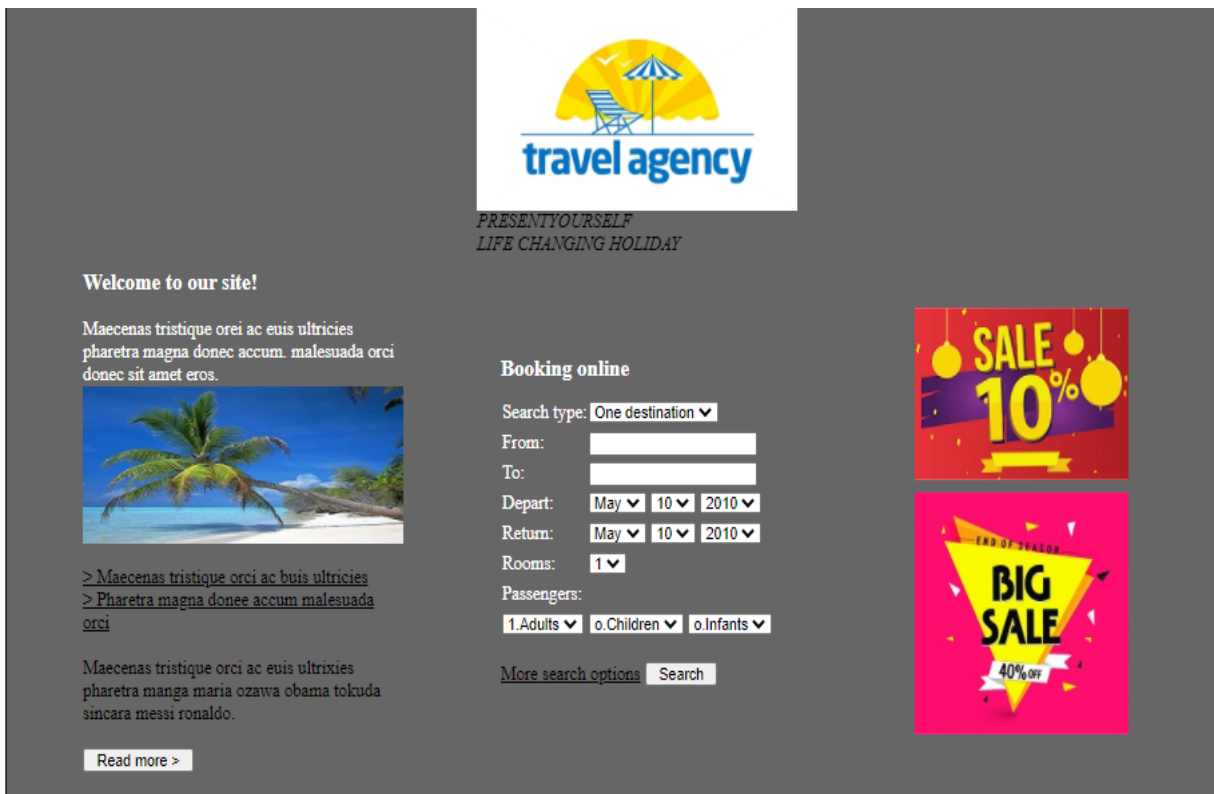
Both divs are the same size now!

```
.div2 {
  width: 100px;
  height: 50px;
  padding: 20px;
  border: 1px solid red;
  box-sizing: border-box;
}
```

Hooray!

## CÂU HỎI, BÀI TẬP

### 5.1 Thiết kế trang web có giao diện như hình bên dưới



5.2 Thiết kế trang web theo mẫu sau:



# BÀI 5: XỬ LÝ TƯƠNG TÁC VỚI JAVASCRIPT

Mã bài: 18.5

## Giới thiệu:

Javascript là ngôn ngữ lập trình xử lý trên tài liệu HTML, được thực thi trên trình duyệt. Ngoài sử dụng HTML, CSS, người thiết kế cũng phải biết lập trình Javascript để tăng thêm hiệu ứng sinh động cho trang web.

## Mục tiêu:

- Có kiến thức cơ bản về Javascript
- Biết lập trình nhúng xử lý tương tác cơ bản trên trang web bằng Javascript
- Cẩn thận, an toàn

## Nội dung chính:

### 1. Tổng quan Javascript

#### 1.1. Các cách nhúng Javascript vào trang web

Đoạn Javascript có thể được đặt trong `<body> ... </body>` hoặc `<head> ... </head>` của tài liệu HTML.

Cách 1: Viết trực tiếp trong tài liệu HTML

```
<script type="text/javascript">
/*Đoạn mã javascript*/
</script>
```

Cách 2: Viết các lệnh Javascript trong một tập tin script có phần mở rộng .js sau đó chèn vào tài liệu HTML.

```
<script type="text/javascript"
src="Scripts/jquery.js">
</script>
```

#### 1.2. Kiểu dữ liệu, khai báo biến

- Khai báo biến: `var <tên_biến>;`

```
var x = 5;
var y = 6;
var z = x + y;
```

- Kiểu dữ liệu: javascript không quan tâm đến kiểu dữ liệu của biến khi mới khai báo. Biến trong Javascript có thể lưu trữ giá trị các kiểu dữ liệu: số (number), chuỗi (string), mảng (array), đối tượng (object), ...

```
var length = 16;           // Number
var lastName = "Johnson"; // String
var cars = ["Saab", "Volvo", "BMW"]; // Array
var x = {firstName:"John", lastName:"Doe"}; // Object
```

- Javascript linh động trong việc xử lý kiểu dữ liệu khi thực hiện các phép tính.

Ví dụ:

```
var x = 16 + "Volvo";
```

→ Kết quả: 16Volvo

```
var x = 16 + 4 + "Volvo";
```

→ Kết quả: 20Volvo

- Kiểu số (number): Javascript không quan tâm số thực/nguyên. Javascript chỉ có một kiểu số.

Ví dụ 1:

```
var x1 = 34.00;
var x2 = 34;
var y = 123e5; // 12300000
var z = 123e-5; // 0.00123
```

- Kiểu luận lý (boolean)

```
var x = true;
var y = false;
```

- Kiểu mảng (array)

```
var cars = ["Saab", "Volvo", "BMW"];
```

Chỉ số phần tử đầu tiên của mảng là 0

→ cars[0] là "Saab"

- Kiểu đối tượng (object)

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

Trong ví dụ trên, đối tượng person có 4 thuộc tính: firstName, lastName, age và eyeColor.



- Toán tử `typeof`: cho biết kiểu dữ liệu của biến/biểu thức.

```
typeof "John"           // Returns string
typeof 3.14             // Returns number
typeof false           // Returns boolean
typeof [1,2,3,4]       // Returns object
typeof {name:'John', age:34} // Returns object
```

- Trong Javascript, một biến chưa được gán giá trị thì sẽ có giá trị là **undefined** và toán tử `typeof` cũng sẽ trả về **undefined**.

- Giá trị rỗng

```
var car = "";           // The value is "", the typeof is string
```

- Giá trị **null**

```
var person = null;     // Value is null, but type is still an object
var person = undefined; // Value is undefined, type is undefined
```

- Sự khác nhau giữa **undefined** và **null**

```
typeof undefined      // undefined
typeof null           // object
null === undefined    // false
null == undefined     // true
```

### 1.3. Các toán tử

- Các toán tử trên số

Toán tử	Mô tả
+	Cộng
-	Trừ
*	Nhân
/	Chia
%	Lấy số dư của phép chia 2 số nguyên
++	Tăng 1 đơn vị
--	Giảm 1 đơn vị

- Các toán tử gán

Toán tử	Ví dụ	Tương đương
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

– Toán tử nối chuỗi

```
txt1 = "John";
txt2 = "Doe";
txt3 = txt1 + " " + txt2;

txt1 = "What a very ";
txt1 += "nice day";
```

– Nối chuỗi và số

```
x = 5 + 5;
y = "5" + 5;
z = "Hello" + 5;
```

– Toán tử so sánh và luận lý

Toán tử	Mô tả
==	So sánh bằng
===	So sánh cùng giá trị và kiểu dữ liệu
!=	Khác giá trị
!==	Khác giá trị hoặc kiểu dữ liệu
>	So sánh lớn hơn
<	So sánh nhỏ hơn
>=	So sánh lớn hơn hoặc bằng
<=	So sánh nhỏ hơn hoặc bằng

Toán tử	Mô tả	Ví dụ
&&	Và	(x < 10 && y > 1) is true
	Hoặc	(x == 5    y == 5) is false
!	Phủ định	!(x == y) is true

– Biểu thức điều kiện

*<Biến> = (<Biểu thức ĐK>) ?<Giá trị khi ĐK đúng>:<Giá trị khi ĐK sai>*

Ví dụ:

```
var voteable = (age < 18) ? "Too young":"Old enough";
```

– Toán tử trên kiểu dữ liệu

Toán tử	Mô tả
typeof	Trả về kiểu dữ liệu của biến/biểu thức
instanceof	Trả về true nếu là đối tượng là một thực thể của object

## 2. Sử dụng cấu trúc điều khiển

### 2.1. if, if ... else ..., switch

```
if (condition) {  
    block of code to be executed if the condition is true  
}
```

Ví dụ:

```
if (hour < 18) {  
    greeting = "Good day";  
}
```

```
if (condition) {  
    block of code to be executed if the condition is true  
} else {  
    block of code to be executed if the condition is false  
}
```

Ví dụ:

```
if (hour < 18) {  
    greeting = "Good day";  
} else {  
    greeting = "Good evening";  
}
```

```
if (condition1) {  
    block of code to be executed if condition1 is true  
} else if (condition2) {  
    block of code to be executed if the condition1 is false and condition2 is true  
} else {  
    block of code to be executed if the condition1 is false and condition2 is false  
}
```

Ví dụ:

```
if (time < 10) {  
    greeting = "Good morning";  
} else if (time < 20) {  
    greeting = "Good day";  
} else {  
    greeting = "Good evening";  
}
```

### 2.2. for, while, break, continue

- Vòng lặp for

```
for (statement 1; statement 2; statement 3) {
    code block to be executed
}
```

Ví dụ:

```
for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}
```

– Vòng lặp for/in: duyệt từng thuộc tính trong đối tượng

```
var person = {fname:"John", lname:"Doe", age:25};

var text = "";
var x;
for (x in person) {
    text += person[x];
}
```

– Vòng lặp while

```
while (condition) {
    code block to be executed
}
```

Ví dụ:

```
while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

– Vòng lặp do/while

```
do {
    code block to be executed
}
while (condition);
```

Ví dụ:

```
do {
    text += "The number is " + i;
    i++;
}
while (i < 10);
```

– Lệnh break, continue

- break: thoát khỏi vòng lặp
- continue: bỏ qua các lệnh bên dưới của vòng lặp hiện tại, tiếp tục vòng lặp tiếp theo

### 3. Sử dụng hàm và mảng

#### 3.1. Mảng

- Tạo mảng

```
var array-name = [item1, item2, ...];
```

Ví dụ:

```
var cars = ["Saab", "Volvo", "BMW"];
```

- Dùng từ khóa new

```
var cars = new Array("Saab", "Volvo", "BMW");
```

- Truy cập phần tử trong mảng

```
var name = cars[0];
cars[0] = "Opel";
```

- Lưu trữ mảng dưới dạng đối tượng

Mảng:

```
var person = ["John", "Doe", 46];
```

Đối tượng:

```
var person = {firstName:"John", lastName:"Doe", age:46};
```

- Thuộc tính và phương thức trên mảng

- Thuộc tính length: cho biết số phần tử trong mảng

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.length; // the length of fruits is 4
```

- Phương thức sort: sắp xếp mảng

```
var y = cars.sort();
```

- Thêm phần tử vào mảng

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Lemon"); // adds a new element (Lemon) to fruits
```

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits[fruits.length] = "Lemon"; // adds a new element (Lemon) to fruits
```

- Duyệt phần tử trong mảng

```
var index;
var fruits = ["Banana", "Orange", "Apple", "Mango"];
for (index = 0; index < fruits.length; index++) {
    text += fruits[index];
}
```

### 3.2. Một số hàm toán học

- Lấy giá trị tuyệt đối

```
a=-4;
b=Math.abs(a); //b=4
```

- Tìm giá trị nhỏ nhất

```
var x = Math.min(0, 150, 30, 20, -8, -200);
```

- Tìm giá trị lớn nhất

```
var y = Math.max(0, 150, 30, 20, -8, -200);
```

- Nhận giá trị ngẫu nhiên trong đoạn [0, 1)

```
var z = Math.random();
```

- Làm tròn đến số nguyên gần nhất

```
var a = Math.round(4.7); //a=5
var b = Math.round(4.4) //b=4
```

- Làm tròn lên số nguyên gần nhất

```
a = Math.ceil(4.4); //a=5
```

- Làm tròn xuống số nguyên gần nhất

```
b = Math.floor(4.7); //b=4
```

- Lấy căn bậc 2

```
var c = Math.sqrt(9); //c=3
```

- Các hằng số

```

Math.E          // cơ số e
Math.PI         // giá trị PI
Math.SQRT2     // căn bậc 2 của 2
Math.SQRT1_2   // căn bậc 2 của 1/2
Math.LN2       // logarithm tự nhiên của 2
Math.LN10      // logarithm tự nhiên của 10
Math.LOG2E     // logarithm 2 của E
Math.LOG10E    // logarithm 10 của E

```

### 3.3. Dữ liệu Date

- Tạo đối tượng date: có 4 cách

```

new Date()
new Date(milliseconds)
new Date(dateString)
new Date(year, month, day, hours, minutes, seconds, milliseconds)

```

Ví dụ:

```

var d1= new Date();
var d2 = new Date(2016,3,27);
var d3 = new Date(2016,3,27,10,30,0,0);

```

- Các hàm đọc trên date

Hàm	Mô tả
getDate()	Ngày trong tháng (1-31)
getDay()	Ngày trong tuần (0-6). 0 tương ứng với Chủ Nhật
getFullYear()	Năm (yyyy)
getHours()	Giờ (0-23)
getMilliseconds()	Số milli giây (0-999)
getMinutes()	Số phút (0-59)
getMonth()	Tháng (0-11)
getSeconds()	Số giây (0-59)
getTime()	Tổng số milli giây (bắt đầu từ ngày 1 tháng 1 năm 1970)

Ví dụ:



```

var ngay = d1.getDay();
var s="";
switch (ngay){
    case 0: s="Chủ Nhật"; break;
    case 1: s="Thứ Hai"; break;
    case 2: s="Thứ Ba"; break;
    case 3: s="Thứ Tư"; break;
    case 4: s="Thứ Năm"; break;
    case 5: s="Thứ Sáu"; break;
    default: s="Thứ Bảy"; break;
}

```

– Các hàm ghi trên date

Hàm	Mô tả
setDate()	Cập nhật ngày trong tháng (1-31)
setFullYear()	Cập nhật năm (tùy chọn tháng và ngày)
setHours()	Cập nhật giờ (0-23)
setMilliseconds()	Cập nhật số milli giây (0-999)
setMinutes()	Cập nhật phút (0-59)
setMonth()	Cập nhật tháng (0-11)
setSeconds()	Cập nhật số giây (0-59)
setTime()	Cập nhật tổng thời gian (bắt đầu từ ngày 1 tháng 1 năm 1970)

### 3.4. Xây dựng hàm

Hàm được định nghĩa bắt đầu bằng từ khóa function, theo sau là tên hàm, sau đó là các tham số được đặt trong cặp dấu (). Hàm có thể có 1 hoặc nhiều tham số hoặc không có tham số. Các lệnh thực thi được đặt trong cặp dấu {}.

– Cấu trúc:

```

function <tên hàm>([ts1[,tsi]]) {
    Lệnh javascript
}

```

Các lệnh thực thi sẽ được thực hiện khi có lời gọi hàm (đúng quy tắc).

Ví dụ:

```

function DocSo(i){
    switch (i){
        case 0: alert("Không"); break;
        case 1: alert("Một"); break;
        case 2: alert("Hai"); break;
        default: alert("bé chưa học"); break;
    }
}
DocSo(1);

```

– Hàm trả về giá trị

```

function GiaiThua(n){
    var kq=1;
    var i;
    for (i=1; i<=n; i++)
        kq*=i;
    return kq;
}
var gtx = GiaiThua(5);

```

## CÂU HỎI, BÀI TẬP

- 5.1. Thiết kế trang web hiển thị số tuổi sau khi nhập năm sinh.
- 5.2. Thiết kế trang web hiển thị chu vi và diện tích hình chữ nhật.
- 5.3. Thiết kế trang web giải và biện luận phương trình  $ax^2 + bx + c = 0$ .

## YÊU CẦU KIỂM TRA, ĐÁNH GIÁ KẾT QUẢ HỌC TẬP

Xây dựng hàm tính toán

Hướng dẫn: tham khảo hàm `getElementById`, thuộc tính `value`, `innerHTML`

# BÀI 6: MỘT SỐ KỸ THUẬT THIẾT KẾ NÂNG CAO

## Mã bài: 18.6

### Giới thiệu:

HTML DOM (Document Object Model) là mô hình đối tượng chuẩn và gia diện lập trình cho tài liệu HTML. Với HTML DOM, Javascript có thể truy cập, thay đổi toàn bộ (bao gồm nội dung, thuộc tính) các element bên trong tài liệu HTML.

### Mục tiêu:

- Biết các thuộc tính, sự kiện của các element trong trang web
- Thay đổi nội dung, định dạng các element bằng Javascript
- Lập trình được tương tác với người dùng trên trang web bằng Javascript
- Chăm thận, an toàn

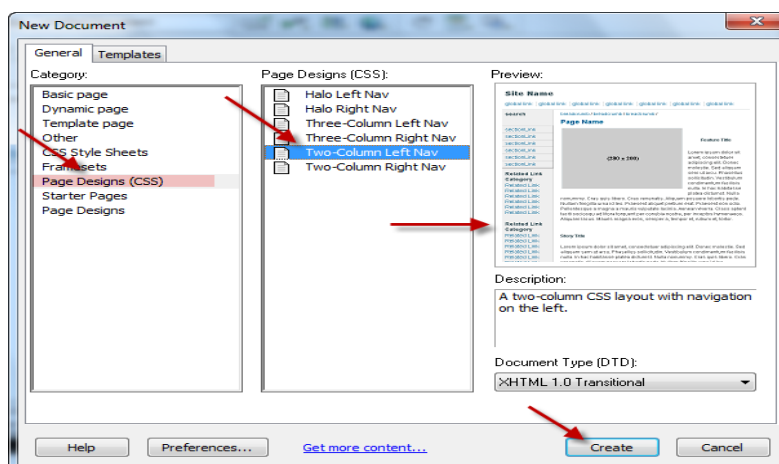
### Nội dung chính:

#### 1. Template trong Dreamwaver

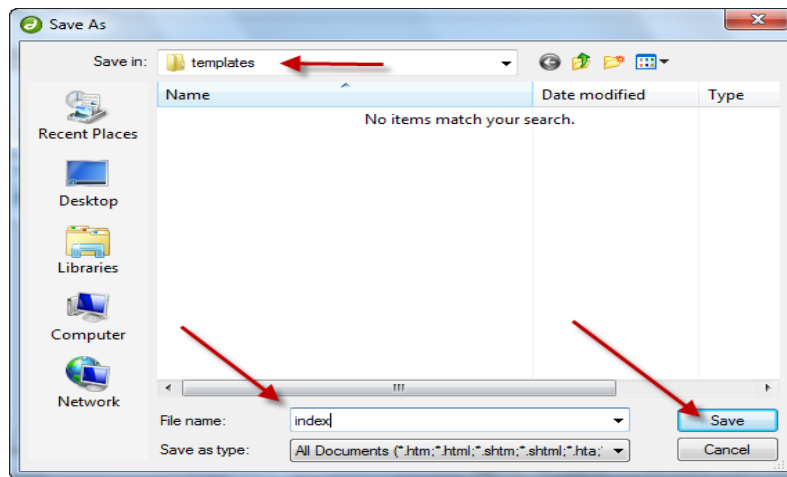
Bạn cũng có thể tạo một website bằng template dựng sẵn của Dreamweaver. Với template, bạn sẽ đi trước một bước và tận dụng ưu điểm của một site hoàn chỉnh và tham khảo code bên trong của nó.

##### 1.1. Chọn template mẫu của Dreamwaver

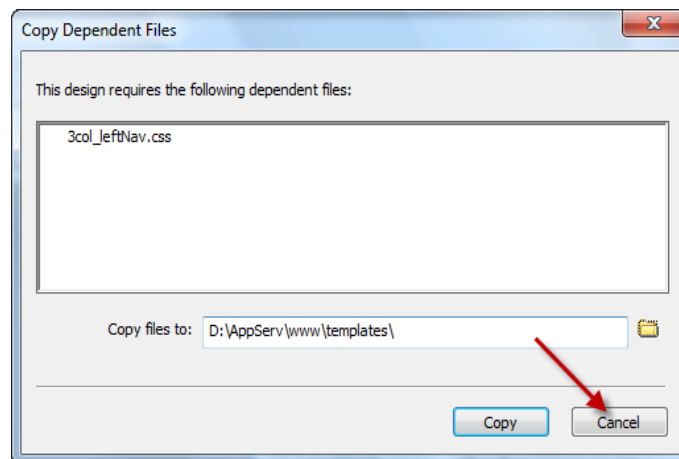
Vào File > New > Tab Generate chọn Page Design (CSS) > Cột giữa chọn Two Column Left Nav > Bấm Create.



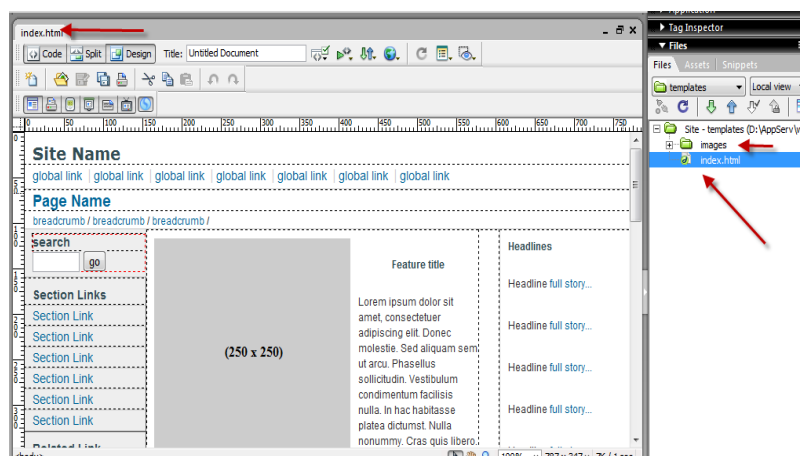
Xuất hiện CS Save As, bạn thấy Dòng Save in đã có tên Site bạn vừa tạo, nhập tên: index bấm Save.



Xuất hiện CS Copy Dependent Files, trong CS này có chứa các files nhất là files ảnh, bạn bấm Cancel . Hoặc bạn cò thể tạo Folder images để chứa các ảnh.



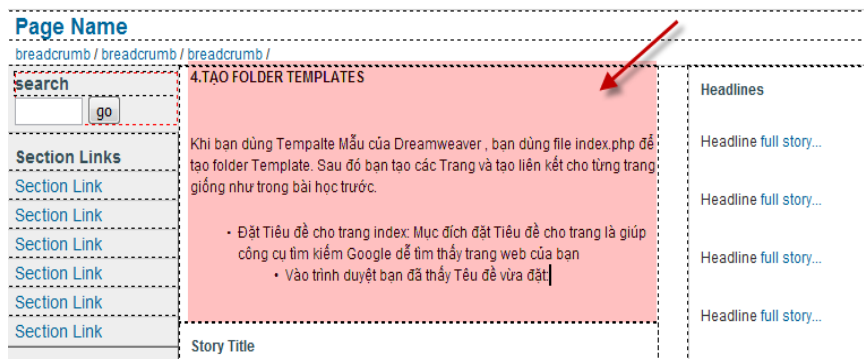
Sau khi tạo xong bạn thấy trong Cột Quản lý Files đã có Folder images và file index.php. Bạn có thể rê các files ảnh vào thư mục images. Bấm lên file index.php xuất hiện nội dung bên trái. Bấm Save All và trình duyệt để xem kết quả.



## 1.2. Chỉnh sửa tên menu

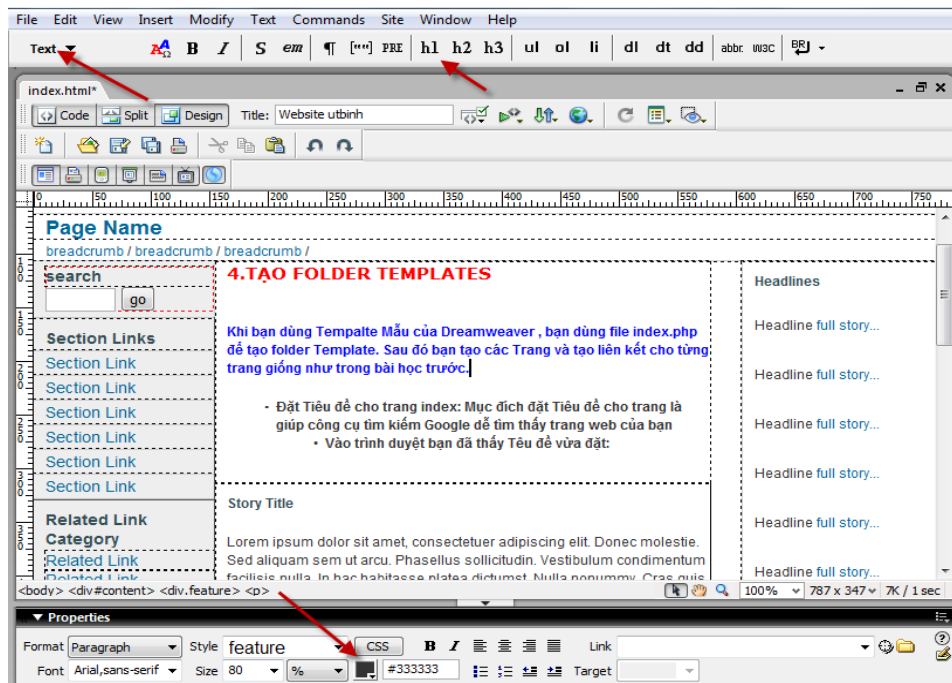
Bước 1: Bôi đen tên Menu đầu tiên





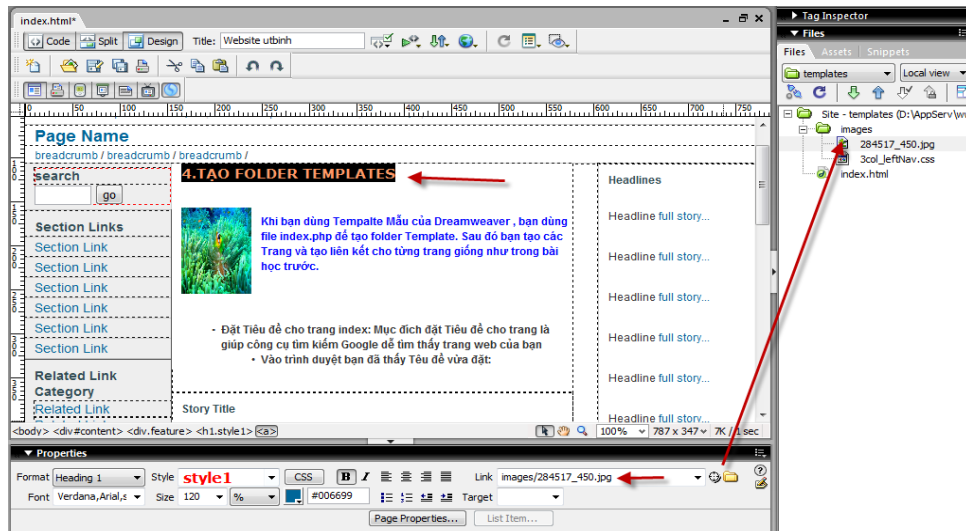
## 1.4. Định dạng văn bản

Trên Thanh chèn (Insert Bar) chọn Text từ Menu bật lên. Bạn sử dụng Thanh chèn và Hộp Kiểm Properties để định dạng văn bản.



## 1.5. Bổ sung hình ảnh vào trang

Chỉ định vị trí chèn hình ảnh > Bật Thanh chèn Common > Bấm nút Image > Đến nơi cần lấy ảnh > Mở trình duyệt xem



## 2. Navigation và Menu

### 2.1. Tạo thanh điều hướng

Với CSS, bạn có thể biến các thanh menu nhàm chán trở nên đẹp mắt hơn.

Cách tạo navigation bằng danh sách sử dụng các thẻ `<ul>` và `<li>`.

```

<ul>
  <li><a href="https://quantrimang.com/">Trang chủ</a></li>
  <li><a href="https://quantrimang.com/lang-cong-nghe">Làng Công nghệ</a></li>
  <li><a href="https://quantrimang.com/cong-nghe">Công nghệ</a></li>
  <li><a href="https://quantrimang.com/khoa-hoc">Khoa học</a></li>
  <li><a href="https://quantrimang.com/cuoc-song">Cuộc sống</a></li>
</ul>

```

```

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #e9d8f4;
}

li a {
  display: block;
  color: white;
  padding: 8px 16px;
  text-decoration: none;
}

```

```

/* Thay đổi màu liên kết khi di chuột qua */
li a:hover {
  background-color: #db7093;
  font-weight: bold;
  color: white;
}

```

## 2.2. Canh chỉnh đường dẫn và thêm đường viền cho menu

Thêm `text-align:center` vào `<li>` hoặc `<a>` để căn chỉnh đường dẫn nằm vào giữa khối.

Thêm thuộc tính `border` vào `<ul>` để viền quanh thanh điều hướng. Nếu bạn cũng muốn đường viền bên trong thanh điều hướng thì thêm `border-bottom` vào tất cả phần tử `<li>` trừ thẻ cuối cùng:

```

ul {
  border: 1px solid #db7093;
}

li {
  text-align: center;
  border-bottom: 1px solid #db7093;
}

li:last-child {
  border-bottom: none;
}

```

## 2.3. Cố định chiều cao thanh điều hướng dọc

Tạo một thanh điều hướng xác định chiều cao, cố định ở bên trái:



```

ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 25%;
  background-color: #f1f1f1;
  height: 100%; /* chiều cao 100% */
  position: fixed; /* cố định thanh điều hướng cả khi cuộn để đọc website */
  overflow: auto; /* nếu nội dung quá dài sẽ có thêm thanh cuộn để xem phần còn lại */
}

```

### 3. Đối tượng nâng cao trong Javascript

HTML DOM (Document Object Model) là mô hình đối tượng chuẩn và giao diện lập trình cho tài liệu HTML. Với HTML DOM, Javascript có thể truy cập, thay đổi toàn bộ (bao gồm nội dung, thuộc tính) các element bên trong tài liệu HTML.

#### 3.1. Xử lý trên HTML DOM

##### 3.1.1. Tìm element

Hàm	Mô tả
<code>document.getElementById(<i>id</i>)</code>	Tìm 1 element bằng thuộc tính id
<code>document.getElementsByTagName(<i>name</i>)</code>	Tìm các element bằng tên thẻ
<code>document.getElementsByClassName(<i>name</i>)</code>	Tìm các element bằng tên class

##### 3.1.2. Cập nhật nội dung, thuộc tính

Lệnh	Mô tả
<code>element.innerHTML = new html content</code>	Cập nhật nội dung HTML của element
<code>element.attribute = new value</code>	Thay giá trị của thuộc tính
<code>element.setAttribute(<i>attribute</i>, <i>value</i>)</code>	Thay đổi giá trị của thuộc tính
<code>element.style.property = new style</code>	Thay đổi giá trị thành phần của style

– Ví dụ 1:

```

function TingTong(){
    var txta = document.getElementById("a");
    var txtb = document.getElementById("b");
    var a = parseInt(txta.value);
    var b = parseInt(txtb.value);
    var tong=a+b;
    var ospan=document.getElementById("stong");
    ospan.innerHTML=tong;
}

```

– Ví dụ 2:

```

function DoiMau(){
    var op = document.getElementById("p1");
    var s=op.className;
    if(s=="c1"){
        op.className="c2";
    }
    else{
        op.className="c1";
    }
    // op.setAttribute("class","c2");
}
function DoiStyle(){
    var op = document.getElementById("p2");
    op.style.color="#0f0";
    op.style.fontWeight="Bold";
    op.style.fontSize="20px";
}

```

### 3.1.3. Thêm, xóa element

Lệnh	Mô tả
<code>document.createElement(<i>element</i>)</code>	Tạo một element
<code>document.removeChild(<i>element</i>)</code>	Xóa element
<code>document.appendChild(<i>element</i>)</code>	Thêm element con
<code>document.replaceChild(<i>element</i>)</code>	Thay thế element con
<code>document.write(<i>text</i>)</code>	Ghi nội dung text vào element

Ví dụ:

```

function InsertImage(id, imageurl, alternate){
    var node = document.getElementById(id);
    var newnode = document.createElement("img");
    newnode.setAttribute("src",imageurl);
    newnode.setAttribute("alt",alternate);
    node.appendChild(newnode);
}

```

### 3.1.4. Xử lý sự kiện của element

Cách 1: Gán thuộc tính sự kiện cho hàm xử lý

```

document.getElementById("main").onclick=function(){
    alert("Clicked on main");
};

```

Cách 2: Gán thuộc tính sự kiện cho lệnh xử lý trong thẻ mở

```

<div id="main" onClick="InsertImage('main','Image/logo.jpg','Logo')">
    MAIN
</div>

```

Cách 3: Dùng hàm addEventListener

```

document.getElementById("main").addEventListener("click",function(){
    alert("Clicked on main");
});

```

## 3.2. JS Browser DOM

### 3.2.1. Window

Đối tượng window được hỗ trợ trên tất cả trình duyệt, tham chiếu đến cửa sổ của trình duyệt.

- Đọc kích thước cửa sổ (không bao gồm thanh cuộn, công cụ)
  - window.innerHeight: chiều cao
  - window.innerWidth: chiều rộng

Ví dụ:

```

window.onresize=function(){
    var h,w;
    h = window.innerHeight;
    w = window.innerWidth;
    alert("Rộng: " + w + ", Cao: " + h);
};

```

- Mở cửa sổ mới: `window.open()`
- Đóng cửa sổ: `window.close()`
- Di chuyển cửa sổ: `window.moveTo()`
- Thay đổi kích thước cửa sổ: `window.resizeTo()`

### 3.2.2. Screen

`window.screen` trả về đối tượng chứa thông tin về màn hình của người sử dụng.

- Chiều rộng của màn hình: `screen.width`
- Chiều cao của màn hình: `screen.height`
- Chiều rộng của màn hình (trừ đi vùng chứa tính năng của hệ điều hành: Windows Taskbar, ...): `screen.availWidth`
- Chiều cao của màn hình (trừ đi vùng chứa tính năng của hệ điều hành: Windows Taskbar, ...): `screen.availHeight`

Ví dụ:

```

function scr(){
    var s="";
    s += "Width: " + screen.width;
    s += ", Height: " + screen.height;
    s += ", availWidth: " + screen.availWidth;
    s += ", availHeight: " + screen.availHeight;
    document.getElementById("status").innerHTML=s;
}

```

### 3.2.3. Location

- Đường dẫn URL của trang: `window.location.href`

```

alert(window.location.href);
window.location.href="http://google.com.vn";

```

- Trả về tên miền trỏ đến webserver của trang:  
`window.location.hostname`

- Trả về đường dẫn tương đối của trang trên webserver:  
`window.location.pathname`
- Trả về giao thức đang truy cập trang: `window.location.protocol`
- Tải trang mới bằng assign: `window.location.assign(URL)`

Ví dụ:

```
function newDoc() {
    window.location.assign("http://www.w3schools.com")
}
```

### 3.2.4. History

`window.history` là đối tượng chứa lịch sử truy cập của trình duyệt.

Nút Back: `window.history.back()`

Nút Forward: `window.history.forward()`

### 3.2.5. Popup alert

- Thông báo: `window.alert("Nội dung thông báo")`

Ví dụ:

```
window.alert("Hello");
```

- Hộp thoại cho người dùng chọn OK hoặc Cancel:  
`window.confirm("Thông báo")`

```
window.confirm("Are you sure?");
```

Hộp thoại cho người dùng nhập giá trị: `window.prompt("Thông báo", "Giá trị ban đầu")`

```
window.prompt("What's your name?");
```

### 3.2.6. Timing

- Đợi một khoảng thời gian sau đó thực hiện các lệnh trong hàm

```
<button onclick="setTimeout(myFunction, 3000)">Try it</button>
```

```
<script>
```

```
function myFunction() {
    alert('Hello');
}
```

```
}
```

```
</script>
```

– Ngừng trước khi thực hiện hàm trong setTimeout

```
<button onclick="myVar = setTimeout(myFunction, 3000)">Try it</button>
```

```
<button onclick="clearTimeout(myVar)">Stop it</button>
```

– Lặp lại các lệnh trong hàm sao một khoảng thời gian

```
var myVar = setInterval(myTimer, 1000);
```

```
function myTimer() {  
    var d = new Date();  
    document.getElementById("demo").innerHTML = d.toLocaleTimeString();  
}
```

## CÂU HỎI, BÀI TẬP

6.1. Thiết kế trang web thay đổi màu nền ngẫu nhiên cho đoạn văn bản mỗi khi click chuột lên nút.

6.2. Thiết kế trang web tự động thêm đường kẻ ngang (màu ngẫu nhiên) mỗi khi click chuột lên nút.

6.3. Thiết kế trang web hiển thị danh sách HSSV (Mã HSSV, Họ lót, Tên, Giới tính, Năm sinh). Khi click chuột lên nút “Thêm” thì chèn dòng mới ở cuối bảng (có các input cho người dùng nhập thông tin). Sau khi nhập thông tin, người dùng click chuột lên nút “Ghi” thì hiển thị các thông tin đã nhập ngay trong dòng vừa mới chèn (các input không còn).

## YÊU CẦU KIỂM TRA, ĐÁNH GIÁ KẾT QUẢ HỌC TẬP

Lập trình thay đổi nội dung, thuộc tính của element

Lập trình thêm, xóa element

# BÀI 7: TẠO GIAO DIỆN CHUYÊN NGHIỆP BẰNG JQUERY

Mã bài: 18.7

## Giới thiệu:

JQuery là thư viện Javascript, dễ học và sử dụng. Với JQuery, người thiết kế web có thể lập trình xử lý tương tác với trình duyệt dễ dàng hơn. Để học và sử dụng JQuery, chúng ta phải có kiến thức về HTML, CSS và Javascript.

## Mục tiêu:

- Biết công dụng của jQuery
- Thiết kế được giao diện trang web chuyên nghiệp với jQuery
- Cẩn thận, an toàn

## Nội dung chính:

### 1. Tổng quan về JQuery

#### 1.1. Giới thiệu

JQuery là thư viện Javascript nhỏ gọn “viết ít, làm nhiều”, giúp việc lập trình xử lý bằng Javascript trở nên đơn giản. Để lập trình cho những tác vụ phổ biến trên trang web bằng Javascript, đòi hỏi người thiết kế phải viết nhiều dòng lệnh. Với jQuery, công việc trở nên đơn giản chỉ với một hoặc vài dòng lệnh.

JQuery cũng đơn giản hóa rất nhiều công việc phức tạp từ Javascript, như các lệnh tải dữ liệu AJAX, thao tác trên HTML DOM.

Thư viện jQuery bao gồm các tính năng sau:

- Thao tác trên HTML DOM
- Thao tác trên CSS
- Phương thức xử lý sự kiện trên HTML
- AJAX
- Tiện ích

#### 1.2. Cài đặt

Tải jQuery mới nhất từ <http://jquery.com/download/> về thư mục Scripts trong site

#### 1.3. Sử dụng

Khai báo sử dụng thư viện jQuery bằng thẻ script

```
<script type="text/javascript" src="Scripts/jquery-x.y.z.js"></script>
```

Trong đó: x, y, z là phiên bản jQuery sử dụng

## 1.4. Các thành phần

**`$(selector).action()`**

- \$: dấu báo lệnh sử dụng thư viện jQuery
- selector: truy vấn lọc/tìm element
- action: hành động trên element

Ví dụ:

`$(this).hide()` - ẩn element hiện hành.

`$("p").hide()` - ẩn tất cả element <p>.

`$(".test").hide()` - ẩn tất cả element có thuộc tính class="test".

`$("#test").hide()` - ẩn tất cả element có thuộc tính id="test".

## 2. Chọn element

### 2.1. Cú pháp và cách chọn tương tự CSS

Tất cả element <p>: `$("p")`

Element có thuộc tính id là "test": `$("#test")`

Tất cả element có thuộc tính class là "test": `$(".test")`

Tham khảo thêm tại:

[http://www.w3schools.com/jquery/jquery\\_ref\\_selectors.asp](http://www.w3schools.com/jquery/jquery_ref_selectors.asp)

Ví dụ:

```
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});
```



## 2.2. Chọn theo mối quan hệ phân cấp

<u>:first</u>	<code>\$("p:first")</code>	The first <p> element
<u>:last</u>	<code>\$("p:last")</code>	The last <p> element
<u>:even</u>	<code>\$("tr:even")</code>	All even <tr> elements
<u>:odd</u>	<code>\$("tr:odd")</code>	All odd <tr> elements
<u>:first-child</u>	<code>\$("p:first-child")</code>	All <p> elements that are the first child of their parent
<u>:first-of-type</u>	<code>\$("p:first-of-type")</code>	All <p> elements that are the first <p> element of their parent
<u>:last-child</u>	<code>\$("p:last-child")</code>	All <p> elements that are the last child of their parent
<u>:last-of-type</u>	<code>\$("p:last-of-type")</code>	All <p> elements that are the last <p> element of their parent
<u>:nth-child(n)</u>	<code>\$("p:nth-child(2)")</code>	All <p> elements that are the 2nd child of their parent
<u>:nth-last-child(n)</u>	<code>\$("p:nth-last-child(2)")</code>	All <p> elements that are the 2nd child of their parent, counting from the last child
<u>:nth-of-type(n)</u>	<code>\$("p:nth-of-type(2)")</code>	All <p> elements that are the 2nd <p> element of their parent
<u>:nth-last-of-type(n)</u>	<code>\$("p:nth-last-of-type(2)")</code>	All <p> elements that are the 2nd <p> element of their parent, counting from the last child
<u>:only-child</u>	<code>\$("p:only-child")</code>	All <p> elements that are the only child of their parent
<u>:only-of-type</u>	<code>\$("p:only-of-type")</code>	All <p> elements that are the only child, of its type, of their parent
<u>parent &gt; child</u>	<code>\$("div &gt; p")</code>	All <p> elements that are a direct child of a <div> element
<u>parent descendant</u>	<code>\$("div p")</code>	All <p> elements that are descendants of a <div> element
<u>element + next</u>	<code>\$("div + p")</code>	The <p> element that are next to each <div> elements
<u>element ~ siblings</u>	<code>\$("div ~ p")</code>	All <p> elements that are siblings of a <div> element

## 2.3. Form selector

<a href="#">:input</a>	<code>\$(":input")</code>	All input elements
<a href="#">:text</a>	<code>\$(":text")</code>	All input elements with type="text"
<a href="#">:password</a>	<code>\$(":password")</code>	All input elements with type="password"
<a href="#">:radio</a>	<code>\$(":radio")</code>	All input elements with type="radio"
<a href="#">:checkbox</a>	<code>\$(":checkbox")</code>	All input elements with type="checkbox"
<a href="#">:submit</a>	<code>\$(":submit")</code>	All input elements with type="submit"
<a href="#">:reset</a>	<code>\$(":reset")</code>	All input elements with type="reset"
<a href="#">:button</a>	<code>\$(":button")</code>	All input elements with type="button"
<a href="#">:image</a>	<code>\$(":image")</code>	All input elements with type="image"
<a href="#">:file</a>	<code>\$(":file")</code>	All input elements with type="file"
<a href="#">:enabled</a>	<code>\$(":enabled")</code>	All enabled input elements
<a href="#">:disabled</a>	<code>\$(":disabled")</code>	All disabled input elements
<a href="#">:selected</a>	<code>\$(":selected")</code>	All selected input elements
<a href="#">:checked</a>	<code>\$(":checked")</code>	All checked input elements

## 2.4. Chọn theo thuộc tính

<a href="#">[attribute]</a>	<code>\$("[href]")</code>	All elements with a href attribute
<a href="#">[attribute=value]</a>	<code>\$("[href='default.htm']")</code>	All elements with a href attribute value equal to "default.htm"
<a href="#">[attribute!=value]</a>	<code>\$("[href!='default.htm']")</code>	All elements with a href attribute value not equal to "default.htm"
<a href="#">[attribute\$=value]</a>	<code>\$("[href\$.jpg]")</code>	All elements with a href attribute value ending with ".jpg"
<a href="#">[attribute =value]</a>	<code>\$("[title='Tomorrow']")</code>	All elements with a title attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen
<a href="#">[attribute^=value]</a>	<code>\$("[title^='Tom']")</code>	All elements with a title attribute value starting with "Tom"
<a href="#">[attribute~=value]</a>	<code>\$("[title~='hello']")</code>	All elements with a title attribute value containing the specific word "hello"
<a href="#">[attribute*=value]</a>	<code>\$("[title*='hello']")</code>	All elements with a title attribute value containing the word "hello"

## 3. Thay đổi nội dung, thuộc tính

### 3.1. Duyệt danh sách các element

Cách 1:

```

var divs = $("div");
var n=divs.length;
var i;
for (i=0; i<n; i++)
{
    var div = divs[i];
    $(div).html(i);
}

```

Cách 2:

```

$("div").each(function(index, element) {
    //index là chỉ số
    //element: là phần tử đang
    //duyệt trong danh sách
    $(element).html(index);
});

```

### 3.2. Tạo element mới

```

// Create element with HTML
var txt1 = "<p>Text.</p>";
// Create with jQuery
var txt2 = $("<p></p>").text("Text.");
// Create with DOM
var txt3 = document.createElement("p");
txt3.innerHTML = "Text.";

```

### 3.3. Đọc, thay đổi nội dung trong element

- Đọc nội dung text

```

$("div").text();

```

- Đọc nội dung html

```

$("div").html();

```

- Thay đổi nội dung text

```

$("div").text("<p>abc</p>");

```

- Thay đổi nội dung html

```

$("div").html("<p>abc</p>");

```

### 3.4. Đọc, thay đổi thuộc tính

- Đọc giá trị thuộc tính

```
$("#a").attr("href");
```

- Thay đổi giá trị thuộc tính

```
$("#a").attr("href","http://brtvc.edu.vn");
```

### 3.5. Chèn nội dung

- append: chèn nội dung vào cuối element
- prepend: chèn nội dung vào đầu element
- after: chèn nội dung vào sau element
- before: chèn nội dung vào trước element

Ví dụ:

```
// Create element with HTML
var txt1 = "<p>Text.</p>";
// Create with jQuery
var txt2 = $("<p></p>").text("Text.");
// Create with DOM
var txt3 = document.createElement("p");
txt3.innerHTML = "Text.";
// Append the new elements
$("#p").append(txt1, txt2, txt3);
```

### 3.6. Làm việc với CSS

- Thêm/xóa giá trị của thuộc tính class
  - o addClass: thêm một hoặc nhiều class
  - o removeClass: xóa một hoặc nhiều class
  - o toggleClass: lần lượt thêm/xóa class

Ví dụ:

```
$("#h1, h2, p").addClass("blue");
$("#div").addClass("important");
$("#div1").addClass("important blue");
$("#p").removeClass("blue");
$("#h1, h2, p").toggleClass("blue");
```

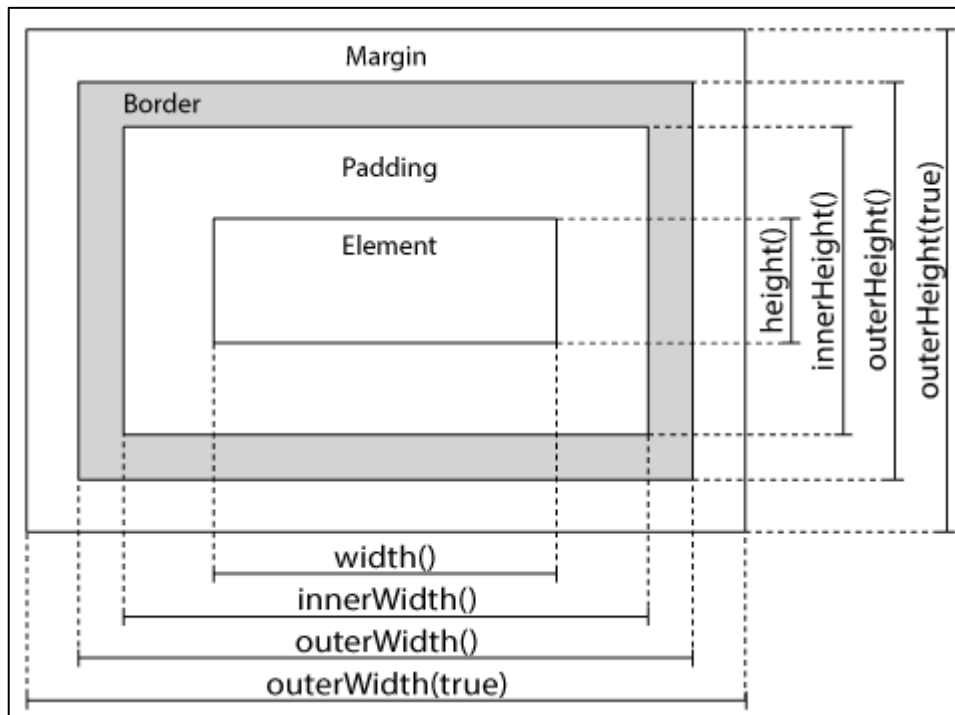
- Sử dụng phương thức css

```
$("#div").css("background-color", "#f00");
```

### 3.7. Thay đổi kích thước

Các phương thức thay đổi kích thước

- width()
- height()
- innerWidth()
- innerHeight()
- outerWidth()
- outerHeight()



Hình 8.1. jQuery dimensions

Ví dụ:

```
var h = $("#div1").height();  
alert(h);  
$("#div2").height(150);  
h=$("#div2").height();  
alert(h);
```

## 4. Xử lý sự kiện

### 4.1. Quy tắc chung

```
$("#<filter>").<event>(function(){  
    //code  
});
```

Ví dụ 1:

```
$(document).ready(function(){  
    alert("Hello ready from document");  
});
```

Ví dụ 2:

```
$(document).ready(function(){  
    $("li a").hover(function(){  
        $(this).addClass("hover");  
    });  
    $("li a").blur(function() {  
        $(this).removeClass("hover");  
    });  
});
```

### 4.2. Xử lý một số sự kiện thường gặp

– ready: khi document được tải

```
$(document).ready(function(){  
    alert("Document ready");  
});
```

– focus: khi control trong form nhận được con trỏ

```
$(document).ready(function(){  
    $(":text").focus(function(){  
        alert("Focus from textbox");  
    });  
});
```

– blur: khi control trong form không còn chiếm con trỏ

```
$(document).ready(function(){  
    $(":text").blur(function(){  
        alert("Textbox lost focus");  
    });  
});
```

- keypress: nhấn phím trong khi control đang nhận con trỏ

```
$(document).ready(function(){
    $("#text").keypress(function(){
        alert("Keypress in textbox");
    });
});
```

- change

```
$(document).ready(function(){
    $("#select").change(function(){
        alert("Select is changed selection");
    });
});
```

- click: khi click chuột

```
$(document).ready(function(){
    $("#button").click(function(){
        alert("Button is clicked");
    });
});
```

- mouseover: khi con trỏ chuột ở trên element

```
$("#div").mouseover(function(){
    $(this).css("background-color", "#f00");
});
```

- mouseout: khi con trỏ chuột di chuyển ra khỏi element

```
$("#div").mouseout(function() {
    $(this).css("background-color", "#fff");
});
```

### 4.3. Xử lý đối tượng Event

Trong mục 5.2, chúng ta đã xây dựng hàm xử lý cho sự kiện mà không dùng đến tham số sự kiện. Hàm xử lý sự kiện đầy đủ phải có tham số sự kiện.

- Vị trí tương đối của con trỏ chuột (`event.pageX`, `event.pageY`)

```

$(document).mousemove(function(event) {
    var pos="";
    pos += "X: " + event.pageX;
    pos += ", Y: " + event.pageY;
    $("#status").html(pos);
});

```

- Loại của sự kiện (event.type)

```

$(document).on("click dblclick mousemove keypress",function(event){
    var s = "Event: ";
    s += event.type;
    $("#status").html(s);
});

```

- Cho biết phím (hoặc nút chuột) nào được nhấn (event.which)

```

$(document).keypress(function(event) {
    var s="You pressed ASCII: ";
    s += event.which;
    $("#status").html(s);
});

```

## 5. Tạo hiệu ứng và hoạt ảnh

### 5.1. Ẩn, hiện

- Ẩn

```

$("#btnhide").click(function(){
    //Not param
    $("#p1").hide();
    //With speed param
    $("#p2").hide(1000);
    //With speed and easing param
    $("#p3").hide(2000, "linear");
    //With speed, easing and
    //callback function param
    $("#p4").hide(2000,"swing",function(){
        alert("Show fourth paragraph with callback");
    });
});

```

- Hiện: tương tự như ẩn, thay hide bằng show
- Ẩn chuyển sang hiện, hiện chuyển sang ẩn (toggle)



```

$("#btntoggle").click(function(){
    //Not param
    $("#p1").toggle();
    //With speed param
    $("#p2").toggle(1000);
    //With speed and easing param
    $("#p3").toggle(2000, "linear");
    //With speed, easing and
    //callback function param
    $("#p4").toggle(2000,"swing",function(){
        alert("Toggle fourth paragraph with callback");
    });
});

```

## 5.2. Fade-in, Fade-out

- Hiện lên (fadeIn)

```

$("#button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn(3000);
    $("#div3").fadeIn(2000,"swing");
    $("#div4").fadeIn(2000,"linear",function(){
    });
});

```

- Ẩn xuống (fadeOut): tương tự hiện lên, thay fadeIn bằng fadeOut
- Ẩn chuyển sang hiện, hiện chuyển sang ẩn (fadeToggle)

```

$("#btntoggle").click(function(){
    //Not param
    $("#p1").fadeToggle();
    //With speed param
    $("#p2").fadeToggle(1000);
    //With speed and easing param
    $("#p3").fadeToggle(2000, "linear");
    //With speed, easing and
    //callback function param
    $("#p4").fadeToggle(2000,"swing",function(){
        alert("Toggle fourth paragraph with callback");
    });
});

```

### 5.3. Sliding

```
$(".btn1").click(function(){
    $(".p").slideUp();
});
$(".btn2").click(function(){
    $(".p").slideDown();
});
$(".btn3").click(function(){
    $(".p").slideToggle();
});
```

Cách dùng các tham số: speed, easing, callback tương tự như mục 5.1 và 5.2.

### 5.4. Tạo hoạt hình

Cấu trúc tổng quát:

```
(selector).animate({styles}, speed, easing, callback);
```

Trong đó: speed, easing và callback là các tham số tùy chọn.

Ví dụ:

```
$("#box").animate({height: "300px"});
```

## CÂU HỎI, BÀI TẬP

7.1. Thiết kế trang web thay đổi màu nền ngẫu nhiên cho đoạn văn bản mỗi khi click chuột lên nút.

7.2. Thiết kế trang web tự động thêm đường kẻ ngang (màu ngẫu nhiên) mỗi khi click chuột lên nút.

7.3. Thiết kế trang web hiển thị danh sách HSSV (Mã HSSV, Họ lót, Tên, Giới tính, Năm sinh). Khi click chuột lên nút “Thêm” thì chèn dòng mới ở cuối bảng (có các input cho người dùng nhập thông tin). Sau khi nhập thông tin, người dùng click chuột lên nút “Ghi” thì hiển thị các thông tin đã nhập ngay trong dòng vừa mới chèn (các input không còn).

## YÊU CẦU KIỂM TRA, ĐÁNH GIÁ KẾT QUẢ HỌC TẬP

Dùng jQuery

Lập trình thay đổi nội dung, thuộc tính của element

Lập trình thêm, xóa element

# BÀI 8: PUBLISH WEBSITE

Mã bài: 18.8

## Giới thiệu:

Là quá trình đưa website đã được thiết kế xong trên máy tính cục bộ của bạn lên môi trường mạng thực sự, tương tác với người dùng thực sự.

## Mục tiêu:

- Làm việc với các bước xuất bản website
- Quảng bá website
- Bảo trì website

## Nội dung chính:

### 1. Đăng ký web hosting miễn phí

#### 1.1. Giới thiệu

Xuất bản website được chia thành nhiều bước:

Lựa chọn tên miền

Lựa chọn web server

Upload website

Cập nhật và chỉnh sửa website

Quảng bá website

#### 1.2. Lựa chọn tên miền



### 2. Publish web lên web hosting miễn phí

Bước 1: Đăng ký tài khoản Hostinger:

<http://www.hostinger.vn/order>

Bước 2: Điền đầy đủ thông tin rồi bấm vào tạo tài khoản

Bước 3: Vào Email để xác nhận tài khoản

- Truy cập vào đường dẫn trên, sau đó nhấp vào nút đăng kí trong bảng Free
- Ngay sau đó trong danh sách tài khoản sẽ xuất hiện tài khoản bạn vừa tạo

*Chú ý: thông thường phải mất ít nhất 3h để tên miền đó có thể truy cập.*

Tài khoản đang hoạt động					
Domain	Gói	Hết hạn vào	Hiện trạng	Ghi chú	Hành động
<a href="#">baihuongdan.zz.mu</a>	Free	-	Hoạt động	-	<input type="button" value="Chuyển"/>

Để truy cập vào trang quản lý tài khoản hosting gắn với tên miền thì bạn nhìn sang bên phải tên miền ở trên, chọn ” Chuyển”

Hiện trạng	Ghi chú	Hành động
Hoạt động	-	<input type="button" value="Chuyển"/>

Thông tin của website cũng như các thông số cài đặt

Thông tin tài khoản	
Domain:	<a href="#">baihuongdan.zz.mu</a>
Địa chỉ IP:	31.170.164.122
Tên đăng nhập:	u401757795
Mật khẩu:	••••••••
Dung lượng sử dụng:	0.02 / 2000 MB
Băng thông:	0.00 / 100000 MB
Home Root:	/home/u401757795
Tên server:	server4.hostinger.vn
Server IP:	31.170.164.19

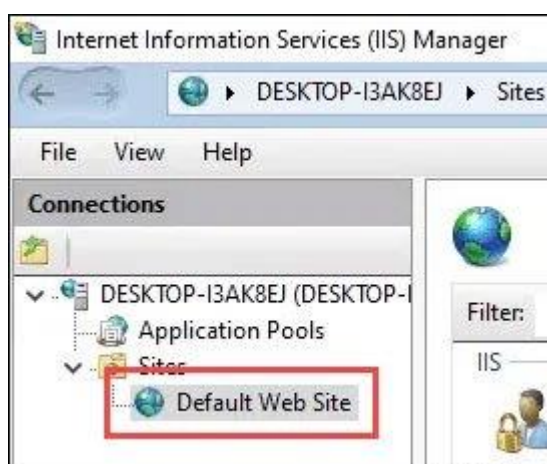
### 3. Publish lên web local IIS

IIS – Internet Information Service là một phần mở rộng của Microsoft, sử dụng trong Windows NT. IIS được dùng để triển khai các dịch vụ HTTP, HTTPS, FTP, FTTTPS, SMTP và NNTP. Trong bài này chúng ta đi tìm hiểu các bước cơ bản để thiết lập một web server với IIS.

Mặc định, IIS không được tích hợp sẵn, khi cần sử dụng, người dùng hoặc người quản trị phải cài đặt thêm thành phần này. IIS có thể được cài đặt trên Windows Server và các Windows Client như Windows 7, Windows 8, Windows 10...

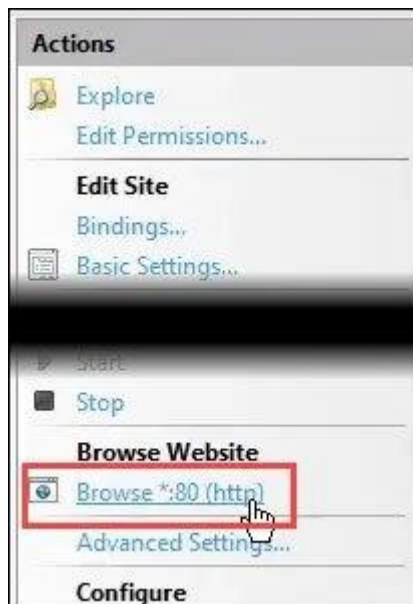
Các bước thực hiện:

- Bước 1: Mở Internet Information Services Manager



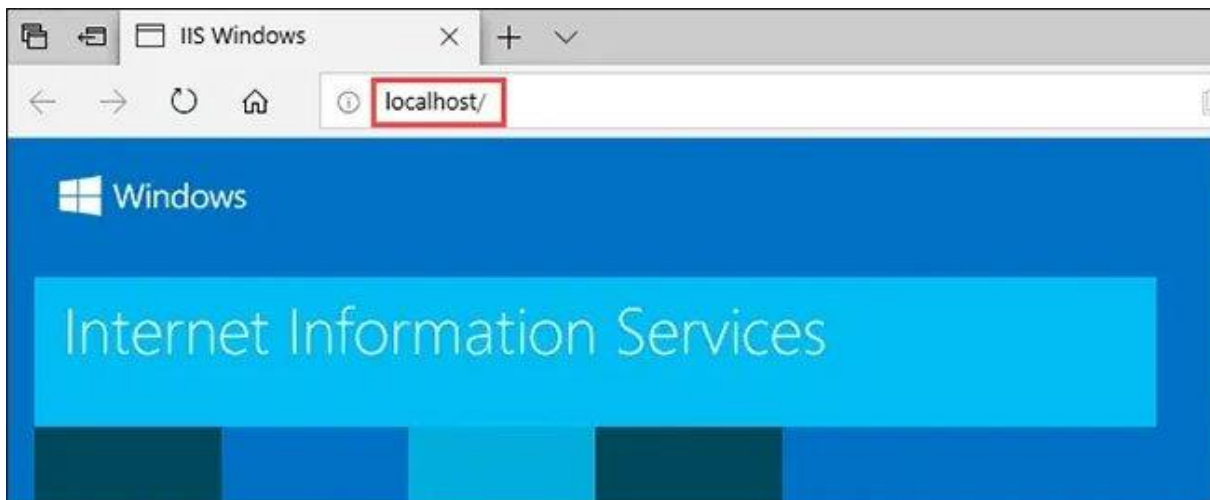
Hình 8.1: Mở rộng menu dạng cây cho đến khi bạn thấy Default Web Site

- Bước 2: Ở bên phải của **IIS Manager**, hãy xem phần **Browse Website**. Nhấp vào **Browse \*:80 (http)**. Trang web mặc định sẽ mở trong trình duyệt web mặc định.



Hình 8.2: Nhấp vào *Browse \*:80 (http)*

Sẽ thấy một trang web như sau. Lưu ý thanh địa chỉ có nội dung **localhost**. Đó là địa chỉ để nhập vào trang web mới.



Hình 8.3: Địa chỉ để nhập vào trang web mới

## CÂU HỎI, BÀI TẬP

- 8.1. Kiểm tra và cài đặt IIS trên máy tính của mình.
- 8.2. Publish website lên IIS.

## TÀI LIỆU THAM KHẢO

[1]. <http://w3schools.com>