

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**

BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN

MÔN: THIẾT KẾ WEB

ĐỀ TÀI: THIẾT KẾ MỘT TRANG WEB TIN TỨC

Giảng viên hướng dẫn : ThS. NGUYỄN LÊ MINH

Sinh viên thực hiện : BÙI MINH QUÝ

Lớp : CQ.62.CNTT

Khóa : 62

Thành phố Hồ Chí Minh, tháng 11 năm 2022

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**

BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN

MÔN: THIẾT KẾ WEB

ĐỀ TÀI: THIẾT KẾ MỘT TRANG WEB TIN TỨC

Giảng viên hướng dẫn : ThS. NGUYỄN LÊ MINH

Sinh viên thực hiện : BÙI MINH QUÝ

Lớp : CQ.62.CNTT

Khóa : 62

Thành phố Hồ Chí Minh, tháng 11 năm 2022

GIỚI THIỆU SƠ LƯỢC
BỘ MÔN: CÔNG NGHỆ THÔNG TIN



Mã sinh viên: 6251071025

Họ và Tên sinh viên: Phạm Văn Giang

Khóa: 62

Lớp: CNTT

1. Tên đề tài

THIẾT KẾ MỘT TRANG WEB NHÀ HÀNG Á ÂU

2. Mục đích, yêu cầu

a. Mục đích:

Hiện nay cách mạng công nghiệp 4.0 phát triển mạnh mẽ, con người có thể kết nối với nhau qua Internet, việc cập nhật tin tức cũng trở nên nhanh chóng hơn, nắm bắt được điều kiện đó, em đã thiết kế một trang web tin tức, giúp mọi người có thể đọc báo bất kì đâu với chiếc điện thoại thông minh của mình hoặc một laptop, máy tính bàn,... Qua đó có thể tiết kiệm thời gian tìm đọc các tờ báo in giấy mà còn nhanh chóng tiện lợi

b. Yêu cầu:

- Yêu cầu công nghệ

- Sử dụng ngôn ngữ lập trình HTML, CSS, Javascript.
- Sử dụng công cụ lập trình Visual Studio Code
- Sử dụng github để lưu trang web.

- Yêu cầu

- Chương trình có đầy đủ các giao diện như: phần menu chọn thể loại tin tức, phần xem tin tức các bài viết, phần liên hệ, phần kết nối với nhau qua mạng xã hội,...
- Trang web cho phép người dùng đọc, xem tin tức,...
- Giao diện: giống như một trang web đọc tin tức.

3. Nội dung và phạm vi đề tài

a. Nội dung:

- Tổng quan trang web.
- Tổng quan về các danh mục trong trang web.
- Phân tích và thiết kế trang web.
- Giao diện.
- Kết quả thu được.

b. Phạm vi:

- Lý thuyết về lập trình HTML, CSS, JavaScript
- Kết hợp sử dụng 3 ngôn ngữ trên để tạo trang web sinh động hấp dẫn.

4. Công nghệ, công cụ và ngôn ngữ lập trình

- Sử dụng ngôn ngữ lập trình HTML, CSS, Javascript.
- Sử dụng thành thạo cả ba ngôn ngữ trên để tạo trang web hoàn chỉnh.
- Sử dụng github để lưu trang web.

5. Các kết quả chính dự kiến sẽ đạt được và ứng dụng

Thiết kế thành công một trang web tin tức đầy đủ các danh mục đã nêu trên.

6. Giáo viên và cán bộ hướng dẫn

Họ tên: ThS. Nguyễn Lê Minh

Đơn vị công tác: Phân hiệu Trường Đại học Giao thông Vận tải.

LỜI CẢM ƠN

Lời đầu tiên em xin gửi lời cảm ơn chân thành đến thầy Minh giảng dạy trong **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông vận tải**.

Những người đã truyền dạy, đã trang bị cho em kho tàng kiến thức về bầu trời công nghệ thông tin rộng lớn.

Ở đây, em không chỉ học được kiến thức về sách vở mà em còn học được các bài học, kỹ năng sống trước khi tạm biệt mái trường đại học thân yêu này và tiến ra biển đời mênh mông rộng lớn. Đặc biệt, em xin gửi lời cảm ơn chân thành và sâu sắc đến thầy Nguyễn Lê Minh, người đã đồng hành cùng em trong suốt quá trình làm báo cáo, người đã bỏ thời gian quý báu, thậm chí là thời gian nghỉ ngơi để hướng dẫn, để định hướng đường đi nước bước cho em. Em thật chẳng biết dùng lời nào để diễn tả được công lao của thầy. **“CẢM ƠN THẦY. CẢM ƠN THẦY VỀ TẤT CẢ”**.

Trong quá trình học tập và tìm hiểu em đã nỗ lực rất nhiều với mong muốn hoàn thành bài tập lớn một cách tốt nhất, nhưng có thể em sẽ có những thiếu sót không thể tránh khỏi, và với những người chưa chứng chạc và trưởng thành như em thì sai lầm là không thể không mắc phải. Em mong thầy có thể thông cảm và cho em những ý kiến, đóng góp để em có thể hoàn thành Bài tập lớn của mình một cách trọn vẹn nhất.

Sau cùng, em xin kính chúc Thầy lời chúc sức khỏe, luôn hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày ... tháng ... năm ...

Giáo viên hướng dẫn

MỤC LỤC

GIỚI THIỆU SƠ LƯỢC	i
LỜI CẢM ƠN	iii
NHẬN XÉT CỦA GIÁO VIÊN	iv
DANH MỤC VIẾT TẮT	7
MỞ ĐẦU	8
1.1 Lý do chọn đề tài.	8
1.2 Hướng tiếp cận của đề tài.	8
1.3 Mục tiêu nghiên cứu.	8
1.4 Đối tượng và phạm vi nghiên cứu.	8
CHƯƠNG 1: LÝ THUYẾT VỀ FRONT-END	9
HTML	9
Tổng quan	9
Các thẻ HTML	14
Bài học HTML	17
CSS	22
Tổng quan	22
Các thuộc tính	23
JAVASCRIPT	34
Tổng quan JavaScript	34
Các Mệnh Đề Điều Khiển.....	38
Các đối tượng JavaScript.....	42
JavaScript BOM.....	43
JavaScript DOM	45
JavaScript Validation.....	47
CHƯƠNG 2: GIAO DIỆN TRANG WEB	48
GIAO DIỆN THỰC HÀNH BUỔI 1	48

Bài 1	48
Bài 2	48
Bài 3	49
Bài 4	49
Bài 5	50
Bài 6	50
GIAO DIỆN THỰC HÀNH BUỔI 2	51
Giao diện trang chủ bài viết.....	51
Thực đơn 1 :	54
Thực đơn 2 :	56
Thực đơn 3 :	57
Thực đơn 4 :	58
Giao diện trang liên hệ.....	59
CHƯƠNG 3: SOURCE CODE.....	60
SOURCE CODE HTML TRANG CHỦ :	60
SOURCE CODE HTML THỰC ĐƠN 1 :	68
SOURCE CODE HTML THỰC ĐƠN 2 :	73
SOURCE CODE HTML THỰC ĐƠN 3 :	77
SOURCE CODE HTML THỰC ĐƠN 4 :	82
SOURCE CODE HTML LIÊN HỆ :	87
SOURCE CODE CSS TRANG CHỦ :	88
SOURCE CODE CSS LIÊN HỆ :	100
SOURCE CODE JAVASCRIPT TRANG CHỦ :	104
SOURCE CODE JAVASCRIPT LIÊN HỆ :	105
SOURCE CODE CSS BOOSTRAP :	107
PHỤ LỤC	108

DANH MỤC VIẾT TẮT

STT	MÔ TẢ	Ý NGHĨA	GHI CHÚ
1	Front end	là một phần của một website	
2	HTML	Hyper Text Markup Language	
3	WEB	là một tập hợp các văn bản, hình ảnh, tệp tin tài liệu thích hợp với World Wide Web và được thực thi ở trình duyệt web.	
4	link	liên kết	
5	CSS	Cascading Style Sheets	
6	Comment	Chú thích	
7	FORM	Biểu mẫu	
8	XML	Extensible Markup Language	
9	SVG	là một định dạng ảnh sử dụng cấu trúc XML	
10	XUL	XML User Interface Language	

MỞ ĐẦU

1.1 Lý do chọn đề tài.

Chúng ta, thế hệ đang sống trong thời đại công nghiệp hóa, hiện đại hóa. Thời đại mà công nghệ thông tin đang ngày một phát triển như vũ bão. Nó có mặt hầu hết trong tất cả các lĩnh vực mà chúng ta đang sinh hoạt, làm việc và cũng như giải trí....Nó giúp đỡ cho chúng ta từ nông nghiệp, công nghiệp, kinh tế, giáo dục, y tế.

Chính vì thế tạo một trang web tin tức là một giải pháp tối ưu nhằm :

- Tiết kiệm thời gian tối đa để cập nhật tin tức trong và ngoài nước mỗi ngày.
- Tiết kiệm chi phí.
- Gắn kết mọi người dễ dàng hơn trên mạng xã hội.

1.2 Hướng tiếp cận của đề tài.

- Nắm vững kiến thức nền tảng của ngôn ngữ lập trình HTML, CSS, Javascript. biết vận dụng kết hợp sáng tạo cả ba ngôn ngữ để tạo một trang web sinh động hấp dẫn.
- Biết vận dụng kết hợp sáng tạo cả ba ngôn ngữ để tạo một trang web sinh động hấp dẫn.
- Tiến hành kiểm tra và chạy thử trang web.

1.3 Mục tiêu nghiên cứu.

- Phát triển thành công một trang web tin tức.
- Khiến sản phẩm trở nên có ích hơn cho xã hội.

1.4 Đối tượng và phạm vi nghiên cứu.

- Đối tượng: Trang chủ bài viết, bài viết 1, bài viết 2,..., liên hệ.
- Phạm vi trang web: Mong muốn sản phẩm có thể giúp mọi người cập nhật tin tức trong ngày, cả trong nước và ngoài nước.

CHƯƠNG 1: LÝ THUYẾT VỀ FRONT-END

HTML

Tổng quan

Khái niệm HTML

HTML là viết tắt của từ Hyper Text Markup Language, có nghĩa là ngôn ngữ đánh dấu siêu văn bản. HTML được sử dụng để định dạng và hiển thị văn bản trên trình duyệt tới người sử dụng. Nó là xương sống của một trang web.

Hyper Text: siêu văn bản có nghĩa là "văn bản trong văn bản". Một văn bản có chứa một liên kết (link) là một siêu văn bản. Mỗi lần bạn click vào một từ từ đó đưa bạn tới một trang web mới, đó là siêu văn bản.

Markup Language: ngôn ngữ đánh dấu là một ngôn ngữ lập trình được sử dụng để giúp văn bản dễ tương tác và linh động hơn. Nó có thể đặt văn bản vào trong các ảnh, bảng, liên kết...

Một tài liệu HTML chứa nhiều thẻ HTML và mỗi thẻ HTML chứa các nội dung khác nhau.

Ví dụ, một trang web chuẩn thì có chứa các thẻ HTML như sau:

```
1  <!DOCTYPE>
2  <html>
3    <head>
4      <!--
5        Thông tin cơ bản của trang web
6        và các liên kết đến css, javascript
7      -->
8    </head>
9    <body>
10     <!-- Nội dung trang web -->
11   </body>
12 </html>
```

Trong đó, thẻ <head> chứa thông tin được ẩn (không hiển thị lên trình duyệt) bao gồm thông tin cơ bản và các liên kết đến file css, javascript, ... Còn thẻ <body> chứa thông tin bạn muốn hiển thị lên trình duyệt.

○ Các đặc điểm của HTML

1. Đây là một ngôn ngữ rất dễ dàng và đơn giản.
2. Rất dễ dàng để trình bày hiệu quả với HTML vì nó có nhiều thẻ định dạng.

3. Đây là một ngôn ngữ đánh dấu vì vậy có thể sử dụng nó một cách linh hoạt để thiết kế trang web cùng với văn bản.

4. Có thể liên kết đến các trang web khác.

5. Là một nền tảng độc lập vì nó có thể hiển thị trên bất kỳ nền tảng nào khác như Windows, Linux và Mac.

6. Có thể thêm các hình ảnh, video, âm thanh vào các trang web khiến nó hấp dẫn và dễ tương tác hơn.

Các thẻ trong HTML

Thẻ HTML chứa ba phần chính: thẻ mở, nội dung và thẻ đóng. Nhưng một số thẻ HTML là các thẻ mở (không cần đóng thẻ).

Khi một trình duyệt đọc một tài liệu HTML, trình duyệt đọc nó từ trên xuống dưới và từ trái sang phải. Các thẻ HTML được sử dụng để tạo các tài liệu HTML và hiển thị thuộc tính của chúng. Mỗi thẻ HTML có các thuộc tính khác nhau.

Định dạng trong HTML

Định dạng HTML là một quá trình định dạng văn bản để có giao diện trông đẹp mắt hơn. Có nhiều thẻ định dạng trong HTML. Các thẻ này được sử dụng để làm cho văn bản đậm, nghiêng hoặc gạch chân. Có 12 lựa chọn định dạng văn bản có sẵn trong HTML.

Tiêu đề trong HTML

Tiêu đề trong HTML hoặc thẻ h trong HTML có thể được định nghĩa là tiêu đề hoặc phụ đề mà bạn muốn hiển thị trên trang web. Khi bạn đặt văn bản trong thẻ tiêu đề <h1>.....</h1>, nó được hiển thị trên trình duyệt có dạng đậm và kích thước của văn bản phụ thuộc vào số của thẻ tiêu đề như h1, h2, h3, h4, h5, h6.

Đoạn văn trong HTML

Đoạn văn trong HTML hoặc thẻ p trong HTML được sử dụng để xác định một đoạn trong một trang web. Khi sử dụng thẻ p trình duyệt tự thêm một dòng trống trước và sau một đoạn văn.

Liên kết trong HTML

Thẻ liên kết trong HTML hay thẻ anchor <a> trong HTML định nghĩa một liên kết được sử dụng để liên kết một trang với một trang khác.

Thuộc tính "href" là thuộc tính quan trọng nhất của thẻ HTML.

Ảnh trong HTML

Thẻ img trong HTML được sử dụng để hiển thị hình ảnh trên trang web. Thẻ img là một thẻ trống chỉ chứa các thuộc tính và không có thẻ đóng.

- Các thuộc tính của thẻ img

src và alt là các thuộc tính quan trọng của thẻ img trong HTML. Dưới đây là tất cả các thuộc tính của thẻ img trong HTML.

1. Thuộc tính src

Đây là một thuộc tính cần thiết mô tả nguồn hoặc đường dẫn của hình ảnh. Nó chỉ thị trình duyệt nơi để tìm kiếm hình ảnh trên máy chủ.

Vị trí của hình ảnh có thể nằm trên cùng một thư mục hoặc máy chủ khác.

2. Thuộc tính alt

Thuộc tính alt xác định một văn bản thay thế cho hình ảnh, nếu nó không thể được hiển thị. Giá trị của thuộc tính alt mô tả hình ảnh bằng văn bản. Thuộc tính alt sử dụng có lợi cho SEO.

3. Thuộc tính width

Nó là một thuộc tính tùy chọn được sử dụng để xác định chiều rộng để hiển thị hình ảnh. Bạn nên sử dụng CSS với thuộc tính width để điều chỉnh thay vì sử dụng thuộc tính width trong thẻ img.

4. Thuộc tính height

Nó là một thuộc tính tùy chọn được sử dụng để xác định chiều cao để hiển thị hình ảnh. Bạn nên sử dụng CSS với thuộc tính height để điều chỉnh thay vì sử dụng thuộc tính height trong thẻ img.

Table trong HTML

Thẻ table trong HTML được sử dụng để hiển thị dữ liệu ở dạng bảng (hàng * cột). Có thể có nhiều cột trong một hàng.

Các thẻ table trong HTML được sử dụng để quản lý việc bố cục trang web. Ví dụ: Phần tiêu đề, thanh điều hướng, nội dung trang, phần chân trang... Nhưng bạn nên sử dụng thẻ div thay vì table để quản lý bố cục của trang.

Các thẻ định nghĩa bảng trong HTML

Thẻ	Mô tả
<table>	Định nghĩa bảng.
<tr>	Định nghĩa một hàng trong một bảng.
<th>	Định nghĩa phần header (dòng đầu tiên) của bảng.
<td>	Định nghĩa 1 ô của bảng.
<caption>	Định nghĩa phụ đề bảng.
<colgroup>	Chỉ định một nhóm của một hoặc nhiều cột trong một bảng để định dạng.
<col>	Được sử dụng với thẻ <colgroup> để chỉ định thuộc tính cột cho mỗi cột.
<tbody>	Được sử dụng để nhóm nội dung body trong một bảng.
<thead>	Được sử dụng để nhóm nội dung header trong một bảng.
<tfooter>	Được sử dụng để nhóm nội dung footer trong một bảng.

Danh sách trong HTML

Danh sách trong HTML được sử dụng để hiển thị danh sách thông tin. Có ba loại khác nhau của danh sách HTML:

- Danh sách có thứ tự hay danh sách có đánh số (ol)
- Danh sách không có thứ tự (ul)
- Danh sách mô tả (dl)

Danh sách có thứ tự trong HTML

Danh sách có thứ tự trong HTML hay còn gọi là danh sách được đánh số trong HTML. Mặc định tất cả các mục của danh sách được đánh dấu bằng các con số tăng dần. Danh sách có thứ tự bắt đầu với thẻ và các mục danh sách bắt đầu bằng thẻ . Danh sách đánh số có 5 kiểu như sau:

- Kiểu số (1, 2, 3)
- Kiểu số La Mã (I, II, III)
- Kiểu số La Mã thường (i, ii, iii)
- Kiểu chữ hoa (A, B, C)
- Kiểu chữ thường (a, b, c)

Danh sách không có thứ tự trong HTML

Trong các danh sách không có thứ tự trong HTML, tất cả các mục danh sách được đánh dấu bằng các dấu chấm tròn. Danh sách không có thứ tự bắt đầu với thẻ và các mục danh sách bắt đầu bằng thẻ .

Có 4 kiểu ký tự bắt đầu của danh sách không có thứ tự trong HTML

- disc
- circle
- square
- none

Danh sách mô tả trong HTML

Danh sách mô tả trong HTML cũng là một kiểu danh sách được hỗ trợ bởi HTML và XHTML. Nó còn được gọi là danh sách định nghĩa, nơi các mục được liệt kê như một từ điển.

Danh sách định nghĩa là rất thích hợp khi bạn muốn trình bày bảng thuật ngữ, danh sách các điều khoản hoặc danh sách giá trị tên khác.

Danh sách định nghĩa HTML bao gồm ba thẻ sau:

- Thẻ <dl> là thẻ bắt đầu của danh sách mô tả.
- Thẻ <dt> định nghĩa một thuật ngữ.
- Thẻ <dd> định nghĩa mô tả.

Form trong HTML

Một HTML Form là một phần của trang web có chứa các điều khiển như các trường văn bản, các trường mật khẩu, các checkbox, các radio button, submit button, các menu...

HTML Form là phương tiện cho người dùng nhập dữ liệu được gửi đến máy chủ để xử lý. HTML Form được sử dụng khi bạn muốn thu thập một số dữ liệu của người truy cập trang web.

Ví dụ: Nếu người dùng muốn mua một số mặt hàng trên online, họ phải điền vào form các thông tin như địa chỉ giao hàng và chi tiết thẻ tín dụng / thẻ ghi nợ để có thể gửi hàng đến địa chỉ đó.

Thuộc tính style trong HTML

Thuộc tính style trong HTML được sử dụng để thay đổi hoặc thêm kiểu cách (style) cho các thành phần HTML đang tồn tại. Có các style mặc định cho các thành phần HTML như màu nền là trắng, văn bản là đen...

Thuộc tính style có thể áp dụng cho bất kỳ thẻ HTML nào. Để áp dụng style trên thẻ HTML, bạn cần có kiến thức cơ bản về các property của css như color, background-color, text-align, font-family, font-size...

Cú pháp của thuộc tính style style = "property:value"

Comment trong HTML

Thẻ comment được sử dụng để chèn chú thích vào tài liệu HTML, các dòng chú thích này sẽ không được hiển thị trên trình duyệt.

Bạn có thể sử dụng các nhận xét để giải thích các dòng code, nó giúp bạn đánh dấu lại vị trí và ý nghĩa của đoạn code bạn sửa lần cuối cùng khi trong tài liệu HTML chứa nhiều code.

Cú pháp: <!-- Đây là một comment -->

Các thẻ HTML

Thẻ Marquee trong HTML

Thẻ marquee trong HTML không phải là một thẻ HTML tiêu chuẩn, được sử dụng để di chuyển chữ hoặc ảnh theo chiều ngang hoặc dọc một cách tự động. Có nghĩa là bạn có thể làm cho chữ hoặc ảnh di chuyển lên trên, xuống dưới, sang trái, sang phải một cách tự động.

Thẻ Textarea trong HTML

Thẻ HTML <textarea> được sử dụng để định nghĩa một ô nhập văn bản nhiều dòng. Có thể nhập đoạn văn bản dài vô hạn và hiển thị với độ dài ô nhập là cố định.

Kích thước của HTML textarea được định nghĩa bởi các thuộc tính <cols> và <rows> hoặc có thể được định nghĩa thông qua tính chất height và width của CSS.

Doctype HTML

Trong các tài liệu HTML bạn đã từng thấy rằng có một khai báo <!DOCTYPE html> declaration trước thẻ <html>. Khai báo <!DOCTYPE html> không phải là một thẻ HTML. Nó được sử dụng để chỉ dẫn cho trình duyệt web về trang HTML.

Thực tế có rất nhiều kiểu của HTML như HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc.

Việc khai báo `<!DOCTYPE>` để tham chiếu đến Document Type Declaration (DTD) trong HTML 4.01; bởi vì HTML 4.01 dựa trên SGML. Nhưng HTML 5 không dựa trên ngôn ngữ SGML.

DTD định nghĩa các nguyên tắc của các ngôn ngữ markup vì thế các trình duyệt sẽ nhận ra và hiển thị được nội dung chính xác.

Thẻ title trong HTML

Thẻ HTML title được sử dụng để cung cấp nhan đề cho trang web của bạn. Nó là một trong những thành phần quan trọng trong SEO.

Thẻ HTML title phải được sử dụng bên trong thẻ `<head>`

Nhan đề của trang web được hiển thị trên title bar của trình duyệt (browser).

Thẻ meta trong HTML

Siêu dữ liệu (metadata) là dạng dữ liệu (thông tin) miêu tả về dữ liệu.

Thẻ HTML `<meta>` hay thẻ meta trong HTML được sử dụng để cung cấp metadata về tài liệu HTML. Metadata sẽ không được hiển thị trên trình duyệt, nhưng nó sẽ được bộ máy tìm kiếm phân tích để lấy thông tin cơ bản về trang web của bạn.

Thẻ HTML meta được sử dụng để định nghĩa các thông tin cơ bản như mô tả trang web, từ khóa, tác giả, chỉnh sửa cuối cùng và các metadata khác.

Metadata có thể được sử dụng bởi các trình duyệt (cách hiển thị nội dung hoặc tải lại trang), bộ máy tìm kiếm (từ khóa), và các dịch vụ web khác.

HTML5 giới thiệu một phương pháp để cho các nhà thiết kế web kiểm soát khung nhìn (khu vực hiển thị của một người dùng trên trang web), thông qua thẻ `<meta>`

Thẻ meta được định nghĩa trong thẻ `<head>`

Thẻ Div trong HTML

Thẻ HTML `<div>` hay thẻ div trong html được sử dụng để nhóm nhiều phần tử HTML lại với nhau..

Chúng ta biết rằng mỗi thẻ trong HTML đều có mục đích cụ thể chẳng hạn như thẻ `p` được sử dụng để chỉ định một đoạn văn bản, các thẻ `<h1>` đến `<h6>` được sử dụng để chỉ định tiêu đề. Nhưng thẻ `<div>` chỉ như một khối chứa nhiều thẻ khác bên trong và phân chia tài liệu HTML thành các

phần khác nhau. Vì thế thẻ div cũng được sử dụng để trình bày bố cục cho trang web.

Thẻ div thường được các nhà phát triển web sử dụng để nhóm các phần tử HTML với nhau và áp dụng các kiểu CSS cho nhiều phần tử cùng một lúc. Ví dụ: Nếu bạn gộp một tập hợp các phần tử đoạn văn vào một phần tử div để bạn có thể tận dụng các kiểu CSS và áp dụng kiểu chữ cho tất cả các đoạn văn cùng một lúc thay vì viết mã cho cùng một kiểu cho mỗi phần tử đoạn văn đó.

Thẻ pre trong HTML

Thẻ HTML <pre> hay thẻ pre trong HTML được sử dụng để để chỉ định các văn bản đã được định dạng trước. Văn bản nằm trong thẻ <pre>.....</pre> được hiển thị trong font có chiều rộng cố định. Thông thường thì font hiển thị là Courier. Các khoảng trắng và xuống dòng trong thẻ pre không bị mất đi giống như các thẻ khác khi hiển thị ra trình duyệt.

Thẻ pre trong HTML được sử dụng rộng rãi để hiển thị ví dụ về ngôn ngữ. Ví dụ: Java, C #, C, C ++ ... bởi vì nó hiển thị y nội dung y trang như khi bạn gõ.

Thẻ code trong HTML

Thẻ HTML <code> hay thẻ code trong HTML được sử dụng để biểu diễn mã máy tính. Nó là một thẻ cụm từ định nghĩa một đoạn mã máy tính. Mặc định, nó hiển thị font monospace mặc định của trình duyệt (được biết đến là font có độ dài cố định).

Thẻ Label trong HTML

Thẻ HTML <label> hay thẻ label trong HTML được sử dụng để định nghĩa nhãn cho phần tử <input>. Chẳng hạn như textbox, email, password, textarea...

Thẻ Input trong HTML

Thẻ HTML <input> hay thẻ input trong HTML được sử dụng để biểu diễn một trường input nơi mà người dùng có thể nhập dữ liệu.

Các phần tử <input> được sử dụng bên trong 1 phần tử <form> để khai báo các điều khiển đầu vào mà cho phép người dùng nhập dữ liệu.

Trường input có thể thay đổi theo nhiều cách, tùy thuộc vào giá trị của thuộc tính type.

Thẻ Button trong HTML

Thẻ <button> trong HTML được sử dụng để tạo các nút có thể click bên trong form trên trang web của bạn. Bạn có thể đặt nội dung như văn bản hoặc hình ảnh vào bên trong thẻ <button>.....</button>.

Bạn nên luôn chỉ định thuộc tính type cho một thẻ <button>. Vì các trình duyệt sử dụng kiểu mặc định khác nhau cho phần tử button.

Thẻ HTML Button có thể được sử dụng bên trong hoặc bên ngoài form.

Bên trong form, nó có tác dụng như một nút gửi hoặc reset form.

Bên ngoài form, nó có thể gọi hàm JavaScript để thực hiện một nhiệm vụ nào đó.

Thẻ HR trong HTML

Thẻ HTML <hr> hay thẻ hr trong HTML được sử dụng để vẽ đường kẻ ngang trong HTML. Nó được sử dụng để phân tách các chủ đề trong một trang web. Nó cũng được gọi là quy tắc đường kẻ ngang (Horizontal Rule) trong HTML.

Thẻ BR trong HTML

Thẻ HTML
 hay Thẻ br trong HTML được sử dụng để xuống dòng trong 1 đoạn văn. Nó là một thẻ trống, có nghĩa là không cần phải đóng thẻ.

Bài học HTML

Text Link trong HTML

Một trang web có thể chứa các đường link khác nhau để liên kết trực tiếp với các trang khác hay trong một phần của một trang. Những liên kết này được gọi là Hyperlinks (siêu liên kết).

Hyperlinks cho phép bạn truy cập vào các trang web khác nhau bằng cách nhấn chuột vào từ, cụm từ, và hình ảnh. Vì vậy, bạn có thể tạo các đường Hyperlinks sử dụng văn bản hoặc hình ảnh.

Image Link trong HTML

Dưới đây là một ví dụ về sử dụng hình ảnh như một siêu liên kết. Chúng ta chỉ cần sử dụng một hình ảnh bên trong một siêu liên kết tại vị trí của văn bản.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Ví dụ image link trong HTML</title>
5 </head>
6 <body>
7 <p>Click vào link sau:</p>
8 <a href="/html" target="_self">
9   
11 </body>
12 </html>

```

Email Link trong HTML

Email Link trong HTML được sử dụng để mở form soạn thảo email từ một trang web, nhưng điều này có thể gây ra các vấn đề về spam cho tài khoản email của bạn. Có một số người có thể chạy các chương trình để thu thập các email này và sau đó sử dụng chúng để spam theo các cách khác nhau.

Frame trong HTML

Frame trong HTML được sử dụng để phân chia cửa sổ trình duyệt của bạn thành các khu vực khác nhau mà mỗi khu vực tải một tài liệu HTML riêng. Một tập hợp các Frame trong cửa sổ trình duyệt được biết đến như là một Frameset. Cửa sổ được chia vào các Frame theo cách tương tự như các bảng: bên trong các hàng và cột.

Iframe trong HTML

Bạn có thể định nghĩa một Iframe trong HTML với thẻ <iframe>. Thẻ <iframe> không liên quan đến thẻ <frameset>, thay vào đó, nó có thể xuất hiện ở bất cứ đâu trong tài liệu HTML của bạn. Thẻ <iframe> xác định một khu vực trong trang mà tại đó trình duyệt có thể hiển thị một trang riêng biệt, bao gồm cả thanh cuộn và Border. Nói một cách đơn giản là thẻ này dùng để nhúng một trang khác vào trang hiện tại.

Thuộc tính src được sử dụng để xác định địa chỉ URL của trang mà chứa Iframe.

Block trong HTML

Tất cả các phần tử trong HTML có thể được chia thành hai loại: block elements - các phần tử cấp khối và inline elements - các phần tử nội tuyến.

Thiết lập màu nền (background) trong HTML

Màu nền (background) trong HTML của trang web mặc định là màu trắng. Bạn không thích màu này và muốn thiết lập một màu nền khác cho

trang của bạn. HTML cung cấp cho bạn hai cách sau để cài đặt nền cho trang web của bạn:

- Sử dụng màu nền.
- Sử dụng hình ảnh làm nền

Màu trong HTML

Màu trong HTML có thể được thiết lập bằng cách sử dụng thẻ `<body>` hoặc bạn có thể thiết lập màu cho từng thẻ đơn khác nhau bằng cách sử dụng thuộc tính `bgcolor`.

Thẻ `<body>` có các thuộc tính sau mà có thể được sử dụng để thiết lập các màu khác nhau:

- `bgcolor`: Thiết lập một màu cho nền của trang.
- `text`: Thiết lập một màu cho văn bản.
- `alink`: Thiết lập một màu cho các active link.
- `link`: Thiết lập một màu cho link văn bản.
- `vlink`: Thiết lập một màu cho visited link (các kết nối mà bạn đã nhấn chuột vào).

Font trong HTML

Font đóng vai trò rất quan trọng trong việc tạo ra một Website thân thiện với người sử dụng, giúp họ dễ theo dõi nội dung hơn.

Khi nói đến Font trong HTML, chúng ta liên tưởng đến các thuộc tính của văn bản như kiểu, kích cỡ và màu văn bản. Font phụ thuộc hoàn toàn vào máy tính và trình duyệt được sử dụng để xem trang web, nhưng bạn có thể sử dụng thẻ HTML `` để thiết lập kiểu văn bản, kích cỡ văn bản và màu cho văn bản trên web. Bạn cũng có thể sử dụng thẻ `<basefont>` để thiết lập cùng một kiểu kích cỡ và màu cho tất cả văn bản của tài liệu HTML.

Thẻ font có 3 thuộc tính là `size`, `color`, `face`. Để thay đổi bất kỳ thuộc tính font nào tại bất kỳ thời gian nào trong trang web của bạn, bạn sử dụng thẻ ``. Văn bản sẽ giữ nguyên cho đến khi bạn đóng thẻ với ``. Với thẻ ``, bạn có thể thay đổi một hoặc tất cả thuộc tính của nó.

Ghi chú: Hai thẻ font và basefont là thẻ cũ và chúng đang bị thay thế trong các phiên bản mới của HTML vì thế bạn không nên sử dụng chúng nhiều, bạn nên sử dụng CSS để thiết lập font.

Thẻ embed - Đa phương tiện trong HTML

Đôi khi bạn cần thêm video hoặc nhạc vào trong trang web của mình. Cách đơn giản nhất để thêm video hay âm thanh vào site của bạn là bao nó trong một thẻ HTML gọi là <embed>.

Bạn cũng có thể sử dụng một thẻ <noembed> đối với các trình duyệt không nhận thẻ <embed>. Ví dụ, bạn có thể sử dụng thẻ <embed> để hiển thị một đoạn phim và sử dụng thẻ <noembed> để hiển thị các ảnh IPG nếu trình duyệt không hỗ trợ thẻ <embed>.

Tạo chữ chạy (Marquee) trong HTML

Dòng chữ chạy xuất hiện rất nhiều trong đời sống, ví dụ như trong chương trình thời sự, trên các biển quảng cáo, ... Dòng chữ chạy trên web cũng như vậy. Một dòng chữ chạy trong HTML là một phần cuộn văn bản được hiển thị dọc theo chiều ngang hay dọc theo chiều dọc trang của bạn phụ thuộc vào các thiết lập. Nó được tạo ra bằng cách sử dụng thẻ <marquee>.

Ghi chú: Thẻ <marquee> này có thể không được hỗ trợ bởi khá nhiều trình duyệt vì thế bạn không nên tin tưởng vào thẻ này. Thay vào đó, bạn có thể sử dụng JavaScript và CSS để tạo ra các hiệu ứng như trên.

Style Sheet trong HTML

CSS mô tả cách mà tài liệu được hiển thị trên màn hình. W3C đã tích cực đẩy mạnh sự sử dụng của Style Sheet trên Web từ khi hiệp hội được thành lập năm 1994.

CSS cung cấp các cách dễ dàng và hiệu quả để thiết lập các thuộc tính khác nhau cho các thẻ HTML. Sử dụng CSS, bạn có thể cài đặt các thuộc tính cho một phần tử HTML đã cho. Mỗi thuộc tính có một tên và một giá trị, ngăn cách nhau bởi dấu hai chấm (:). Mỗi thuộc tính ngăn cách bởi dấu chấm phẩy (;).

Sử dụng Javascript trong HTML

Một script là một phần nhỏ trong chương trình mà có thể thêm khả năng tương tác cho trang web của bạn. Ví dụ, một script sẽ tạo ra một hộp cảnh

báo pop-up hoặc cung cấp một dropdown menu. Script này có thể được tạo ra bằng cách sử dụng JavaScript hoặc VBScript.

Bạn có thể viết các hàm bằng cách sử dụng bất kỳ ngôn ngữ script nào và sau đó bạn có thể kích hoạt những hàm này bằng cách sử dụng các thuộc tính HTML.

Ngày nay, chỉ có Javascript và các framework liên kết là được sử dụng bởi hầu hết các lập trình viên web, VBScript vẫn không được hỗ trợ bởi các trình duyệt lớn.

Layout trong HTML

Layout giúp tạo bố cục và làm cho trang web trông đẹp hơn. Việc thiết kế Layout là khá tốn kém thời gian và công sức.

Ngày nay, tất cả các site hiện đại đang sử dụng nền tảng CSS và Javascript để tạo các trang web động. Nhưng bạn có thể tạo ra một Layout đẹp bằng cách đơn giản là sử dụng các table hoặc các thẻ div, span trong HTML kết hợp với các thẻ định dạng. Trong phạm vi bài này, húng tôi sẽ cung cấp cho bạn một vài ví dụ về cách tạo một Layout đơn giản nhưng hiệu quả cho web của bạn bằng cách chỉ sử dụng HTML và các thuộc tính.

Sự kiện (Event) trong HTML

Khi người sử dụng vào trang web của bạn, họ làm những việc như nhấn chuột vào các đường link, đọc qua nội dung văn bản và hình ảnh.... Đó là những ví dụ mà chúng ta gọi là sự kiện (event) trong HTML và trong thuật ngữ Javascript và VBScript.

Chúng ta có thể viết các hàm xử lý sự kiện bằng cách sử dụng Javascript hoặc VBScript để định nghĩa các hành động tương ứng với sự kiện đó.

Các Font tham khảo trong HTML

Bạn sẽ có cái nhìn và cảm nhận (Look and Feel) khác nhau về một trang web tùy vào các font khác nhau ở các hệ điều hành khác nhau như Windows, Linux hoặc Mac iOS. Dưới đây chúng tôi liệt kê một danh sách các font khác nhau có trong các hệ điều hành khác nhau.

Thẻ là thẻ cũ từ phiên bản HTML 4.0 và bây giờ tất cả các font được thiết lập bằng cách sử dụng CSS. Dưới đây là cú pháp đơn giản để thiết lập font cho một trang web.

```
1 body { font-family: "new century schoolbook"; }  
2 hoặc  
3 <body style="font-family:new century schoolbook;">
```

Mã hóa ASCII trong HTML

Có $2^7 = 128$ ký tự in được mà có thể được biểu diễn bởi các mã 7 bit ASCII. Một số bộ ký tự khác không dành cho biểu diễn HTML và được sử dụng cho điều khiển phần cứng.

Bảng ASCII trong HTML

ASCII là chữ viết tắt của American Standard Code for Information Interchange. Có 128 mã ASCII tiêu chuẩn, mỗi mã có thể được biểu diễn bởi 7 chữ số nhị phân từ 0000000 đến 1111111.

Phần mở rộng ASCII thêm vào 128 ký tự mà đa dạng giữa máy tính, chương trình và các font.

CSS

Tổng quan

Định nghĩa

CSS là viết tắt của Cascading Style Sheets. Đây là một ngôn style sheet được sử dụng để mô tả giao diện và định dạng của một tài liệu viết bằng ngôn ngữ đánh dấu (markup). Nó cung cấp một tính năng bổ sung cho HTML. Nó thường được sử dụng với HTML để thay đổi phong cách của trang web và giao diện người dùng. Nó cũng có thể được sử dụng với bất kỳ loại tài liệu XML nào bao gồm cả XML đơn giản, SVG và XUL.

CSS được sử dụng cùng với HTML và JavaScript trong hầu hết các trang web để tạo giao diện người dùng cho các ứng dụng web và giao diện người dùng cho nhiều ứng dụng di động.

Công dụng của CSS

- **Giải quyết một vấn đề lớn**

Trước khi có CSS, các thẻ như phông chữ, màu sắc, kiểu nền, các sắp xếp phần tử, đường viền và kích thước phải được lặp lại trên mọi trang web. Đây là một quá trình rất dài tốn thời gian và công sức. Ví dụ: Nếu bạn đang phát triển một trang web lớn nơi phông chữ và thông tin màu được

thêm vào mỗi trang, nó sẽ trở thành một quá trình dài và tốn kém. CSS đã được tạo ra để giải quyết vấn đề này. Đó là một khuyến cáo của W3C.

- **Tiết kiệm rất nhiều thời gian**

Định nghĩa kiểu CSS được lưu trong các tệp CSS bên ngoài vì vậy có thể thay đổi toàn bộ trang web bằng cách thay đổi chỉ một tệp.

- **Cung cấp thêm các thuộc tính**

CSS cung cấp các thuộc tính chi tiết hơn HTML để định nghĩa giao diện của trang web.

Các thuộc tính

CSS Background

Chương này hướng dẫn bạn làm thế nào để thiết lập nền của các phần tử HTML khác nhau. Bạn có thể thiết lập các thuộc tính nền sau của một phần tử:

1. background-color thuộc tính được sử dụng để thiết lập màu nền của một phần tử.
2. background-image thuộc tính được sử dụng để thiết lập hình nền của một phần tử.
3. background-repeat thuộc tính được sử dụng để kiểm soát sự lặp lại của một hình ảnh ở chế độ nền.
4. background-position thuộc tính được sử dụng để kiểm soát vị trí của một hình ảnh ở chế độ nền.
5. background-attachment thuộc tính được sử dụng để kiểm soát việc di chuyển của một hình ảnh ở chế độ nền.
6. background thuộc tính được sử dụng như một cách viết tắt để chỉ định một số tính chất nền khác.

CSS Border

CSS Border hay thuộc tính border trong CSS được sử dụng để thiết lập đường viền trên một phần tử.

Các thuộc tính border trong CSS được sử dụng để chỉ định kiểu cách, màu sắc và kích thước của đường viền của một phần tử. Các thuộc tính CSS border được đưa ra dưới đây:

- border-style
- border-color
- border-width
- border-radius
- border

Font trong CSS

Để thiết lập font cho một trang HTML hoặc cho từng phần tử trong trang, bạn sử dụng các thuộc tính sau:

- Thuộc tính font-family: được sử dụng để thay đổi kiểu văn bản
- Thuộc tính font-style: được sử dụng để tạo một font chữ nghiêng hoặc chếch
- Thuộc tính font-variant: được sử dụng để tạo những chữ hoa nhỏ (small-cap)
- Thuộc tính font-weight: được sử dụng để tăng giảm độ đậm của font
- Thuộc tính font-size: được sử dụng để thiết lập kích cỡ font
- Thuộc tính font: sử dụng thuộc tính này nếu bạn muốn viết ít code hơn mà vẫn cài đặt được các thuộc tính liên quan tới font ở trên.

Định dạng văn bản (Text) trong CSS

Với một phần tử bất kỳ, bạn có thể thiết lập các thuộc tính text sau:

- Thuộc tính color được sử dụng để thiết lập màu cho văn bản.
- Thuộc tính direction được sử dụng để thiết lập hướng cho văn bản.
- Thuộc tính letter-spacing được sử dụng để thêm hoặc bớt khoảng cách giữa các chữ cái trong một từ.
- Thuộc tính word-spacing được sử dụng để tăng hoặc giảm khoảng cách giữa các từ trong một câu.
- Thuộc tính text-indent được sử dụng để tạo độ thụt của văn bản trong một đoạn văn.

- Thuộc tính text-align được sử dụng để căn chỉnh văn bản trong một tài liệu.
- Thuộc tính text-decoration được sử dụng để tạo cách dấu gạch ở chân, ở trên, ở giữa văn bản.
- Thuộc tính text-transform được sử dụng để chuyển văn bản thành chữ hoa hoặc chữ thường.
- Thuộc tính white-space được sử dụng để định dạng và điều khiển phần khoảng trắng của văn bản.
- Thuộc tính text-shadow được sử dụng để thiết lập hình bóng (shadow như trong word) xung quanh văn bản.

Image trong CSS

Hình ảnh đóng vai trò quan trọng trong bất kỳ Webpage nào. Hiện nay, khó có thể tìm thấy một Webpage hiện đại nào mà không có hình ảnh. Tuy nhiên, không phải cứ thêm quá nhiều hình ảnh (trừ các Webpage cho thời trang hay mẫu quần áo) là tốt. Việc trang trí hình ảnh đẹp và hợp lý sẽ làm trang của bạn tạo ấn tượng với người dùng.

Để làm cho hình ảnh đẹp hơn, bạn có thể sử dụng các thuộc tính trong CSS. CSS có các thuộc tính:

- Thuộc tính border: Thiết lập độ rộng của đường viền bao quanh hình ảnh.
- Thuộc tính height: Thiết lập chiều cao của hình ảnh.
- Thuộc tính width: Thiết lập độ rộng của hình ảnh.
- Thuộc tính –moz-opacity: thiết lập độ trong suốt của hình

Link trong CSS

Để thiết lập các trạng thái khác nhau cho link, bạn có thể sử dụng các thuộc tính:

- Trạng thái :link – biểu thị rằng trang web này trình duyệt chưa lưu (tức người sử dụng lần đầu tiên click vào đường dẫn này).
- Trạng thái :visited – biểu thị rằng đường dẫn tới trang web này đã được lưu bởi trình duyệt (tức là người sử dụng đã click vào đường dẫn này trước đó rồi).

- Trạng thái :hover – biểu thị rằng khi người sử dụng di chuyển chuột qua phần tử mà chứa link đó (tức là phần tử đó là một link khi người sử dụng di chuyển chuột qua phần tử đó).
- Trạng thái :active – biểu thị đường link là active khi người sử dụng click chuột vào.

Lưu ý: Trạng thái :hover PHẢI theo sau các trạng thái :link và :visited trong định nghĩa CSS thì mới có hoạt động. Nếu thiếu một trong hai trạng thái trên, thì các Style Rule bạn đã định nghĩa cho :hover sẽ không có giá trị.

Trạng thái :active PHẢI theo sau trạng thái :hover trong định nghĩa CSS.

Bảng trong CSS

Bảng là một công cụ hiển thị dữ liệu rõ ràng và hiệu quả. Mặc dù với các công nghệ mới (như Kendo UI), thì việc hiển thị dữ liệu bằng các thẻ div thường được sử dụng hơn. Tuy nhiên, với các ứng dụng Webpage nhỏ và với lượng dữ liệu hiển thị là không lớn thì sử dụng bảng là khá tiện lợi. Chương này sẽ trình bày cách sử dụng các thuộc tính khác nhau trong CSS để làm cho bảng của bạn đẹp hơn. Dưới đây là một số thuộc tính trong CSS:

- Thuộc tính border được sử dụng để thiết lập đường viền cho bảng.
- Thuộc tính border-collapse xác định rằng các đường viền của bảng nên được vào hợp thành một đường viền.
- Thuộc tính caption-side được sử dụng trong phần tử <caption>. Theo mặc định, chúng sẽ được hiển thị ở phần bên trên của bảng. Sử dụng thuộc tính này, bạn có thể xác định vị trí hiển thị của phần tử caption này.
- Thuộc tính empty-cells xác định xem có hiển thị đường viền không nếu một ô là trống.
- Thuộc tính table-layout cho phép bạn thiết lập layout cho bảng.

Căn lề trong CSS

Để xác định phần không gian xung quanh các phần tử, bạn sử dụng thuộc tính margin trong CSS. Thuộc tính margin thiết lập kích cỡ của phần khoảng trống BÊN NGOÀI đường viền. Với margin, bạn cũng có thể xác

định một giá trị âm cho thuộc tính này để tạo các phần nội dung gối lên nhau.

Giá trị của thuộc tính margin không được kế thừa bởi các phần tử con. Bạn hãy nhớ rằng các lề dọc lân cận (các lề trên và lề dưới) sẽ kết hợp thành một lề.

Để căn lề cho một phần tử, bạn có các thuộc tính sau:

- Thuộc tính margin: sử dụng thuộc tính này bạn có thể thiết lập tất cả style liên quan tới việc căn lề chỉ trong một khai báo CSS.
- Thuộc tính margin-bottom căn lề dưới của một phần tử.
- Thuộc tính margin-top căn lề trên của một phần tử.
- Thuộc tính margin-left căn lề trái của một phần tử.
- Thuộc tính margin-right căn lề phải của một phần tử.

Tất cả các thuộc tính liên quan tới Margin trên đều có thể nhận các giá trị sau:

- auto: Trình duyệt tự động ước lượng việc căn lề cho phần tử.
- length: Xác định độ rộng (đơn vị px, pt, cm, ...) của lề. Giá trị mặc định là 0.
- %: Xác định mối quan hệ giữa lề với độ rộng của phần tử chứa lề.
- inherit: Kế thừa thuộc tính này từ phần tử cha chứa phần tử có thuộc tính margin này.

List trong CSS

Trình bày hay liệt kê một danh sách có kèm theo các bullet hoặc dưới dạng đánh số giúp phần văn bản của bạn dễ đọc và tạo cảm giác thân thiện hơn. Để tạo style riêng cho phần list, bạn có thể sử dụng các thuộc tính trong CSS. Có 5 thuộc tính trong CSS:

- Thuộc tính list-style-type cho phép bạn kiểm soát hình dạng hay bề ngoài của Marker (giống như Bullet) chẳng hạn ở dạng hình tròn, hình vuông, hay dạng số, ...
- Thuộc tính list-style-position xác định rằng marker nên xuất hiện bên trong hay bên ngoài luồng hiển thị nội dung.

- Thuộc tính list-style-image: sử dụng một hình ảnh để làm marker thay cho các bullet hoặc số.
- Thuộc tính list-style: sử dụng thuộc tính này, bạn có thể khai báo một lần mà vẫn có thể xác định được tất cả các thuộc tính trên.
- Thuộc tính marker-offset xác định khoảng cách giữa một marker và phần text trong danh sách.

Thuộc tính padding trong CSS

Thuộc tính padding trong CSS cho phép bạn xác định khoảng không gian giữa nội dung hiển thị của một phần tử với đường viền (border) của nó.

Giá trị của thuộc tính padding có thể ở dạng chiều dài (đơn vị px, pt, cm, ...) , dạng %, hoặc từ khóa inherit. Nếu giá trị của thuộc tính là inherit, thì phần tử sẽ kế thừa giá trị của thuộc tính padding này từ phần tử cha của nó. Nếu giá trị là dạng %, thì đó là tỉ lệ phần trăm với hộp chứa phần tử này.

Với CSS, bạn không những có thể xác định padding cho toàn bộ phần không gian bao quanh nội dung, mà còn có thể xác định padding cho từng phần không gian này, ví dụ như phần trên, phần dưới, cạnh trái hoặc cạnh phải. Dưới đây là các thuộc tính trong CSS giúp bạn thực hiện việc này:

- Thuộc tính padding-bottom xác định phần padding bên dưới của một phần tử.
- Thuộc tính padding-top xác định phần padding bên trên của một phần tử.
- Thuộc tính padding-left xác định phần padding bên trái của một phần tử.
- Thuộc tính padding-right xác định phần padding bên phải của một phần tử.
- Thuộc tính padding giúp bạn xác định toàn bộ các thuộc tính trên, tức là có thể xác định toàn bộ phần không gian padding bao quanh một phần tử.

Con trỏ trong CSS

Thuộc tính cursor trong CSS giúp bạn xác định kiểu con trỏ hiển thị tới người dùng.

Một cách sử dụng phổ biến của thuộc tính này là sử dụng các hình ảnh làm con trỏ hiển thị trong các nút submit trên các form. Theo mặc định, khi con trỏ hover qua một link, thì con trỏ chuột thay đổi từ dạng con trỏ thành hình dạng một bàn tay.

Thuộc tính outline trong CSS

Nhắc đến Outline trong CSS có thể khiến một số bạn nhầm lẫn nó với Border, tuy nhiên giữa hai thuộc tính này có các điểm khác nhau khá lớn như sau:

- Một outline là một đường được vẽ ngay bên ngoài cạnh đường viền của các phần tử. Các Outline không chiếm giữ phần không gian, bởi vì chúng luôn luôn được đặt trên một hộp chứa phần tử.
- Không giống border, bạn không thể xác định độ rộng, màu và style cho từng cạnh khác nhau bởi vì outline là một đường.
- Không giống border, outline không thay đổi kích cỡ và vị trí của phần tử
- Outline không nhất thiết phải ở dạng hình chữ nhật.

Ghi chú: Các thuộc tính liên quan tới outline không được hỗ trợ bởi IE 6 và Netscape 7.

Liên quan tới outline, trong CSS bạn có thể xác định các thuộc tính sau:

- Thuộc tính *outline-width* xác định độ rộng của outline.
- Thuộc tính *outline-style* xác định line style cho outline.
- Thuộc tính *outline-color* xác định màu của outline.
- Thuộc tính *outline* để xác định tất cả thuộc tính của outline trên.

Thuộc tính *outline-width* xác định độ rộng của outline. Giá trị của thuộc tính này nên ở dạng độ dài (đơn vị pt, px, cm, ...) hoặc một trong các giá trị *thin*, *medium*, hoặc *thick*, khá giống với thuộc tính *border-width* trong CSS.

Kích cỡ trong CSS

Để thay đổi kích cỡ, bạn có thể sử dụng các thuộc tính trong CSS sau:

- Thuộc tính *height* xác định chiều cao của hộp.

- Thuộc tính width xác định độ rộng của hộp.
- Thuộc tính line-height xác định chiều cao của một dòng văn bản.
- Thuộc tính max-height thiết lập chiều cao tối đa của một hộp.
- Thuộc tính min-height thiết lập chiều cao tối thiểu của một hộp.
- Thuộc tính max-width thiết lập độ rộng tối thiểu của một hộp.
- Thuộc tính min-width thiết lập độ rộng tối thiểu của một hộp.

Scrollbar trong CSS

Đôi khi bạn gặp trường hợp mà phải hiển thị một lượng nội dung lớn hơn nhiều phần không gian được cấp phát cho nó. Bài này sẽ trình bày cách xử lý vấn đề này.

Scrollbar trong CSS, CSS cung cấp thuộc tính overflow để nói cho trình duyệt tạo ra một thanh cuộn. Thuộc tính này có thể nhận một trong các giá trị sau:

Giá trị	Mô tả
visible	Cho phép nội dung có thể tràn qua các đường viền của phần tử chứa nó
hidden	Phần nội dung sẽ bị cắt tại cạnh đường viền và sẽ không có thanh cuộn (scrollbar) nào được hiển thị
scroll	Kích cỡ của phần tử chứa không thay đổi, nhưng hiển thị (hay thêm vào) các thanh cuộn (scrollbar) để cho phép người dùng thực hiện thao tác scroll để xem hết phần nội dung
auto	Tương tự như giá trị scroll, nhưng thanh cuộn (scrollbar) sẽ chỉ được hiển thị nếu kích cỡ của nội dung lớn hơn không gian hộp hiển thị và bị tràn

Thuộc tính visibility trong CSS

Thuộc tính visibility trong CSS giúp bạn ẩn một phần tử nào đó. Sử dụng thuộc tính này cùng với JavaScript, bạn có thể tạo ra các menu phức tạp và các layout đa tính năng.

Chẳng hạn, bạn có thể sử dụng thuộc tính visibility để ẩn các thông báo lỗi và chỉ hiển thị các thông báo này nếu cần hiển thị tới người sử dụng, hoặc để ẩn các câu trả lời trong các câu hỏi cho tới khi người dùng đã trả lời xong.

Bảng sau liệt kê các giá trị mà thuộc tính visibility có thể nhận:

Giá trị	Mô tả
visible	Hiển thị nội dung tới người dùng
hidden	Ẩn nội dung
collapse	Chỉ sử dụng với các cột và hàng trong bảng

Căn chỉnh vị trí trong CSS

Căn chỉnh vị trí trong CSS giúp bạn thiết lập vị trí các phần tử trong tài liệu HTML. Bạn có thể đặt bất cứ phần tử HTML nào tại bất cứ đâu bạn muốn. Hoặc bạn có thể xác định vị trí tương đối của phần tử này trong mối quan hệ (lệ thuộc) vào vị trí tự nhiên trong trang hoặc vị trí tuyệt đối dựa trên phần tử cha.

Thuộc tính z-index trong CSS

CSS cung cấp cho bạn khả năng tạo các Layer đa dạng và có thể hiển thị chồng lên nhau, tức là phần tử này chồng lên phần tử khác. Bạn có thể thực hiện việc này thông qua thuộc tính z-index trong CSS.

Thuộc tính z-index được sử dụng đi kèm với thuộc tính position để tạo các Layer. Sử dụng thuộc tính này, bạn có thể làm cho một phần tử hiển thị ở lớp trên còn phần tử kia hiển thị ở lớp bên dưới.

Với thuộc tính z-index, bạn có thể tạo các Layout phức tạp. Phần tử nào có thuộc tính z-index lớn hơn thì sẽ hiển thị ở lớp trên và ngược lại.

Pseudo Class trong CSS

Các Class trong CSS có thể được sử dụng với Pseudo Class với cú pháp sau:

```
1 | selector.class:pseudo-class {property: value}
```

2

Bảng dưới liệt kê một số cách sử dụng của Pseudo Class:

Giá trị	Mô tả
:link	Sử dụng class này để thêm style đặc biệt cho một unvisited link.
:visited	Sử dụng class này để thêm style đặc biệt cho một visited link.
:hover	Sử dụng class này để thêm style đặc biệt cho một phần tử khi người dùng di chuyển chuột qua nó
:active	Sử dụng class này để thêm style đặc biệt cho một phần tử khi active
:focus	Sử dụng class này để thêm style đặc biệt cho một phần tử khi nó được focus
:first-child	Sử dụng class này để thêm style đặc biệt cho một phần tử mà là phần tử con đầu tiên của phần tử khác
:lang	Sử dụng class này để xác định một ngôn ngữ để sử dụng trong một phần tử cụ thể

Trong khi định nghĩa các Pseudo Class trong một khối <style>...</style>, bạn cần ghi nhớ:

- a: hover PHẢI theo sau a: link và a: visited trong phần định nghĩa CSS, nếu không nó sẽ không có hiệu quả.
- a: active PHẢI theo sau a: hover trong CSS, nếu không nó sẽ không có hiệu quả.
- Các tên được sử dụng cho Pseudo Class là không phân biệt kiểu chữ.
- Pseudo Class là khác với các Class khác trong CSS, tuy nhiên chúng có thể kết hợp với nhau.

Pseudo Element trong CSS

Pseudo Element (có thể gọi là các phần tử giả) được sử dụng để thêm các hiệu ứng đặc biệt cho một số Selector. Bạn không cần sử dụng JavaScript hoặc bất cứ ngôn ngữ Script nào để sử dụng các hiệu ứng này. Cú pháp đơn giản của Pseudo Element trong CSS như sau:

```
1 | selector:pseudo-element {property: value}
```

Dưới đây là một số cách sử dụng phổ biến của Pseudo Element:

Giá trị	Mô tả
:first-line	Sử dụng phần tử này để thêm Style đặc biệt cho dòng đầu tiên của đoạn văn trong một Selector
:first-letter	Sử dụng phần tử này để thêm Style đặc biệt cho chữ cái đầu tiên của đoạn văn trong một Selector
:before	Sử dụng phần tử này để chèn nội dung nào đó vào trước một phần tử
:after	Sử dụng phần tử này để chèn nội dung nào đó vào sau một phần tử

Các quy tắc trong CSS

- Quy tắc @import: sử dụng quy tắc này để import các Style Sheet vào trong Style Sheet hiện tại.
- Quy tắc @charset: xác định bộ ký tự mà Style Sheet sử dụng.
- Quy tắc @font-face: miêu tả bề mặt font sử dụng trong tài liệu.
- Quy tắc !important chỉ rằng đây là một Rule được định nghĩa bởi người dùng, và nên nhận quyền ưu tiên cao nhất và ghi đè các giá trị của cùng thuộc tính mà được xác định bởi người khác.

Ghi chú: Ngoài các quy tắc trên, còn có một số quy tắc khác sẽ được trình bày trong các chương tiếp theo.

Kiểu Media trong CSS

Một trong những đặc điểm quan trọng của Style Sheet là chúng xác định cách một Webpage hiển thị trên các thiết bị khác nhau: màn hình, trên paper, ...

Hiện tại, chúng ta có hai cách xác định thiết bị cho Style Sheet:

- Xác định thiết bị từ các quy tắc @media hoặc @import.
- Xác định thiết bị bên trong phần Document Language.

JAVASCRIPT

Tổng quan JavaScript

Định nghĩa JavaScript

Javascript là ngôn ngữ lập trình kịch bản hướng đối tượng được phát triển bởi Netscape Communications cho các ứng dụng client/server. Javascript là một ngôn ngữ lập trình thông dịch. JavaScript hiển thị các trang web theo cách tương tác và năng động góp phần tạo nên trang web động. Điều này cho phép các trang bắt các sự kiện, hiển thị các hiệu ứng đặc biệt, tự động tạo nội dung HTML, xác thực dữ liệu, tạo cookie, lấy thông tin trình duyệt của người dùng, v.v.

JavaScript được biết đến lần đầu tiên là LiveScript, nhưng Netscape đã đổi tên thành JavaScript, có thể là do sự hứng thú được tạo ra bởi Java. JavaScript xuất hiện lần đầu tiên trong Netscape 2.0 vào năm 1995 với tên LiveScript. Ngôn ngữ javascript đã được nhúng trong Netscape, Internet Explorer và các trình duyệt web khác.

Ngôn ngữ JavaScript được định nghĩa như sau:

- JavaScript là một ngôn ngữ lập trình hướng đối tượng, thông dịch.
- Được thiết kế để tạo các ứng dụng mạng tập trung.
- Bổ sung và tích hợp với Java.
- Bổ sung và tích hợp với HTML.
- Mã nguồn mở và đa nền tảng.

Cú pháp trong JavaScript

JavaScript có thể được thực hiện bởi sử dụng các lệnh JavaScript mà được đặt trong thẻ HTML <script>... </script> trong một trang web.

Thẻ `<script>` báo cho trình duyệt biết rằng tất cả văn bản ở giữa các thẻ này như là một script. Ví dụ một cú pháp trong JavaScript đơn giản như sau:

```
1 <script ...>
2   JavaScript code
3 </script>
```

Sau đây là 2 thuộc tính quan trọng của thẻ script:

- **Language** – Thuộc tính này xác định ngôn ngữ scripting nào bạn đang sử dụng. Giá trị của nó sẽ là javascript. Mặc dù các phiên bản gần đây của HTML (và XHTML) cho phép không cần sử dụng thuộc tính này nữa.
- **Type** – Thuộc tính này là những gì được đề nghị sử dụng và giá trị của nó nên được thiết lập là "text/javascript".

Ví dụ:

```
1 <script language="javascript" type="text/javascript">
2   JavaScript code
3 </script>
```

Nhúng JavaScript vào HTML

Các cách được ưu tiên nhất để nhúng JavaScript vào HTML như sau:

- Trong thẻ HTML `<head>...</head>`
- Trong thẻ HTML `<body>...</body>`
- Liên kết với file .js từ bên ngoài, đường dẫn của file .js được khai báo trong thẻ HTML `<head>...</head>`

Biến trong JavaScript

Một biến JavaScript đơn giản là một tên của vị trí lưu trữ. Có hai loại biến trong JavaScript: biến cục bộ (local) và biến toàn cầu (global).

Một số quy tắc trong khi khai báo biến JavaScript:

- Tên phải bắt đầu bằng ký tự (từ a đến z hoặc A đến Z), dấu gạch dưới (`_`) hoặc ký hiệu đô la (`$`).
- Sau chữ cái đầu tiên chúng ta có thể sử dụng các chữ số (0 đến 9), ví dụ `value1`.
- Các biến JavaScript phân biệt chữ hoa chữ thường, ví dụ `x` và `X` là các biến khác nhau.

Biến toàn cầu JavaScript

Một biến toàn cầu JavaScript được khai báo bên ngoài hàm hoặc được khai báo với đối tượng window. Nó có thể được truy cập từ bất kỳ hàm nào.

Ví dụ về biến toàn cầu trong JavaScript.

```
1  <!DOCTYPE html>
2
3  <html>
4  <head>
5  <meta charset="utf-8">
6  <title>Ví dụ biến toàn cầu trong JavaScript</title>
7  <script>
8      var value = 50; //biến toàn cầu
9      function a() {
10         alert(value);
11     }
12     function b() {
13         alert(value);
14     }
15 </script>
16 </head>
17 <body>
18     <input type="button" value="Button1" onclick="a();" />
19     <input type="button" value="Button2" onclick="b();" />
20 </body>
21 </html>
```

Kiểu dữ liệu trong JavaScript

JavaScript cung cấp các loại dữ liệu khác nhau để giữ các loại giá trị khác nhau. Có hai kiểu dữ liệu trong JavaScript.

- Kiểu dữ liệu nguyên thủy.
- Kiểu dữ liệu không nguyên thủy (tham chiếu).

JavaScript là một ngôn ngữ kiểu động, có nghĩa là bạn không cần phải chỉ định loại biến vì nó được động cơ JavaScript sử dụng động. Bạn cần sử dụng var ở đây để chỉ định kiểu dữ liệu. Nó có thể chứa bất kỳ loại giá trị nào như số, chuỗi, v.v. Ví dụ:

```
1  <script>
2      var a = 40; // lưu trữ một số
3      var b = "Luffy"; //lưu trữ một chuỗi
4  </script>
```

Toán tử trong JavaScript

Các toán tử trong JavaScript là các ký hiệu được sử dụng để thực hiện các phép toán trên các toán hạng. Ví dụ:

```
1 | var sum = 10 + 20;
```

2

Ở đây, + là toán tử số học và = là toán tử gán.

Các loại toán tử sau trong JavaScript:

- Toán tử số học.
- Toán tử so sánh (quan hệ).
- Toán tử bit.
- Toán tử logic.
- Toán tử gán.
- Các toán tử đặc biệt.

Hàm (function) trong JavaScript

- Hàm trong JavaScript là một chương trình con được thiết kế để thực hiện một công việc cụ thể.
- Hàm trong JavaScript được thực thi khi chúng được gọi từ một hàm khác, từ một sự kiện hoặc gọi từ chính nó.
- Hàm trong JavaScript luôn trả về một giá trị, nếu không định nghĩa giá trị cụ thể, hàm sẽ trả về undefined.
- Hàm trong JavaScript có thể có 0 hoặc nhiều tham số.

Sự kiện (Event) trong JavaScript

Sự kiện (Event) trong JavaScript là sự tương tác của JavaScript với HTML xảy ra khi người dùng hoặc trình duyệt thao tác với một trang web.

Các sự kiện là một phần của Document Object Model (DOM) Level 3 và mỗi phần tử HTML chứa một tập hợp các sự kiện mà có thể kích hoạt chương trình JavaScript.

Ví dụ về các sự kiện HTML:

- Khi người dùng click chuột
- Khi một trang web đã được tải
- Khi một hình ảnh đã được tải
- Khi di chuyển chuột qua một phần tử
- Khi một trường đầu vào được thay đổi
- Khi một biểu mẫu HTML được submit
- Khi người dùng nhấn một phím

Các Mệnh Đề Điều Khiển

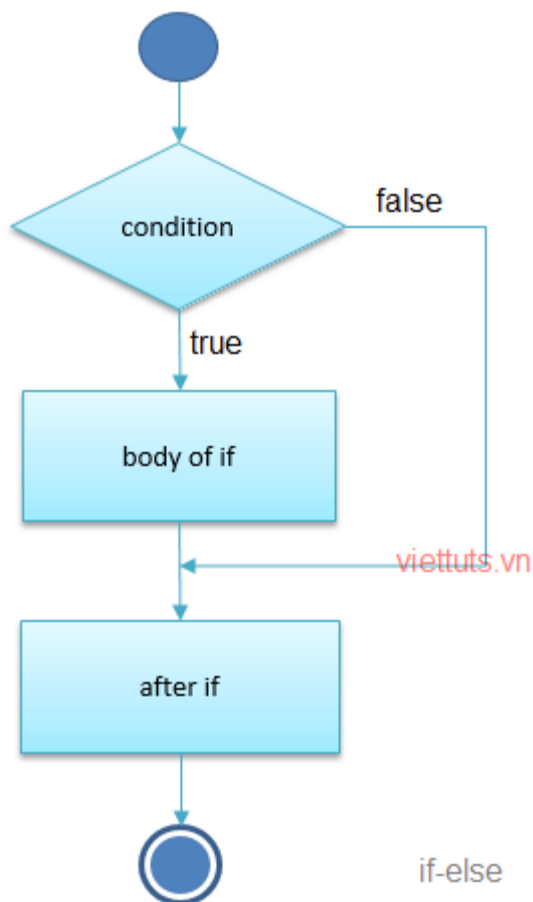
Mệnh đề if-else trong JavaScript

Mệnh đề if-else trong JavaScript được sử dụng để kiểm tra một biểu thức điều kiện nào đó có đúng hay không, nếu đúng thì thực thi những câu lệnh bên trong khối lệnh if và ngược lại nếu sai thì nó sẽ bỏ qua những câu lệnh đó. Có ba dạng của câu lệnh if trong JavaScript.

- Mệnh đề if.
- Mệnh đề if-else.
- Mệnh đề if-elseif-else.

Ngoài ra, còn có dạng mệnh đề if lồng nhau.

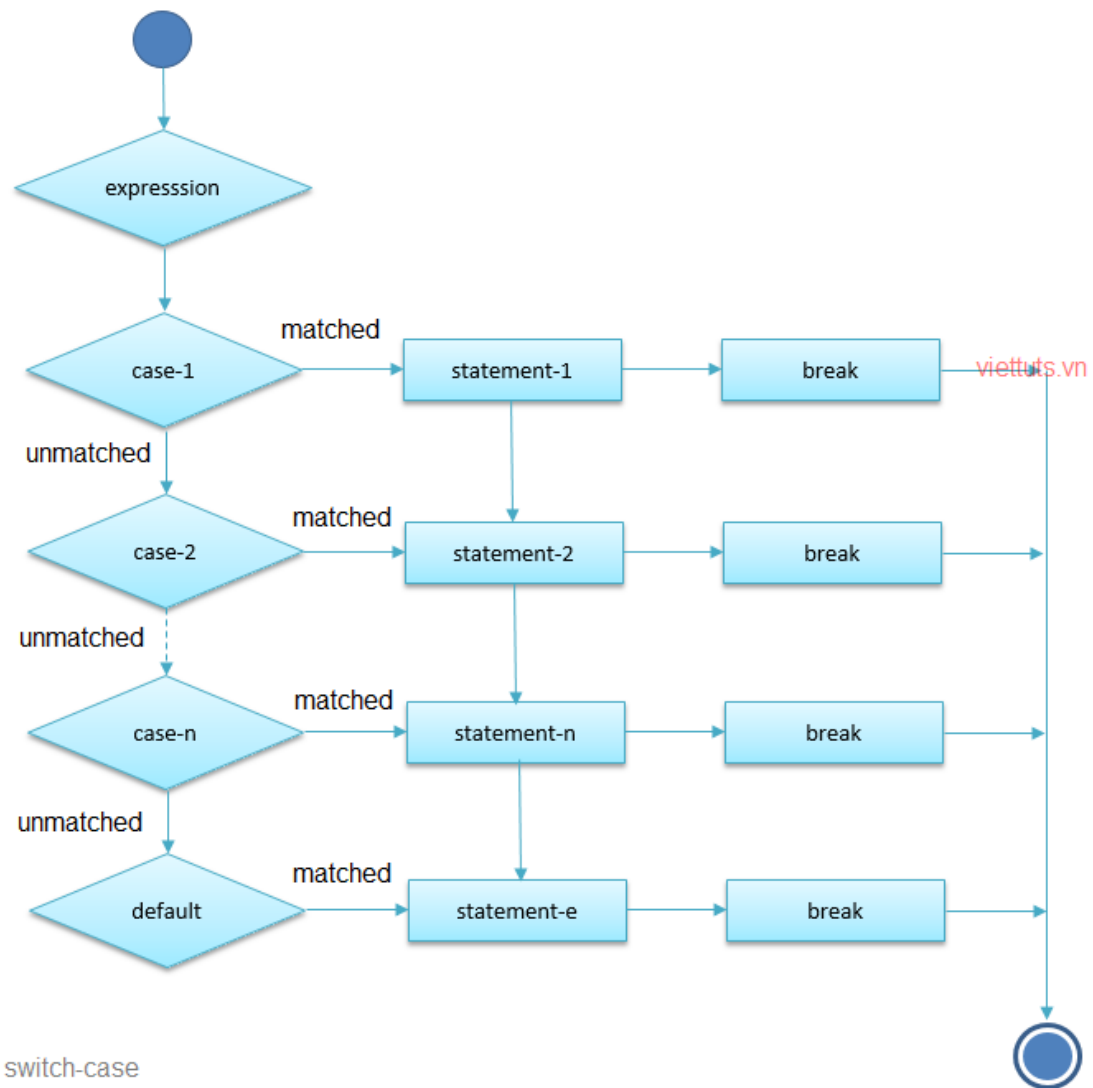
Lưu đồ thuật toán của mệnh đề if trong JavaScript:



Mệnh đề Switch-case trong JavaScript

Mệnh đề switch-case trong JavaScript được sử dụng để thực thi 1 hoặc nhiều khối lệnh từ nhiều điều kiện. Nó gần giống với mệnh đề if-else, nhưng nó thuận tiện hơn vì nó có thể được sử dụng với số, ký tự, v.v.

Lưu đồ thuật toán của mệnh đề switch-case trong JavaScript:

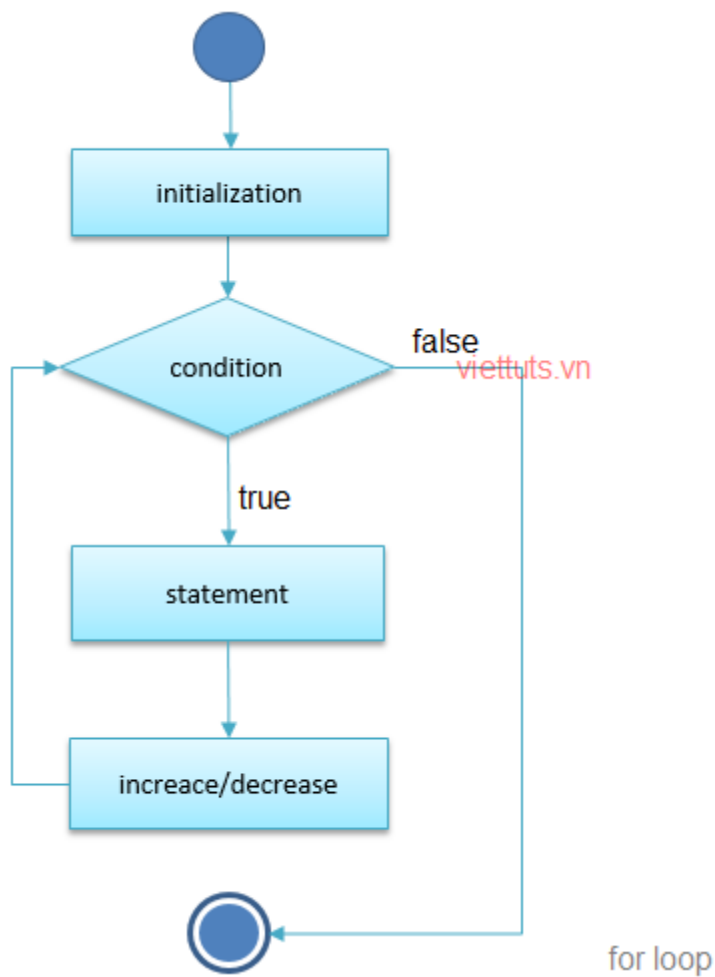


switch-case

Vòng lặp for trong JavaScript

Vòng lặp for trong JavaScript được sử dụng để lặp một phần của chương trình nhiều lần. Nếu số lần lặp là cố định thì vòng lặp for được khuyến khích sử dụng, còn nếu số lần lặp không cố định thì nên sử dụng vòng lặp while hoặc do while.

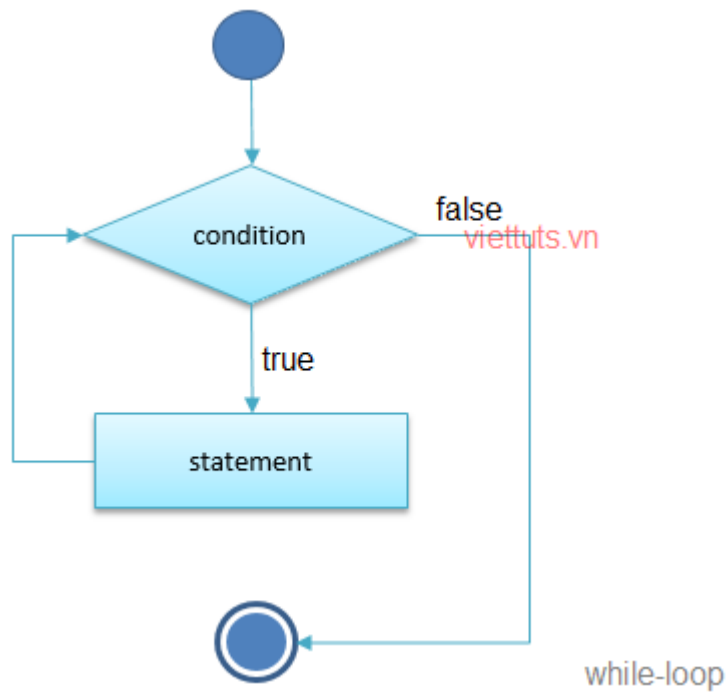
Lưu đồ thuật toán của vòng lặp for trong JavaScript:



Vòng lặp while trong JavaScript

Vòng lặp while trong JavaScript được sử dụng để lặp một phần của chương trình một vài lần. Nếu số lần lặp không được xác định trước thì vòng lặp while được khuyến khích sử dụng trong trường hợp này.

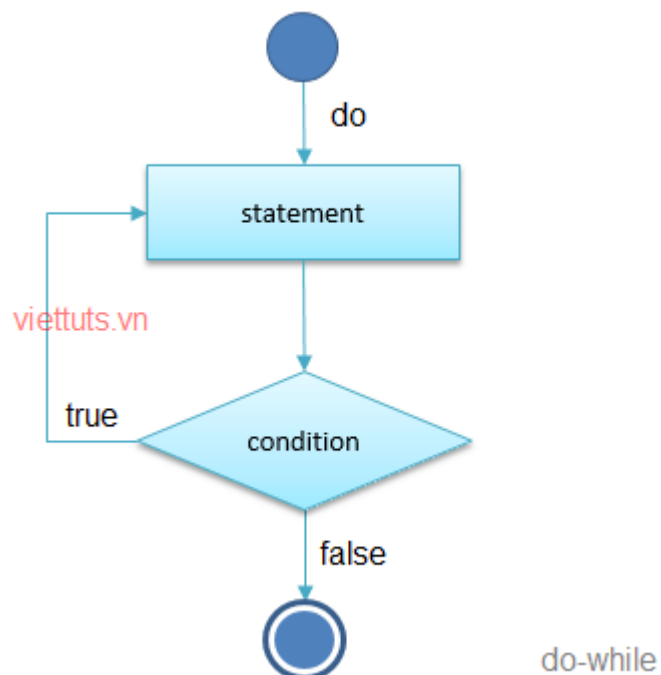
Lưu đồ thuật toán của vòng lặp while trong JavaScript:



Vòng lặp do-while trong JavaScript

Vòng lặp do-while trong JavaScript được sử dụng để lặp một phần của chương trình một vài lần. Tương tự như vòng lặp while, ngoại trừ do-while thực hiện lệnh ít nhất một lần ngay cả khi điều kiện là False - nghĩa là thực hiện trước sau đó mới kiểm tra điều kiện.

Lưu đồ thuật toán của vòng lặp do-while trong JavaScript:



Các đối tượng JavaScript

Đối tượng trong JavaScript

Một đối tượng JavaScript là một thực thể có trạng thái và hành vi (các thuộc tính và phương thức). Ví dụ: xe hơi, bút, xe đạp, ghế, kính, bàn phím, màn hình, v.v.

JavaScript là một ngôn ngữ dựa trên đối tượng. Mọi thứ đều là một đối tượng trong JavaScript.

JavaScript dựa trên mẫu chứ không dựa trên lớp. Ở đây, chúng ta không tạo lớp để lấy đối tượng. Nhưng, chúng ta trực tiếp tạo ra các đối tượng.

Mảng (Array) trong JavaScript

Mảng (Array) trong JavaScript là một đối tượng đại diện cho một tập hợp các loại phần tử tương tự.

Có 3 cách để tạo mảng trong JavaScript:

- Theo mảng chữ.
- Bằng cách tạo thể hiện của Array (sử dụng từ khóa new).
- Bằng cách sử dụng Constructor của đối tượng Array (sử dụng từ khóa new).

Chuỗi (String) trong JavaScript

Chuỗi (String) trong JavaScript là một đối tượng đại diện cho một chuỗi ký tự. Có 2 cách để tạo chuỗi trong JavaScript:

- Theo chuỗi chữ.
- Theo đối tượng String (sử dụng từ khóa new).

Đối tượng Date trong JavaScript

Đối tượng Date trong JavaScript có thể được sử dụng để lấy năm, tháng và ngày từ hệ thống. Bạn có thể hiển thị bộ hẹn giờ trên trang web nhờ sự trợ giúp của đối tượng Date trong JavaScript.

Bạn có thể sử dụng các hàm tạo Date khác nhau để tạo đối tượng Date. Nó cung cấp các phương thức để nhận và thiết lập ngày, tháng, năm, giờ, phút và giây.

Đối tượng Math trong JavaScript

Đối tượng Math trong JavaScript cung cấp một số hằng số và phương thức để thực hiện phép toán. Không giống như đối tượng Date, nó không có các constructor.

Đối tượng Number trong JavaScript

Đối tượng Number trong JavaScript cho phép bạn sử dụng để đại diện cho một giá trị số. Nó có thể là số nguyên hoặc dấu phẩy động. Đối tượng Number JavaScript tuân theo chuẩn IEEE để biểu diễn các số dấu phẩy động.

Với sự trợ giúp của hàm tạo Number(), bạn có thể tạo đối tượng số trong JavaScript. Ví dụ:

```
1 | var n=new Number(value);
```

Nếu không thể chuyển đổi giá trị thành số, nó trả về NaN (Không phải số) có thể được kiểm tra bằng phương thức isNaN().

Bạn cũng có thể gán trực tiếp một số cho một biến. Ví dụ:

```
1 | <script>
2 |     var x = 102;    // số nguyên
3 |     var y = 102.7;  // dấu phẩy động
4 |     var z = 13e4;   // số mũ e
5 |     var n = new Number(16); // tạo số nguyên từ đối tượng Number
6 | </script>
```

Boolean trong JavaScript

Đối tượng Boolean trong JavaScript là một đối tượng đại diện cho giá trị trong hai trạng thái: true hoặc false. Bạn có thể tạo đối tượng Boolean bằng hàm tạo Boolean() như được đưa ra dưới đây.

```
1 | Boolean b=new Boolean(value);
```

Giá trị mặc định của đối tượng boolean trong JavaScript là false.

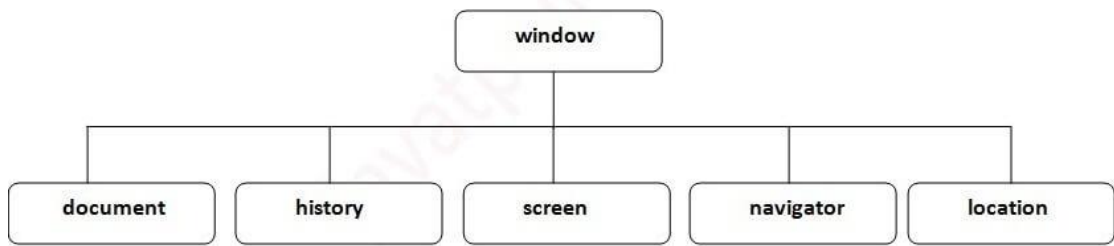
JavaScript BOM

Đối tượng Browser trong JavaScript

Đối tượng Browser trong JavaScript - BOM là viết tắt của Browser Object Model được sử dụng để tương tác với trình duyệt.

Đối tượng mặc định của browser là window có nghĩa là bạn có thể gọi tất cả các chức năng của window bằng cách chỉ định cửa sổ hoặc trực tiếp.

Bạn có thể sử dụng rất nhiều thuộc tính (các đối tượng khác) được định nghĩa bên dưới đối tượng window như document, history, screen, điều hướng, navigator, location, innerHeight, innerWidth



Lưu ý: Đối tượng document đại diện cho một tài liệu html. Nó tạo thành DOM (Mô hình đối tượng tài liệu).

Đối tượng window trong JavaScript

Đối tượng window đại diện cho một cửa sổ trong trình duyệt. Một đối tượng window được tạo ra tự động bởi trình duyệt.

Window là đối tượng của trình duyệt, nó không phải là đối tượng của javascript. Các đối tượng javascript là String, Array, Date, v.v.

Lưu ý: nếu tài liệu html chứa frame hoặc khung nội tuyến, trình duyệt sẽ tạo thêm các đối tượng window cho mỗi frame.

Đối tượng History trong JavaScript

Đối tượng history trong Javascript đại diện cho một loạt các URL được truy cập bởi người sử dụng. Bằng cách sử dụng đối tượng này, bạn có thể tải lại trang trước, trang sau hoặc bất kỳ trang cụ thể nào.

Đối tượng history là thuộc tính của đối tượng window, vì vậy nó có thể được truy cập bởi:

window.history hoặc history

Đối tượng Navigator trong JavaScript

Đối tượng navigator trong JavaScript được sử dụng để phát hiện trình duyệt. Nó có thể được sử dụng để lấy thông tin trình duyệt như tên ứng dụng, phiên bản ứng dụng, userAgent, v.v.

Đối tượng navigator là thuộc tính của đối tượng window, vì vậy nó có thể được truy cập bằng cách: window.navigator hoặc navigator

Đối tượng Screen trong JavaScript

Đối tượng screen trong JavaScript chứa thông tin của màn hình trình duyệt. Nó có thể được sử dụng để hiển thị chiều rộng màn hình, chiều cao, colorDepth, pixelDepth, v.v.

Đối tượng screen là thuộc tính của đối tượng window, vì vậy nó có thể được truy cập bằng cách: `window.screen` hoặc `screen`

Cookie trong JavaScript

Cookie là dữ liệu, được lưu trữ trong các tệp văn bản nhỏ, trên máy tính của bạn.

Khi một máy chủ web đã gửi một trang web đến một trình duyệt, kết nối sẽ bị tắt và máy chủ sẽ quên mọi thứ về người dùng.

Cookie được phát minh để giải quyết vấn đề "cách ghi nhớ thông tin về người dùng":

- Khi người dùng truy cập trang web, tên của anh ấy có thể được lưu trữ trong cookie.
- Lần tới khi người dùng truy cập trang, cookie "ghi nhớ" tên của anh ấy.

Cookie được lưu trong các cặp tên-giá trị như:

```
username = David Bishop
```

Khi trình duyệt yêu cầu một trang web từ máy chủ, các cookie thuộc trang được thêm vào yêu cầu. Bằng cách này, máy chủ nhận được dữ liệu cần thiết để "ghi nhớ" thông tin về người dùng.

Không có ví dụ nào dưới đây sẽ hoạt động nếu trình duyệt của bạn tắt hỗ trợ cookie.

JavaScript DOM

Đối tượng Document trong JavaScript

Đối tượng document trong JavaScript đại diện cho toàn bộ tài liệu HTML.

Khi tài liệu html được tải trong trình duyệt, nó sẽ trở thành một đối tượng document. Nó là phần tử gốc đại diện cho tài liệu html. Nó có các thuộc

tính và phương thức. Nhờ sự giúp đỡ của đối tượng document, chúng tôi có thể thêm nội dung động vào trang web.

Đối tượng document là thuộc tính của đối tượng window, vì vậy nó có thể được truy cập bằng cách:

window.document hoặc document

"Document Object Model (DOM) - Mô hình đối tượng document (DOM) là giao diện nền tảng và trung lập ngôn ngữ cho phép các chương trình và tập lệnh tự động truy cập và cập nhật nội dung, cấu trúc và css của tài liệu". Nguồn : W3C.

Document.getElementById() trong JavaScript

Phương thức document.getElementById() trả về phần tử của id được chỉ định.

Trong trang trước, chúng ta đã sử dụng document.form1.name.value để lấy giá trị của một textfield. Thay vào đó, chúng ta có thể sử dụng phương thức document.getElementById() để lấy giá trị của văn bản đầu vào. Nhưng chúng ta cần định nghĩa id cho trường đầu vào.

Ví dụ về phương thức document.getElementById() để in khối lập phương của một số đã cho.

```
1 <script type="text/javascript">
2     function getcube() {
3         var number = document.getElementById("number").value;
4         alert(number * number * number);
5     }
6 </script>
7 <form>
8     Enter No:<input type="text" id="number" name="number" />
9     <br> <input type="button" value="cube" onclick="getcube()" />
10 </form>
```

Document.getElementsByName() trong JavaScript

Phương thức document.getElementsByName() trong JavaScript trả về tất cả các phần tử của name được chỉ định.

Cú pháp của phương thức getElementsByName() được đưa ra dưới đây:

document.getElementsByName("name") . Ở đây, name là bắt buộc.

Document.getElementsByTagName() trong JavaScript

Phương thức document.getElementsByTagName() trong JavaScript trả về tất cả phần tử của tên thẻ được chỉ định.

Cú pháp của phương thức `getElementsByTagName()` được đưa ra dưới đây:

```
document.getElementsByTagName("tagName")
```

Trong đó, `tagName` là bắt buộc, ví dụ như thẻ `p`, `div`, `table`, `a`, `ul`, `ol`, `li`, ...

Thuộc tính innerHTML trong JavaScript

Thuộc tính `innerHTML` trong JavaScript có thể được sử dụng để viết html động trên trang web html.

Nó được sử dụng chủ yếu trong các trang web để tạo ra html động như form đăng ký, form nhận xét, liên kết, v.v.

Thuộc tính innerText trong JavaScript

Thuộc tính `innerText` trong JavaScript có thể được sử dụng để viết các văn bản động trên các trang web html. Ở đây, văn bản sẽ không được hiểu là văn bản html nhưng là văn bản bình thường.

Nó được sử dụng chủ yếu trong các trang web để tạo ra các nội dung động như viết thông báo xác nhận, độ mạnh mật khẩu, v.v.

JavaScript Validation

Validate Form bằng JavaScript

Điều quan trọng là cần phải validate form do người dùng gửi vì nó có thể có giá trị không phù hợp.

Validate HTML form có thể được thực hiện bởi JavaScript.

JavaScript cung cấp cho bạn cơ sở validate form hợp lệ ở phía máy khách để xử lý sẽ nhanh hơn xác nhận phía máy chủ. Vì vậy, hầu hết các nhà phát triển web thích validate form bằng JavaScript.

Thông qua JavaScript, chúng tôi có thể xác nhận tên, mật khẩu, email, ngày tháng, số điện thoại di động, vv.

Validate Email bằng JavaScript

Chúng ta có thể xác thực email bằng JavaScript.

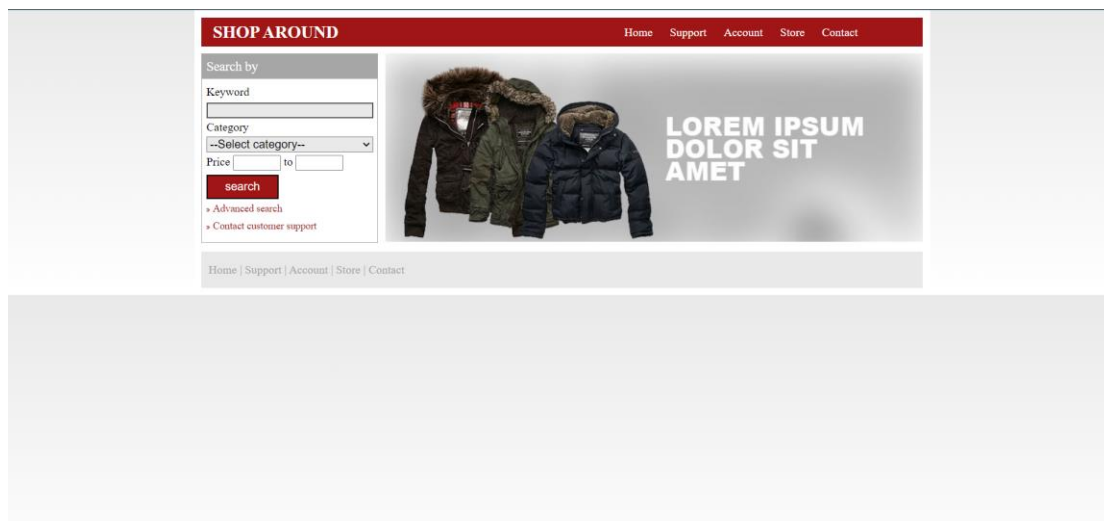
Có nhiều tiêu chí cần phải tuân theo để xác thực email như:

- email phải chứa ký tự '@' và '.'
- Phải có ít nhất một ký tự trước và sau '@'
- Phải có ít nhất hai ký tự sau '.' (dấu chấm).

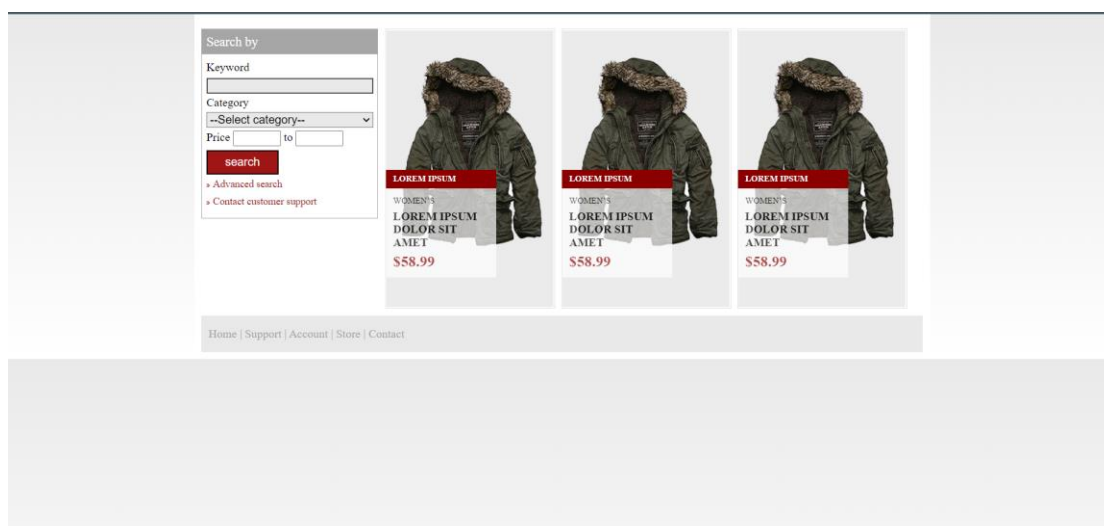
CHƯƠNG 2: GIAO DIỆN TRANG WEB

GIAO DIỆN THỰC HÀNH BUỔI 1

Bài 1



Bài 2



Bài 3

Categories

Category 1

Category 2

Category 3

Category 4

Category 5

Category 6

Category 7

Category 8

Category 9

Search by

Keyword

Category

--Select category--


Price to

search

[Advanced search](#)

[Contact customer support](#)


Home | [Support](#) | [Account](#) | [Store](#) | [Contact](#)




LOREM IPSUM
DOLOR SIT
AMET

Bài 4


Register Login Checkout Delivery MyAccount **Shopping Cart: NoItem**




Product-Name



Product-Name



Product-Name



Product-Name

Comment

Email

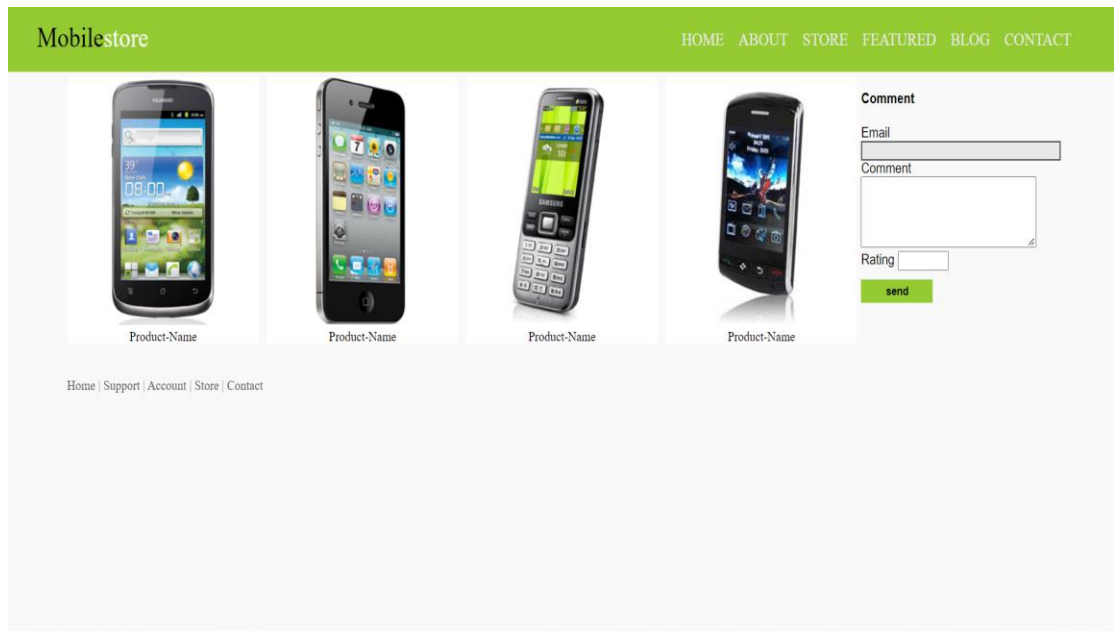
Comment

Rating

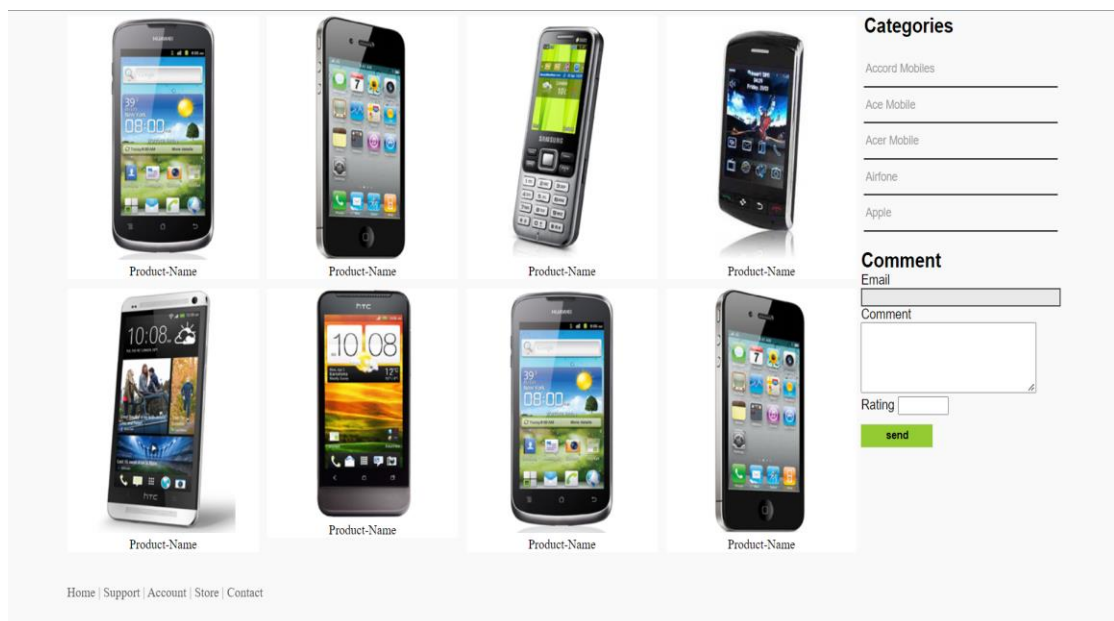
send

Home | [Support](#) | [Account](#) | [Store](#) | [Contact](#)

Bài 5



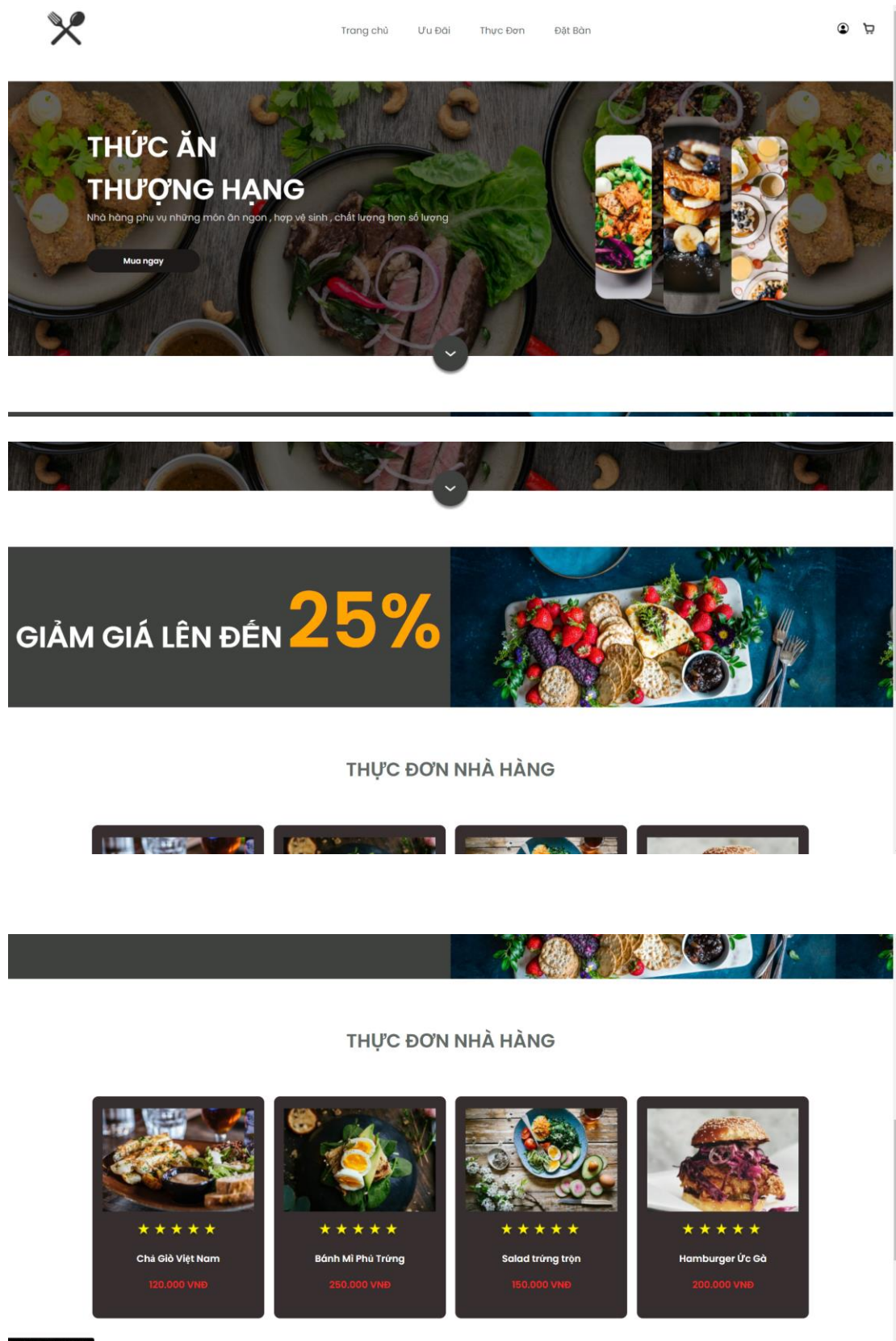
Bài 6

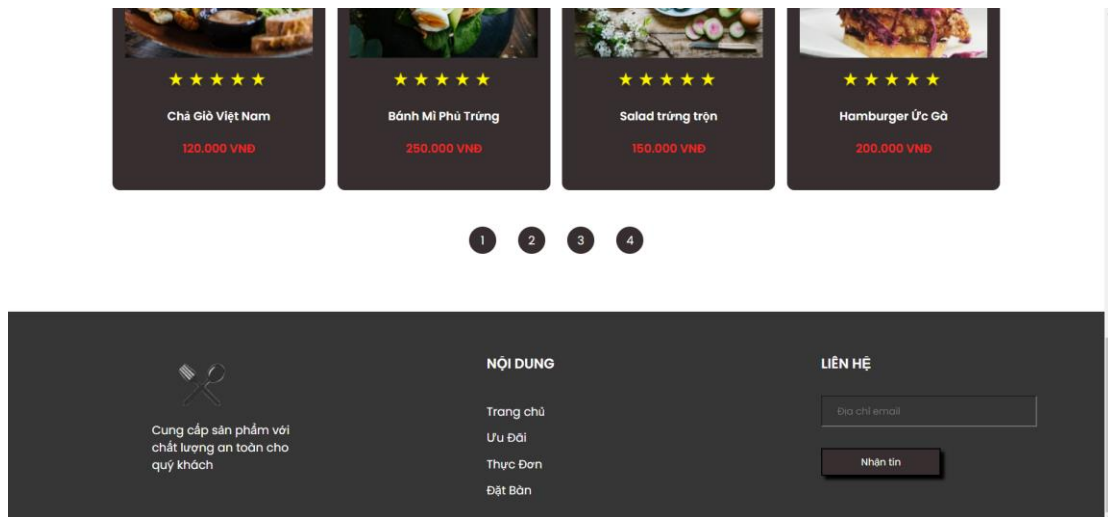


GIAO DIỆN THỰC HÀNH BUỔI 2

Giao diện trang chủ bài viết.

Tổng quan :





Phần header gồm có:

- Logo của nhà hàng
- Thanh Menu

Phần menu gồm có các mục để tham khảo như:

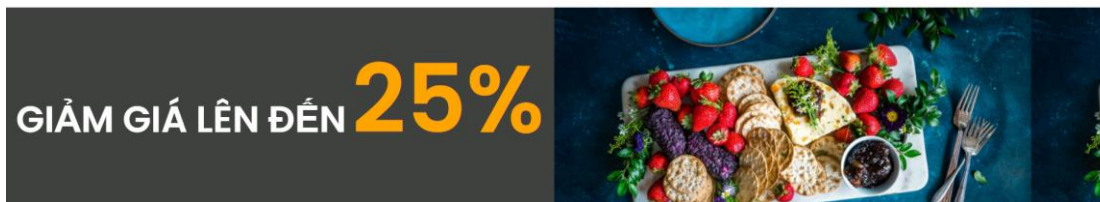
- Trang chủ
- Ưu đãi
- Thực đơn
- Đặt bàn

Thanh Menu có các tác vụ chung của 1 nhà hàng , có thể xem được thực đơn , khuyến mãi ưu đãi , và cách thức đặt bàn của Nhà hàng

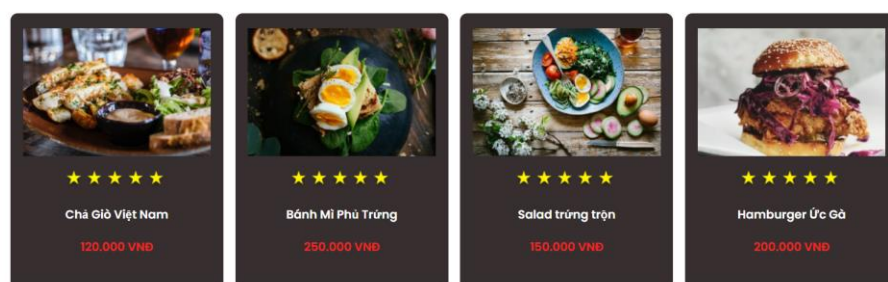


Các phần đầu của trang web gồm có :

- Phần giới thiệu của nhà hàng , tổng quan có thể nhìn thấy được cách mua đồ ăn qua trang web
- Phần giá ưu đãi nhất của nhà hàng đồng thời cũng sẽ được đăng lên trang chủ
- Phần thực đơn được đăng lên với những món ăn được nhiều người đánh giá cao và ngon nhất của nhà hàng , giúp cho khách hàng có thể chọn lựa những món ngon

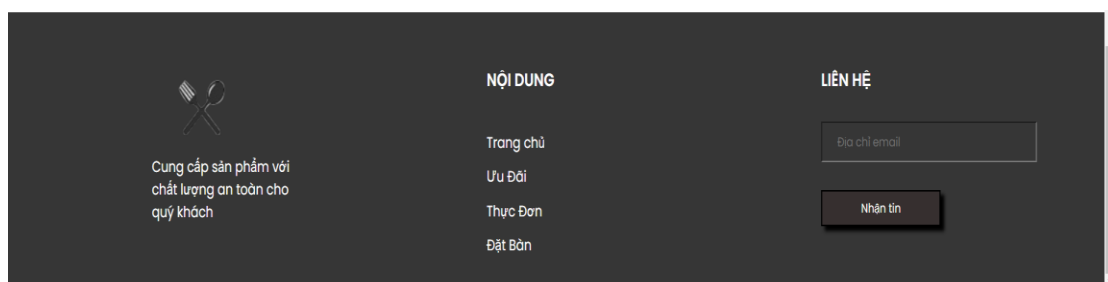


THỰC ĐƠN NHÀ HÀNG



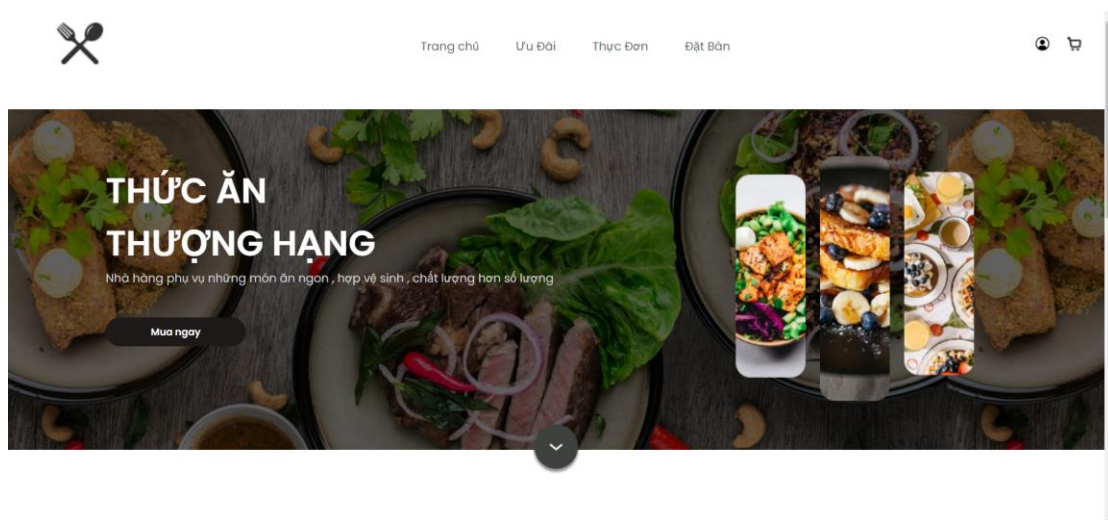
Phần chân trang gồm có các mục:

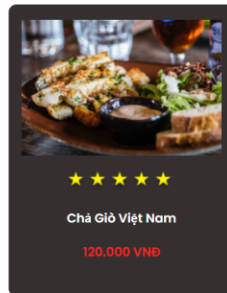
- Logo thương hiệu của Nhà hàng
- Nội dung cuối trang chủ
 - Trang chủ
 - Ưu đãi
 - Thực đơn
 - Đặt bàn
- Liên hệ để khách hàng có thể tương tác với nhà hàng




Thực đơn 1 :

Tổng quan :





Chả giò (chả ram) : Món ăn biến tấu của ba miền
Món này đặc biệt không bao giờ thiếu trong các mâm cỗ lễ tết hay giỗ chạp của người bắc. Cũng là những nguyên liệu cơ bản, món chả giò của người bắc ngoài thành phần chính là thịt nạc vai xay, bao giờ cũng có mộc nhĩ và giá đỗ, su hào cắt sợi, vắt khô. Chả giò truyền thống bao gồm thịt nạc xay, củ đậu, cà rốt, nấm hương, miến, mộc nhĩ và trứng. Nhân được thái nhỏ, trộn đều và được gói lại xinh xắn bên ngoài lớp bánh tráng. Hương vị thơm ngon, sự cân bằng về hương vị đã tạo nên sự độc đáo cho món ăn này. Miếng chả giò được chiên vàng đều. Ngay khi cắn một miếng ta có thể cảm nhận được sự giòn tan, hơi mặn của bánh tráng. Vị ngọt, chín tới của rau củ bên trong. Tất cả đã tạo nên tổng thể hương vị riêng biệt, chỉ có tại ẩm thực Việt Nam.

	NỘI DUNG	LIÊN HỆ
 Cung cấp sản phẩm với chất lượng an toàn cho quý khách	Trang chủ Ưu Đãi Thực Đơn Đặt Bàn	<input type="text" value="Địa chỉ email"/> <input type="button" value="Nhận tin"/>

Mô tả chi tiết :

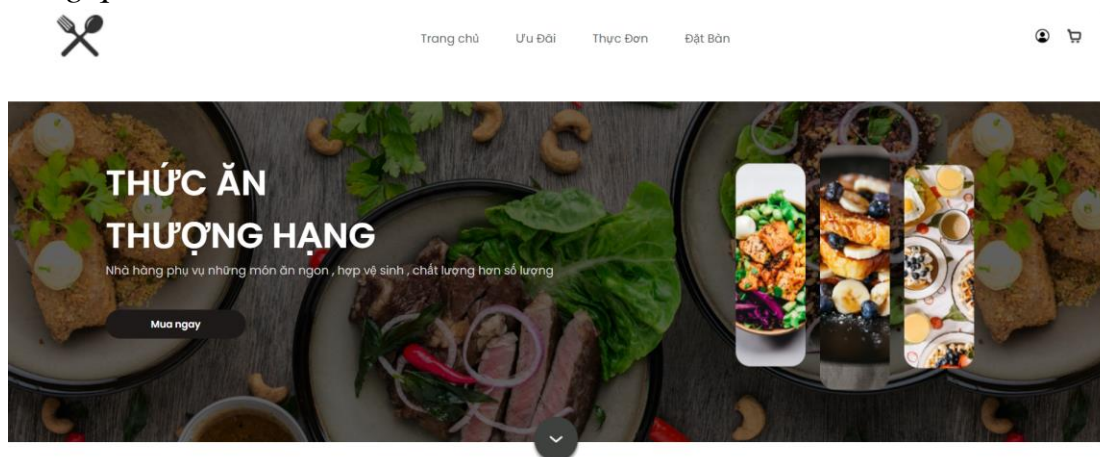
Phần bên phải mô tả các bài viết khác độc giả đọc bài báo tiếp mà không cần quay về tranh chủ.

Phần header của trang chủ thực đơn 1 , sẽ có giao diện như giao diện của trang chủ ban đầu.

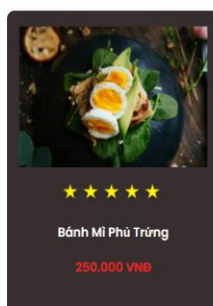
Khác với lúc đầu thì ở phần thực đơn 1 sẽ có thể thấy được những giới thiệu về món ăn , những đánh giá của khách hàng đã thưởng thức tại nhà hàng , tên của món ăn , giá cả.

Thực đơn 2 :

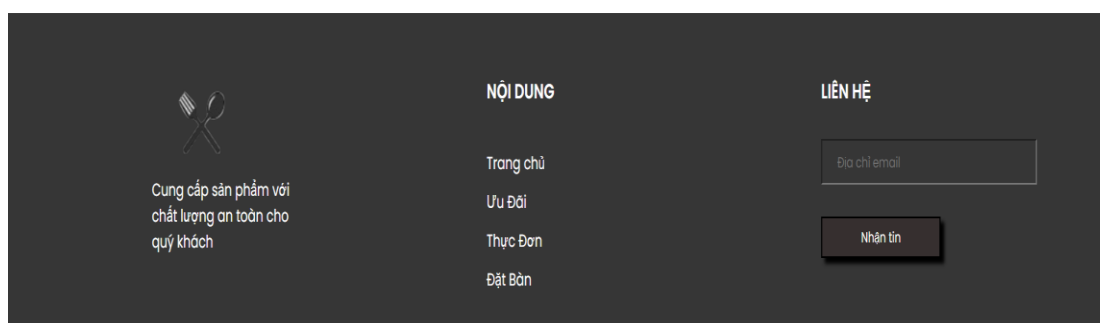
Tổng quan :



GIỚI THIỆU MÓN ĂN CỦA NHÀ HÀNG

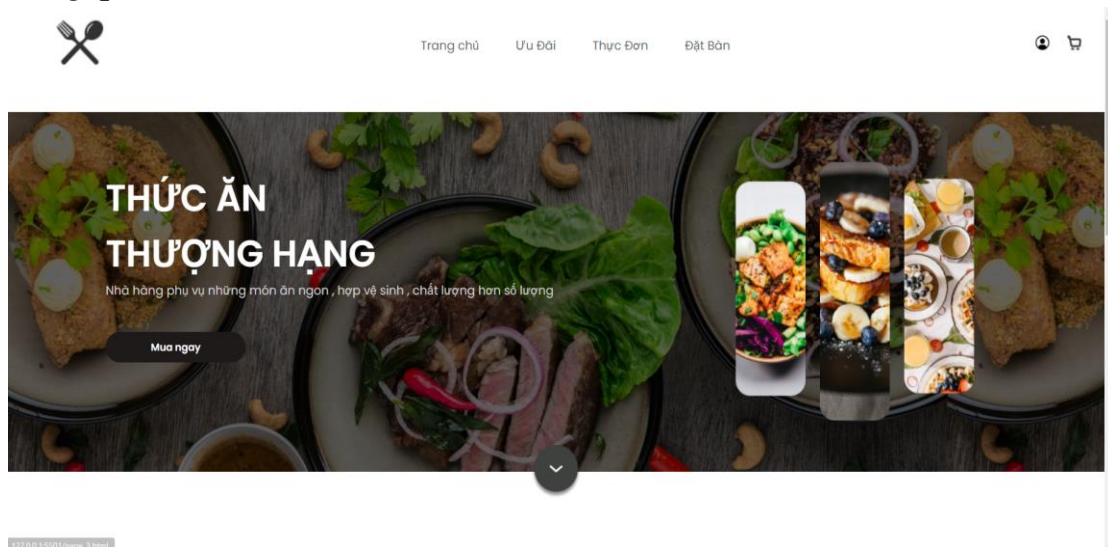


Bánh mì phở trứng: Lớp vỏ giòn giòn, ruột bánh mềm kèm với trứng thơm cùng nhiều loại rau tươi khiến bánh mì kẹp trứng có hương vị thơm ngon, kích thích vị giác của bạn. Nếu bạn đang suy nghĩ nên ăn gì vào bữa sáng để vừa đảm bảo dinh dưỡng vừa không mất quá nhiều thời gian, công sức chế biến thì bánh mì kẹp trứng là lựa chọn hoàn hảo đấy. Cách làm bánh mì kẹp trứng rất đơn giản, chỉ cần bạn chuẩn bị một số nguyên liệu và thực hiện đúng các bước dưới đây, bạn đã có món ăn sáng ngon miệng rồi.

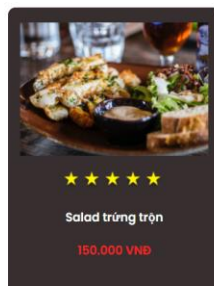


Thực đơn 3 :

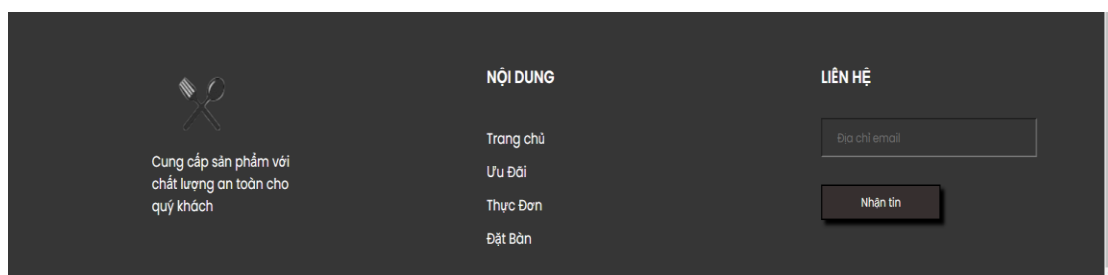
Tổng quan :



GIỚI THIỆU MÓN ĂN CỦA NHÀ HÀNG

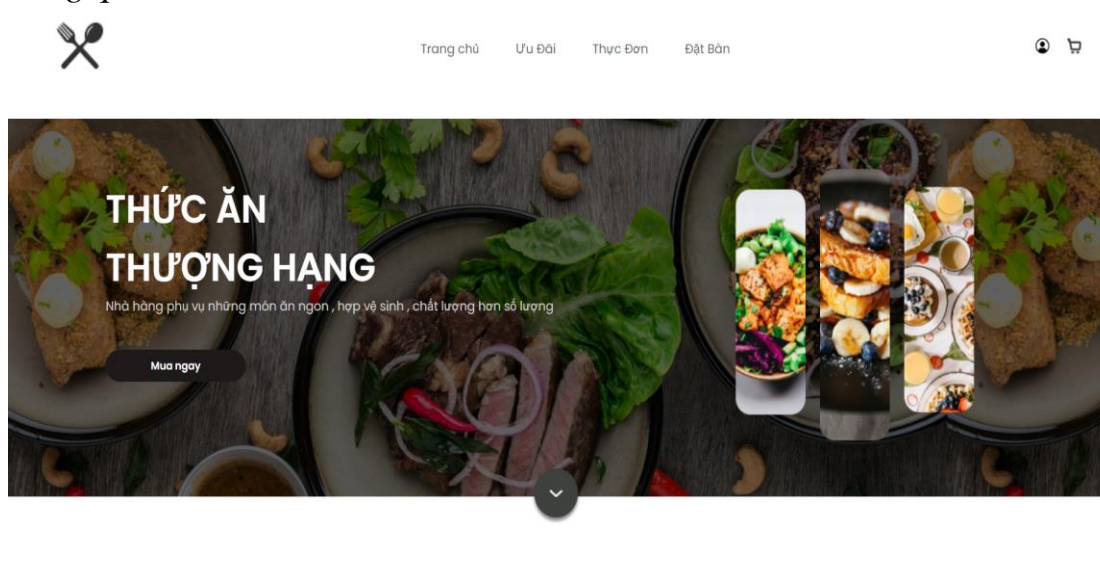


Salad trứng trộn : Món salad bơ trứng với cách làm cực kỳ đơn giản, nhanh chóng nhưng lại vô cùng hấp dẫn. Rau củ tươi ngon, giòn giòn kết hợp với hỗn hợp nước sốt chua ngọt cùng trứng và bơ béo béo cực kỳ kích thích vị giác đấy nhé. Món salad trứng giảm cân béo béo thơm ngon dễ ăn. Bởi đa phần là các loại rau củ nên món ăn chứa nhiều chất xơ sinh năng lượng ít giúp hỗ trợ quá trình ăn giảm bớt Calories của mình đấy.



Thực đơn 4 :

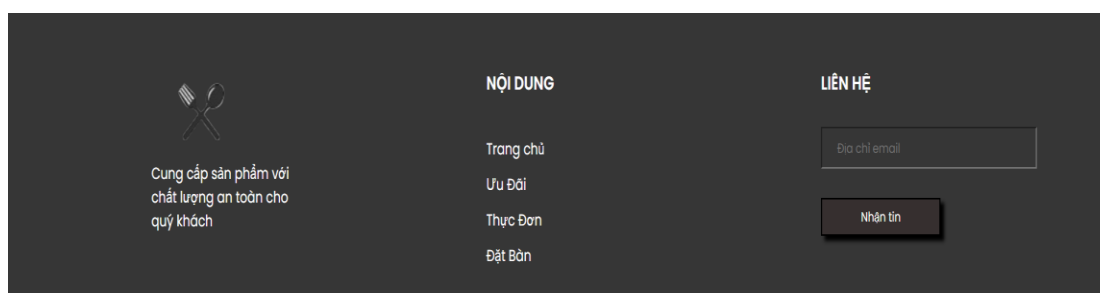
Tổng quan :



GIỚI THIỆU MÓN ĂN CỦA NHÀ HÀNG




Hamburger Úc Gà : Trong thời buổi giá cả leo thang, những chiếc burger gà thơm ngon ngoài tiệm trở nên thật đắt tiền. Cách làm hamburger gà vừa tiết kiệm tiền, vừa đảm bảo an toàn vệ sinh cho gia đình và hơn thế nữa là cả nhà bạn sẽ rất thích thú vì được thưởng thức những chiếc burger cực ngon do chính tay bạn làm ra. Hamburger Gà với vị bánh mì thơm thơm, mềm mềm, kết hợp với một miếng thịt gà to, một ít rau xà lách giòn giòn, vài giọt sốt chua cay...tạo nên một hương vị không thể nào quên được cho gia đình bạn.



Giao diện trang liên hệ

Tổng quan



The image shows a login form titled "ĐĂNG NHẬP" (Login) centered on a light gray background. The form is a white rectangle with a thin gray border. It contains four input fields stacked vertically, each with a placeholder label: "Họ và tên" (Full name), "Tên tài khoản" (Account name), "Mật khẩu" (Password), and "Email". Below the input fields is a blue button with the text "ĐĂNG NHẬP" in white capital letters.

Trang liên hệ chúng là hệ thống đăng nhập của Nhà Hàng :

- Họ và tên
- Tên tài khoản
- Mật khẩu
- Email dùng để nhận tin nhắn xác nhận của nhà hàng

CHƯƠNG 3: SOURCE CODE

SOURCE CODE HTML TRANG CHỦ :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>NHÀ HÀNG Á ÂU </title>

  <link rel="stylesheet" href="style.css">

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

</head>

<body>

  <div id="wrapper">

    <div id="header">

      <a href="" class="logo">

      </a>

      <div id="menu">

        <div class="item">

          <a href="">Trang chủ</a>

        </div>

        <div class="item">

          <a href="">Ưu Đãi</a>

        </div>

        <div class="item">

          <a href="">Thực Đơn</a>

        </div>

      </div>

    </div>

  </body>

</html>
```

```

<div class="item">

    <a href="">Đặt Bàn</a>

</div>

</div>

<div id="actions">

    <div class="item">

        <a href="Form/lien-he,.html"></a>

    </div>

    <div class="item">

    </div>

</div>

</div>

<div id="banner">

    <div class="box-left">

        <h2>

            <span>THỨC ĂN</span>

            <br>

            <span>THUỢNG HẠNG</span>

        </h2>

        <p> Nhà hàng phục vụ những món ăn ngon , hợp vệ sinh , chất lượng hơn số
lượng</p>

        <button>Mua ngay</button>

    </div>

    <div class="box-right">

```

```

</div>

<div class="to-bottom">

    <a href="">

    </a>

</div>

</div> <br><br><br><br>

<div id="saleoff">

    <div class="box-left">

        <h1>

            <span>GIẢM GIÁ LÊN ĐẾN</span>

            <span>25%</span>

        </h1>

    </div>

    <div class="box-right"></div>

</div>

<div id="wp-products">

    <h2>THỰC ĐƠN NHÀ HÀNG</h2>

    <ul id="list-products">

        <!-- món 1 -->

        <div class="item">

            <a href="page_1.html"></a>

            <div class="stars">

                <span>

                </span>

                <span>


```



```

</span>

<span>
    
</span>

<span>
    
</span>

<span>
    
</span>

</div>

<div class="name">Chả Giò Việt Nam</div>

<div class="price">120.000 VNĐ</div>

</div>

<!-- Món 2 -->

<div class="item">

    <a href="page_2.html"></a>

    <div class="stars">

        <span>
            
        </span>

        <span>
            
        </span>

        <span>
            
        </span>

    </div>

</div>

```

```

        <span>

        </span>

        <span>

        </span>

    </div>

    <div class="name">Bánh Mì Phở Trứng</div>

    <div class="price">250.000 VNĐ</div>

</div>

<!-- Món 3 -->

<div class="item">

    <a href="page_3.html"></a>

    <div class="stars">

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

    </div>


```

```

        </span>

    </div>

    <div class="name">Salad trứng trộn</div>

    <div class="price">150.000 VNĐ</div>

</div>

<!-- Món 4 -->

<div class="item">

    <a href="page_4.html"></a>

    <div class="stars">

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

    </div>

    <div class="name">Hamburger Úc Gà</div>

    <div class="price">200.000 VNĐ</div>

</div>

```

```
</ul>

<div class="list-page">

  <div class="item">

    <a href="">1</a>

  </div>

  <div class="item">

    <a href="">2</a>

  </div>

  <div class="item">

    <a href="">3</a>

  </div>

  <div class="item">

    <a href="">4</a>

  </div>

</div>

</div>

<!-- footer -->

<footer>

  <div id="footer">

    <div class="box">

      <div class="logo" >

      </div>

      <p>Cung cấp sản phẩm với chất lượng an toàn cho quý khách</p>

    </div>

    <div class="box">

      <h3>NỘI DUNG</h3>

      <ul class="quick-menu">
```

```
<div class="item">

    <a href="">Trang chủ</a>

</div>

<div class="item">

    <a href="">Ưu Đãi</a>

</div>

<div class="item">

    <a href="">Thực Đơn</a>

</div>

<div class="item">

    <a href="">Đặt Bàn</a>

</div>

</ul>

</div>

<div class="box">

    <h3>LIÊN HỆ</h3>

    <form action="">

        <input type="text" placeholder="Địa chỉ email">

        <button>Nhận tin</button>

    </form>

</div>

</div>

<script src="script.js"></script>

</footer>

<script src="js/jquery/jquery-2.2.4.min.js"></script>

<script src="js/bootstrap/popper.min.js"></script>
```

```
<script src="js/bootstrap/bootstrap.min.js"></script>

<script src="js/plugins/plugins.js"></script>

<script src="js/active.js"></script>

</body>

</html>
```

SOURCE CODE HTML THỰC ĐƠN 1 :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Web bán thức ăn </title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div id="wrapper">

    <div id="header">

      <a href="" class="logo">

      </a>

      <div id="menu">

        <div class="item">

          <a href="">Trang chủ</a>

        </div>

        <div class="item">

          <a href="">Ưu Đãi</a>
```

```

</div>

<div class="item">

    <a href="">Thực Đơn</a>

</div>

<div class="item">

    <a href="">Đặt Bàn</a>

</div>

</div>

<div id="actions">

    <div class="item">

    </div>

    <div class="item">

    </div>

</div>

</div>

<div id="banner">

    <div class="box-left">

        <h2>

            <span>THỨC ĂN</span>

            <br>

            <span>THUỞNG HẠNG</span>

        </h2>

        <p> Nhà hàng phục vụ những món ăn ngon , hợp vệ sinh , chất lượng hơn số
lượng</p>

        <button>Mua ngay</button>

    </div>

```

```
<div class="box-right">

</div>

<div class="to-bottom">

  <a href="">

  </a>

</div>

</div> <br><br>

<div id="wp-products">

  <h2>GIỚI THIỆU MÓN ĂN CỦA NHÀ HÀNG</h2>

  <ul id="list-products">

    <div class="item">

      <div class="stars">

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

      </div>

    </div>

  </ul>

</div>
```


</div>

<div class="name">Chả Giò Việt Nam</div>

<div class="price">120.000 VNĐ</div>

</div>

<div>

Chả giò (chả ram) : Món ăn biến tấu của ba miền
 Món này đặc biệt không bao giờ thiếu trong các mâm cỗ lễ tết hay giỗ chạp của người bắc. Cũng là những nguyên liệu cơ bản, món chả giò của người bắc ngoài thành phần chính là thịt nạc vai xay, bao giờ cũng có mộc nhĩ và giá đỗ, su hào cắt sợi, vắt khô.

Chả giò truyền thống bao gồm thịt nạc xay, củ đậu, cà rốt, nấm hương, miến, mộc nhĩ và trứng. Nhân được thái nhỏ, trộn đều và được gói lại xinh xắn bên ngoài lớp bánh tráng. Hương vị thơm ngon, sự cân bằng về hương vị đã tạo nên sự độc đáo cho món ăn này. Miếng chả giò được chiên vàng đều. Ngay khi cắn một miếng ta có thể cảm nhận được sự giòn tan, hơi mặn của bánh tráng. Vị ngọt, chín tới của rau củ bên trong. Tất cả đã tạo nên tổng thể hương vị riêng biệt, chỉ có tại ẩm thực Việt Nam.

</div>

</div>

<div id="footer">

<div class="box">

<div class="logo" >

</div>

<p>Cung cấp sản phẩm với chất lượng an toàn cho quý khách</p>

```

</div>

<div class="box">

    <h3>NỘI DUNG</h3>

    <ul class="quick-menu">

        <div class="item">

            <a href="">Trang chủ</a>

        </div>

        <div class="item">

            <a href="">Ưu Đãi</a>

        </div>

        <div class="item">

            <a href="">Thực Đơn</a>

        </div>

        <div class="item">

            <a href="">Đặt Bàn</a>

        </div>

    </ul>

</div>

<div class="box">

    <h3>LIÊN HỆ</h3>

    <form action="">

        <input type="text" placeholder="Địa chỉ email">

        <button>Nhận tin</button>

    </form>

</div>

</div>

<script src="script.js"></script>

```

```
</body>
```

```
</html>
```

SOURCE CODE HTML THỰC ĐƠN 2 :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Web bán thức ăn </title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <div id="wrapper">

    <div id="header">

      <a href="" class="logo">

      </a>

      <div id="menu">

        <div class="item">

          <a href="">Trang chủ</a>

        </div>

        <div class="item">

          <a href="">Ưu Đãi</a>

        </div>

        <div class="item">

          <a href="">Thực Đơn</a>
```

```

</div>

<div class="item">

    <a href="">Đặt Bàn</a>

</div>

</div>

<div id="actions">

    <div class="item">

    </div>

    <div class="item">

    </div>

</div>

</div>

<div id="banner">

    <div class="box-left">

        <h2>

            <span>THỨC ĂN</span>

            <br>

            <span>THƯƠNG HẠNG</span>

        </h2>

        <p> Nhà hàng phục vụ những món ăn ngon , hợp vệ sinh , chất lượng hơn số
lượng</p>

        <button>Mua ngay</button>

    </div>

    <div class="box-right">

```

```


</div>

<div class="to-bottom">

  <a href="">

  </a>

</div>

</div> <br><br>

<div id="wp-products">

  <h2>GIỚI THIỆU MÓN ĂN CỦA NHÀ HÀNG</h2>

  <ul id="list-products">

    <div class="item">

      <div class="stars">

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

      </div>

    </div>

  </ul>

</div>
```


</div>

<div class="name">Bánh Mì Phở Trứng</div>

<div class="price">250.000 VNĐ</div>

</div>

<div>

 Bánh mì phở trứng : Lớp vỏ giòn giòn, ruột bánh mềm kèm với trứng thơm cùng nhiều loại rau tươi khiến bánh mì kẹp trứng có hương vị thơm ngon, kích thích vị giác của bạn. Nếu bạn đang suy nghĩ nên ăn gì vào bữa sáng để vừa đảm bảo dinh dưỡng vừa không mất quá nhiều thời gian, công sức chế biến thì bánh mì kẹp trứng là lựa chọn hoàn hảo đấy. Cách làm bánh mì kẹp trứng rất đơn giản, chỉ cần bạn chuẩn bị một số nguyên liệu và thực hiện đúng các bước dưới đây, bạn đã có món ăn sáng ngon miệng rồi.

</div>

</div>

<div id="footer">

<div class="box">

<div class="logo" >

</div>

<p>Cung cấp sản phẩm với chất lượng an toàn cho quý khách</p>

</div>

<div class="box">

<h3>NỘI DUNG</h3>

<ul class="quick-menu">

<div class="item">

Trang chủ

</div>

<div class="item">

```

        <a href="">Ưu Đãi</a>

    </div>

    <div class="item">

        <a href="">Thực Đơn</a>

    </div>

    <div class="item">

        <a href="">Đặt Bàn</a>

    </div>

</ul>

</div>

<div class="box">

    <h3>LIÊN HỆ</h3>

    <form action="">

        <input type="text" placeholder="Địa chỉ email">

        <button>Nhận tin</button>

    </form>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

```

SOURCE CODE HTML THỰC ĐƠN 3 :

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Web bán thức ăn </title>

<link rel="stylesheet" href="style.css">

</head>

<body>

  <div id="wrapper">

    <div id="header">

      <a href="" class="logo">

      </a>

      <div id="menu">

        <div class="item">

          <a href="">Trang chủ</a>

        </div>

        <div class="item">

          <a href="">Ưu Đãi</a>

        </div>

        <div class="item">

          <a href="">Thực Đơn</a>

        </div>

        <div class="item">

          <a href="">Đặt Bàn</a>

        </div>

      </div>

      <div id="actions">

        <div class="item">

        </div>

      </div>

    </div>

  </body>

</html>
```



```

</div>

<div class="item">

</div>

</div>

</div>

<div id="banner">

    <div class="box-left">

        <h2>

            <span>THỨC ĂN</span>

            <br>

            <span>THƯƠNG HÀNG</span>

        </h2>

        <p> Nhà hàng phục vụ những món ăn ngon , hợp vệ sinh , chất lượng hơn số
lượng</p>

        <button>Mua ngay</button>

    </div>

    <div class="box-right">

    </div>

    <div class="to-bottom">

        <a href="">

        </a>

    </div>

</div> <br><br>

```

```
<div id="wp-products">

  <h2>GIỚI THIỆU MÓN ĂN CỦA NHÀ HÀNG</h2>

  <ul id="list-products">

    <div class="item">

      <div class="stars">

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

        <span>

        </span>

      </div>

      <div class="name">Salad trứng trộn</div>

      <div class="price">150.000 VNĐ</div>

    </div>

  </ul>

  <div>

    <span>
```

Salad trứng trộn : Món salad bơ trứng với cách làm cực kỳ đơn giản, nhanh chóng nhưng lại vô cùng hấp dẫn. Rau củ tươi ngon, giòn giòn kết hợp với hỗn hợp nước sốt chua ngọt cùng trứng và bơ beo béo cực kỳ kích thích vị giác đấy nhé.

Món salad trứng giảm cân beo béo thơm ngon dễ ăn. Bởi đa phần là các loại rau củ nên món ăn chứa nhiều chất xơ sinh năng lượng ít giúp hỗ trợ quá trình ăn thâm hụt Calories của mình đấy.

```
</span>

</div>

</div>

<div id="footer">

  <div class="box">

    <div class="logo" >

    </div>

    <p>Cung cấp sản phẩm với chất lượng an toàn cho quý khách</p>

  </div>

  <div class="box">

    <h3>NỘI DUNG</h3>

    <ul class="quick-menu">

      <div class="item">

        <a href="">Trang chủ</a>

      </div>

      <div class="item">

        <a href="">Ưu Đãi</a>

      </div>

      <div class="item">

        <a href="">Thực Đơn</a>

      </div>

      <div class="item">

        <a href="">Đặt Bàn</a>

      </div>

    </ul>

  </div>

</div>
```

```

        </div>

    </ul>

</div>

<div class="box">

    <h3>LIÊN HỆ</h3>

    <form action="">

        <input type="text" placeholder="Địa chỉ email">

        <button>Nhận tin</button>

    </form>

</div>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

```

SOURCE CODE HTML THỰC ĐƠN 4 :

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Web bán thức ăn </title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

```

```
<div id="wrapper">

  <div id="header">

    <a href="" class="logo">

    </a>

    <div id="menu">

      <div class="item">

        <a href="">Trang chủ</a>

      </div>

      <div class="item">

        <a href="">Ưu Đãi</a>

      </div>

      <div class="item">

        <a href="">Thực Đơn</a>

      </div>

      <div class="item">

        <a href="">Đặt Bàn</a>

      </div>

    </div>

    <div id="actions">

      <div class="item">

      </div>

      <div class="item">

      </div>

    </div>

  </div>

</div>
```

```

<div id="banner">

  <div class="box-left">

    <h2>

      <span>THỨC ĂN</span>

      <br>

      <span>THƯỢNG HẠNG</span>

    </h2>

    <p> Nhà hàng phục vụ những món ăn ngon , hợp vệ sinh , chất lượng hơn số
lượng</p>

    <button>Mua ngay</button>

  </div>

  <div class="box-right">

  </div>

  <div class="to-bottom">

    <a href="">

    </a>

  </div>

</div> <br><br>

<div id="wp-products">

  <h2>GIỚI THIỆU MÓN ĂN CỦA NHÀ HÀNG</h2>

  <ul id="list-products">

    <div class="item">

      <div class="stars">

```

```
<span>
    
```

```
</span>
```

```
<span>
```

```
    
```

```
</span>
```

```
<span>
```

```
    
```

```
</span>
```

```
<span>
```

```
    
```

```
</span>
```

```
<span>
```

```
    
```

```
</span>
```

```
</div>
```

```
<div class="name">Hamburger Úc Gà</div>
```

```
<div class="price">200.000 VNĐ</div>
```

```
</div>
```

```
<div>
```

```
<span>
```

Hamburger Úc Gà : Trong thời buổi giá cả leo thang, những chiếc burger gà thơm ngon ngoài tiệm trở nên thật đắt tiền. Cách làm hamburger gà vừa tiết kiệm tiền, vừa đảm bảo an toàn vệ sinh cho gia đình và hơn thế nữa là cả nhà bạn sẽ rất thích thú vì được thưởng thức những chiếc burger cực ngon do chính tay bạn làm ra.

```
</span>
```

```
<span>
```

Hamburger Gà với vị bánh mì thơm thơm, mềm mềm, kết hợp với một miếng thịt gà to, một ít rau xà lách giòn giòn, vài giọt sốt chua cay...tạo nên một hương vị không thể nào quên được cho gia đình bạn.

```

    </span>

</div>

</div>

<div id="footer">

    <div class="box">

        <div class="logo" >

        </div>

        <p>Cung cấp sản phẩm với chất lượng an toàn cho quý khách</p>

    </div>

    <div class="box">

        <h3>NỘI DUNG</h3>

        <ul class="quick-menu">

            <div class="item">

                <a href="">Trang chủ</a>

            </div>

            <div class="item">

                <a href="">Ưu Đãi</a>

            </div>

            <div class="item">

                <a href="">Thực Đơn</a>

            </div>

            <div class="item">

                <a href="">Đặt Bàn</a>

            </div>

        </ul>

    </div>

    <div class="box">

```



```

        <h3>LIÊN HỆ</h3>

        <form action="">

            <input type="text" placeholder="Địa chỉ email">

            <button>Nhận tin</button>

        </form>

    </div>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

```

SOURCE CODE HTML LIÊN HỆ :

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

    <title>NHÀ HÀNG Á ÂU</title>

</head>

<body>

    <div class="register-form">

        <h2>ĐĂNG NHẬP</h2>

        <div class="input-row">

            <input type="text" placeholder="Họ và tên" id="username" autocomplete="off" />

```

```

        <small>Error message</small>

    </div>

    <div class="input-row">

        <input type="text" placeholder="Tên tài khoản" id="subject" autocomplete="off" />

        <small>Error message</small>

    </div>

    <div class="input-row">

        <input type="number" placeholder="Mật khẩu" id="message" autocomplete="off" />

        <small>Error message</small>

    </div>

    <div class="input-row">

        <input type="Email" placeholder="Email" id="email" autocomplete="off" />

        <small>Error message</small>

    </div>

    <div class="input-row">

        <button class="btn" id="btn-register">

            Đăng nhập

        </button>

    </div>

</div>

</body>

<script src="./main.js"></script>

</html>

```

SOURCE CODE CSS TRANG CHỦ :

```

*{

    padding:0;

```

```
margin:0;

box-sizing: border-box;

font-family:'Poppins', sans-serif;

cursor: pointer;
}

#wrapper {

  /* Kích thước màn hình là 1280 x 720 */

  width: 100%;

  height: 100vh;

}

#header {

  width: 100%;

  padding:0px 30px;

  margin-top:3px;

  display: flex;

  justify-content: space-between;

  align-items: center;

}

#menu {

  list-style:none;

  display: flex;

}

#menu .item {

  margin:0px 25px;

}
```

```
#menu .item a {  
    color:#626A67;  
    text-decoration: none;  
}  
  
#actions {  
    display: flex;  
}  
  
#actions .item {  
    margin-left:22px;  
}  
  
#banner {  
    width: 100%;  
    background-image :url("assets/banner.png");  
    height: 463px;  
    margin-top:40px;  
    display: flex;  
    padding:0px 133px;  
    position:relative;  
}  
  
#banner .box-left ,#banner .box-right {  
    width: 50%;  
}  
  
#banner .box-left h2 {  
    font-size:48px;  
    margin-top:75px;  
    color:#fff;  
}  
  
#banner .box-left p {
```

```
    color:#D0D0D0;
}

#banner .box-left button {

    width: 191px;

    height: 40px;

    margin-top:41px;

    background:#1d1a1a;

    border:none;

    outline:none;

    color:#fff;

    font-weight: bold;

    border-radius: 20px;

    transition:0.4s;

}

#banner .box-left button:hover {

    background:orange;

}

#banner .box-right {

    padding-left:200px;

    display: flex;

    justify-content: center;

    align-items: center;

}

#banner .box-right img {

    margin:0px 10px;

    animation :animation 3s infinite linear;

}
```

```
#banner .box-right img:nth-child(1)
{
    animation-delay: 0.5s;
}
#banner .box-right img:nth-child(2)
{
    animation-delay: 1s;
}
@keyframes animation {
    0%{
        transform:translateY(0px);
    }
    50%{
        transform: translateY(-10px);
    }
    100%{
        transform: translateY(0px);
    }
}
.to-bottom {
    position:absolute;
    bottom:-40px;
    left:50%;
    transform:translateX(-50%);
}
```

```
#wp-products {  
  padding-top:80px;  
  padding-bottom: 78px;  
  padding-left:134px;  
  padding-right:134px;  
}  
#wp-products h2 {  
  text-align: center;  
  margin-bottom: 70px;  
  font-size:32px;  
  color:#626a67;  
}  
#list-products {  
  display: flex;  
  list-style: none;  
  justify-content: space-around;  
  align-items: center;  
  flex-wrap: wrap;  
}  
#list-products .item {  
  width: 290px;  
  height: 374px;  
  background:#362f2f;  
  border-radius: 10px;  
  margin-bottom: 50px;  
}  
#list-products .item_1 {  
  width: 300px;
```

```
height: 300px;

background:#362f2f;

border-radius: 10px;

margin-bottom: 50px;
}

#list-products .item img {

display: block;

margin:0px auto;

margin-top:20px;
}

#list-products .item .stars {

display: flex;

margin:0px auto;

width: 50%;
}

#list-products .item .name {

text-align: center;

color:#fff;

font-weight: bold;

margin-top:21px;
}

#list-products .item .desc {

text-align: center;

color:#626a67;
}

#list-products .item .price {

text-align: center;

color:#f42424;
```



```
    font-weight: bold;

    margin-top:20px;
}

.list-page {

    width: 50%;

    margin:0px auto;
}

.list-page {

    display: flex;

    list-style: none;

    justify-content: center;

    align-items: center;
}

.list-page .item {

    margin:0px 15px;

    width: 37px;

    height: 37px;

    background:#362f2f;

    border-radius: 50%;

    display: flex;

    justify-content: center;

    align-items: center;
}

.list-page .item a {

    color:#fff;

    text-decoration: none;
}

#saleoff {
```

```
width: 100%;

display: flex;

height: 270px;
}

#saleoff .box-left ,#saleoff .box-right {

width: 50%;

}

#saleoff .box-left {

background:#3e413e;

text-align: center;

min-height: 270px;

}

#saleoff .box-left h1 {

padding-top:28px;

}

#saleoff .box-left span:nth-child(1)

{

color:#fff;

font-size:50px;

}

#saleoff .box-left span:nth-child(2)

{

color:orange;

font-size:128px;

}

#saleoff .box-right {

background-image:url("assets/off.png");

}
```

```
#comment {  
  
    width: 1012px;  
  
    height: 478px;  
  
    margin:0px auto;  
  
    margin-top:184px;  
  
    margin-bottom: 50px;  
  
}  
  
#comment h2 {  
  
    text-align: center;  
  
    margin-bottom: 50px;  
  
    font-size:32px;  
  
    color:#626a67;  
  
}  
  
#comment-body{  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: center;  
  
    width: 1000px;  
  
    height: 400px;  
  
    list-style: none;  
  
    overflow: hidden;  
  
}  
  
#list-comment {  
  
    width: 100%;  
  
    height: 100%;  
  
    list-style: none;  
  
    transition:0.5s;
```

```
}  
  
#list-comment .item {  
    width: 100%;  
    height: 100%;  
    background:#362f2f;  
    border-radius: 20px;  
    padding-top:50px;  
}  
  
#list-comment .item .avatar {  
    text-align: center;  
    margin-top:13px;  
}  
  
#list-comment .item .stars {  
    display: flex;  
    margin:0px auto;  
    width: 50%;  
    margin-top:32px;  
    justify-content: center;  
    align-items: center;  
}  
  
#list-comment .item .name {  
    text-align: center;  
    font-weight: bold;  
    color:#fff;  
}  
  
#list-comment .item .text {  
    padding:20px 50px;  
    color:#fff;
```

```
}  
  
#footer {  
    width: 100%;  
    background:rgb(0,0,0,0.79);  
    height: 290px;  
    padding:0px 63px;  
    padding-top:57px;  
    display: flex;  
    justify-content: space-around;  
}  
  
#footer .box {  
    width: 190px;  
    color:#fff;  
}  
  
#footer .box .quick-menu {  
    margin-top:40px;  
}  
  
  
#footer .box .quick-menu .item{  
    margin-bottom:12px;  
}  
  
#footer .box .quick-menu ,.item a{  
    color:#fff;  
    text-decoration: none;  
}
```

```

#footer .box form input {

    width: 294px;

    height: 42px;

    background:transparent;

    padding-left:20px;

    color:#fff;

    margin-top:30px;

}

#footer .box form button {

    background:#362f2f;

    box-shadow: 5px 5px 4px rgba(0,0,0.25);

    width: 163px;

    height: 38px;

    margin-top:29px;

    color:#fff;

}

```

SOURCE CODE CSS LIÊN HỆ :

```

*,
::after,
::before {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    background-color: #f6f6f6;

    color: #34465c;

```

```
font-size: 16px;

font-family: Roboto, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;
}

.register-form {

width: 400px;

padding: 2rem;

box-shadow: 0 0 30px rgba(0, 0, 0, 0.1);

background-color: #fff;

border-radius: 7px;
}

.register-form h2 {

margin-bottom: 1.5rem;

text-align: center;
}

.input-row {

margin-bottom: 1.5rem;
}

.input-row input {

width: 100%;

border: 1px solid #ccc;

height: 40px;

line-height: 40px;

padding: 0 1rem;

outline: none;
```

```
    transition: all 0.3s ease-out;
  }

  .input-row input:focus {
    border: 1px solid #000abf;
    outline: none;
  }

  .input__note textarea {
    width: 100%;
    min-height: 200px;
    border: 1px solid #ccc;
    user-select: none;
    resize: none;
    padding: 1rem;
  }

  .input__note textarea:focus {
    border: 1px solid #000abf;
  }

  .input-row {
    position: relative;
  }

  .input-row:last-child {
    margin-bottom: 0;
  }

  .input-row.success input {
    border-color: #000abf;
  }

  .input-row.error input {
    border-color: #e74c3c;
```



```
}  
  
.input-row small {  
    color: #e74c3c;  
    position: absolute;  
    bottom: -20px;  
    left: 0;  
    visibility: hidden;  
    font-size: 12px;  
}  
  
.input-row.error small {  
    visibility: visible;  
}  
  
.btn {  
    font-size: 12px;  
    font-weight: 400;  
    text-transform: uppercase;  
    letter-spacing: 1px;  
    -webkit-border-radius: 2px;  
    -moz-border-radius: 2px;  
    -ms-border-radius: 2px;  
    border-radius: 2px;  
    -webkit-transition: 0.5s;  
    -o-transition: 0.5s;  
    transition: 0.5s;  
    padding: 8px 15px !important;  
}  
  
#btn-register {  
    display: inline-block;
```

```
color: #fff;

background-color: #000abf;

border: 2px solid #000abf;

outline: 0;

cursor: pointer;

text-decoration: none !important;

position: relative;

z-index: 20;

overflow: hidden;

}
```

SOURCE CODE JAVASCRIPT TRANG CHỦ :

```
const next = document.querySelector('.next')

const prev = document.querySelector('.prev')

const comment = document.querySelector('#list-comment')

const commentItem = document.querySelectorAll('#list-comment .item')

var translateY = 0

var count = commentItem.length

next.addEventListener('click', function (event) {

  event.preventDefault()

  if (count == 1) {

    // Xem hết bình luận

    return false

  }

  translateY += -400

  comment.style.transform = `translateY(${translateY}px)`

  count--
```

```

    })

    prev.addEventListener('click', function (event) {

        event.preventDefault()

        if (count == 3) {

            // Xem hết bình luận

            return false

        }

        translateY += 400

        comment.style.transform = `translateY(${translateY}px)`

        count++

    })

```

SOURCE CODE JAVASCRIPT LIÊN HỆ :

```

const usernameEle = document.getElementById('username');
const emailEle = document.getElementById('email');
const subject = document.getElementById('subject')
const message = document.getElementById('message')
const btnRegister = document.getElementById('btn-register');
const inputEles = document.querySelectorAll('.input-row');

btnRegister.addEventListener('click', function () {

    Array.from(inputEles).map((ele) =>

        ele.classList.remove('success', 'error')

    );

    let isValid = checkValidate();

    if (isValid) {

        alert('Gửi thành công');

    }

});

```

```
function checkValidate() {  
  
    let usernameValue = usernameEle.value;  
  
    let emailValue = emailEle.value;  
  
    let subjectValue = subject.value  
  
    let messageValue = message.value  
  
    let isCheck = true;  
  
    if (usernameValue == "") {  
  
        setError(usernameEle, 'Tên không được để trống');  
  
        isCheck = false;  
  
    } else {  
  
        setSuccess(usernameEle);  
  
    }  
  
    if (messageValue == "") {  
  
        setError(message, 'Message không được');  
  
        isCheck = false;  
  
    } else {  
  
        setSuccess(message);  
  
    }  
  
    if (subjectValue == "") {  
  
        setError(subject, 'Môn học không được để trống');  
  
        isCheck = false;  
  
    } else {  
  
        setSuccess(subject);  
  
    }  
  
    if (emailValue == "") {  
  
        setError(emailEle, 'Email không được để trống');  
  
        isCheck = false;  
  
    } else if (!isEmail(emailValue)) {
```

```

        setError(emailEle, 'Email không đúng định dạng');

        isCheck = false;

    } else {

        setSuccess(emailEle);

    }

    return isCheck;

}

function setSuccess(ele) {

    ele.parentNode.classList.add('success');

}

function setError(ele, message) {

    let parentEle = ele.parentNode;

    parentEle.classList.add('error');

    parentEle.querySelector('small').innerText = message;

}

function isEmail(email) {

    return /^(^<>()\[\\]\.,;:\s@""]+(\.[^<>()\[\\]\.,;:\s@""]+)*|(".*"))@((\[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3})|((\[a-zA-Z\d-0-9]+\.)+[a-zA-Z]{2,}))$/.test(

        email

    );

}

```

SOURCE CODE CSS BOOTSTRAP :

https://github.com/BuiMinhQuy/BTL_Webdesign/tree/main/Css

PHỤ LỤC

Phụ lục 1: Link web : <http://127.0.0.1:5501/index.html>

Phụ lục 2: Link github: https://github.com/BuiMinhQuy/BTL_Webdesign

Phụ lục 3: Source code trang chủ bài viết:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/index.html

Phụ lục 4: Source code thực đơn 1:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/page_1.html

Phụ lục 5: Source code thực đơn 2:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/page_2.html

Phụ lục 6: Source code thực đơn 3:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/page_3.html

Phụ lục 7: Source code thực đơn 4:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/page_4.html

Phụ lục 8: Source code phần liên hệ:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/Form/lien-he%2C.html

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/Form/main.js

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/Form/style.css

Phụ lục 9: Source code css:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/style.css

Phụ lục 10: Source code javascript:

https://github.com/BuiMinhQuy/BTL_Webdesign/blob/main/script.js

Phụ lục 11: Source code css bootstrap:

https://github.com/BuiMinhQuy/BTL_Webdesign/tree/main/Css

