# Phần 1. Làm Quen HTML

## 1.Giới Thiệu

### Index.html

<!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="./assets/style.css">

    <title>HTML</title>

</head>

<body>

    <p id="paragraph">Hôm nay là một ngày đẹp trời!</p>

    <script src="main.js"></script>

</body>

</html>

### Style.css

#paragraph {

    font-size: 18px;

    color: red;

}

### Main.js

const p = document.getElementById('paragraph');

p.onclick = function() {

    this.innerText = 'Bạn đã CLICK vào đoạn văn bản!';

};

# Phần 2. Các Thẻ HTML

## 1.Thẻ h1, h2 (tiêu đề, phần mở đầu)

Trong HTML có 6 thẻ được dùng để làm tiêu đề từ thẻ h1 - h6. Trong đó chữ h được viết tắt của từ heading (ý nghĩa: tiêu đề, phần mở đầu). Trong bài này, mình giới thiệu trước cho các bạn 2 thẻ là <h1> và <h2> vì đây là 2 thẻ được sử dụng phổ biến nhất trong số các thẻ tiêu đề.

## 2.Thẻ p (đoạn văn bản)

## 3.Thẻ b và thẻ strong (chữ in đậm)

-Nội dung trong thẻ <b> và <strong> đều được trình duyệt hiển thị dưới dạng chữ in đậm, kết quả hiển thị là giống nhau.

* Thẻ <b> sử dụng khi đơn giản là muốn gây sự chú ý vào một đoạn nội dung
* Thẻ <strong> sử dụng khi muốn thể hiện sự quan trọng, nghiêm trọng, khẩn cấp, v.v

## 4.Thẻ i và thẻ em (Cả thẻ <i> và <em> đều được trình duyệt hiển thị nội dung dưới dạng chữ in nghiêng.)

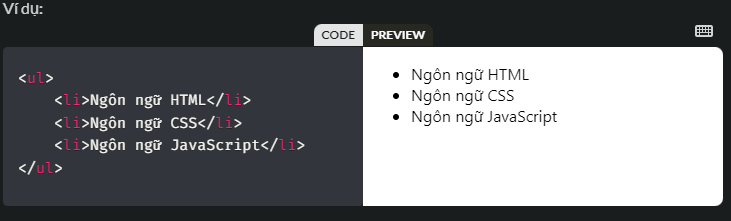
* Thẻ <i> mang ngữ nghĩa thể hiện các thuật ngữ chuyên ngành, tên một tác phẩm, trích dẫn một câu nói, nhắc tới một ngôn ngữ khác trong đoạn văn, v.v.
* Thẻ <em> mang ngữ nghĩa thể hiện việc nhấn mạnh nội dung (nhấn mạnh trọng âm, điều này hoạt động trên cả trình đọc màn hình), v.v

## 5.Thẻ u (chữ gạch chân)

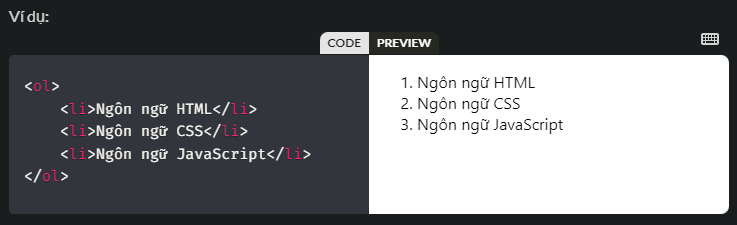
## 6.Thẻ a (liên kết)

## 7.Thẻ img (hình ảnh)

## 8.Thẻ ul, li (tạo danh sách không có số thứ tự, chỉ hiển thì chấm tròn đầu câu)



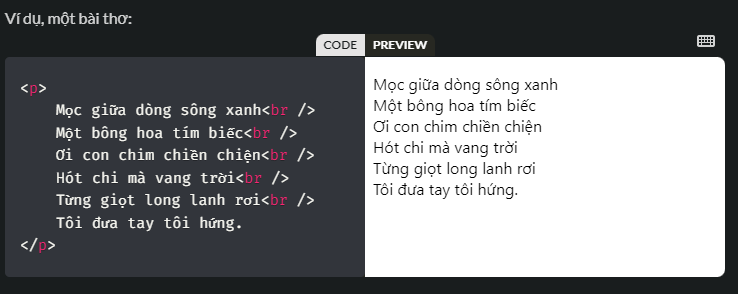
## 9.Thẻ ol, li (tạo danh sách có số thứ tự, hiển thị số ở đầu câu)



Ngoài ra, thẻ <ol> có thêm một số thuộc tính:

* reversed để đảo ngược chiều đếm số. Thay vì 1, 2, 3 sẽ thành 3, 2, 1.
* start đặt giá trị bắt đầu, mặc định là 1.
* type đặt kiểu cho marker:
* 1 số tự nhiên
* a bảng chữ cái, chữ thường
* A bằng chữ cái, chữ in hoa
* i số La Mã, chữ thường
* I số La Mã, chữ in hoa

## 10.Thẻ br (Thẻ ngắt dòng (Line Break element) trong HTML là <br />, được sử dụng để tạo ra một dòng mới trong một đoạn văn. Đây là một thẻ tự đóng.)



# Phần 3. Sử Dụng Liên Kết

## 1. Thuộc tính target

Thuộc tính target sử dụng trong thẻ <a> quyết định trang web được mở từ trang hiện tại hoặc mở trong một tab mới của trình duyệt.

Có 2 giá trị được sử dụng phổ biến:

* \_self là mở liên kết từ trang hiện tại (mặc định)
* \_blank là mở liên kết trong tab mới

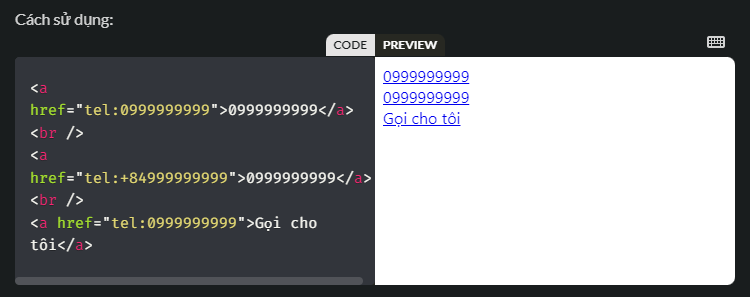
- Trường hợp sử dụng target="\_blank" hiệu quả nhất đó là khi tránh làm mất dữ liệu của người dùng hoặc mở liên kết khác tên miền.

- Để tối ưu hơn về hiệu năng và bảo mật, hãy thêm rel="noopener" vào toàn bộ các thẻ a có target="\_blank".

## 2. Phương thức tel, mailto

Sử dụng 2 phương thức trong thuộc tính href để làm điều này:

* mailto để liên kết tới một email. Ví dụ: mailto:example@gmail.com
* tel để liên kết tới một số điện thoại. Ví dụ: tel:0999999999 hoặc <tel:+84999999999>



Với mailto, bạn có thêm một danh sách các tham sau:

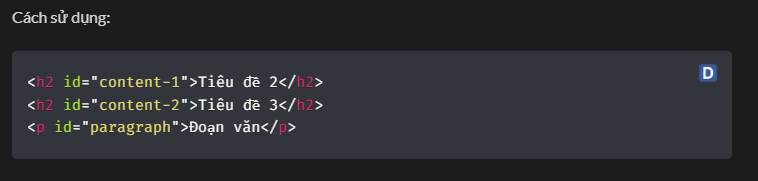
* cc: danh sách email sẽ nhận được bản sao của thư (không bắt buộc).
* bcc: danh sách email ẩn sẽ nhận được bản sao của thư (không bắt buộc).
* subject: chủ đề của thư (không bắt buộc).
* body: nội dung của mail (không bắt buộc).
* ?: dấu phân cách giữa phần mailto và tham số (không bắt buộc).
* &: dấu phân cách tham số (không bắt buộc).

## 3. Thuộc tính href với hash và id

### -hash: Hash (hash mark - dấu thăng #) trong URL trỏ trình duyệt tới một vị trí cụ thể trong trang web. Thông thường, bạn sẽ thấy # được sử dụng ở phần sau của một URL.



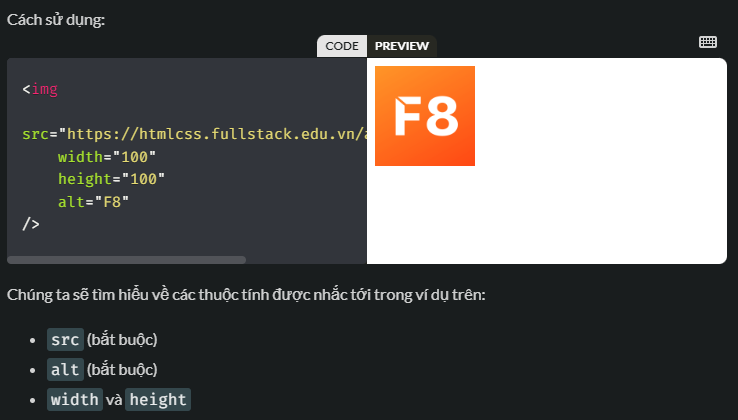
- ID (identifier - định danh) là một thuộc tính để tạo định danh duy nhất cho một phần tử HTML trong cùng một trang (theo tiêu chuẩn của W3C).



# Phần 4. Sử Dụng Hình Ảnh

## 1.Thẻ img

Thẻ <img> (image - hình ảnh) được dùng để hiển thị hình ảnh trên trang web. Đây là một thẻ tự đóng (self-closing tag)



## 2. Thuộc tính alt khi sử dụng thẻ img

Thuộc tính alt (alternative - thay thế) dùng để chỉ định văn bản thay thế trong trường hợp hình ảnh không hiển thị.



## 3. Các định dạng ảnh

### # Định dạng JPG/JPEG

JPG (tên gốc là JPEG - Joint Photographic Experts Group) là một trong số định dạng ảnh bitmap được sử dụng phổ biến nhất trên các trang web, chiếm khoảng 75% trong số các trang web có sử dụng hình ảnh.

-Ưu điểm:

* Mức độ nén cao
* Được hỗ trợ trong mọi trình duyệt
* Hỗ trợ hiển thị hàng triệu màu sắc

-Nhược điểm:

* Hình ảnh có thể bị vỡ
* Không hỗ trợ nền ảnh trong suốt
* Chất lượng bị giảm khi độ nén cao
* Không hỗ trợ ảnh động

-Dùng khi:

* Hình ảnh trong nội dung bài viết (blog, bài post trên Facebook, v.v)
* Ảnh quảng cáo, banner không đòi hỏi ảnh trong suốt

### # Định dạng PNG

PNG (Portable Network Graphics) cũng là một trong số định dạng ảnh bitmap được sử dụng phổ biến nhất trên các trang web, chiếm khoảng 80% trong số các trang web có sử dụng hình ảnh.

PNG hỗ trợ ảnh trong suốt:



-Ưu điểm:

* Chất lượng hình ảnh không đổi khi bị nén
* Được hỗ trợ trong mọi trình duyệt
* Hỗ trợ hiển thị hàng triệu màu sắc
* Hỗ trợ ảnh trong suốt

-Nhược điểm:

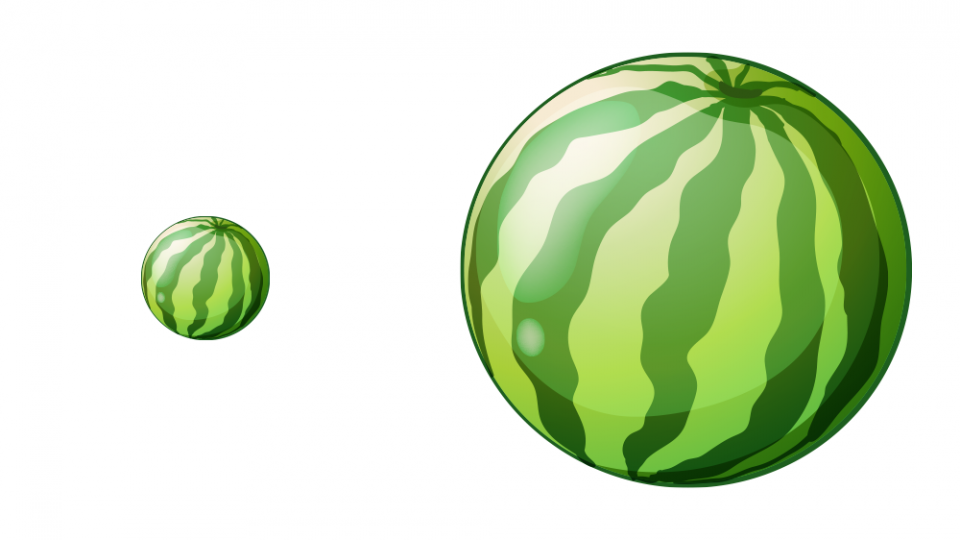
* Hình ảnh có thể bị vỡ
* Không hỗ trợ ảnh động

-Dùng khi:

* Ảnh trên web yêu cầu độ trong suốt (logo, biểu tượng, v.v)
* Các hình ảnh có kết hợp màu phức tạp như màu chuyển
* Các hình ảnh cần chỉnh sửa lại nhiều lần

### # Định dạng SVG

SVG (Scalable Vector Graphics) là một định dạng vector, không phải là dạng bitmap như các định dạng đã đề cập. Sự khác biệt dễ nhận thấy là dạng vector sẽ không bị vỡ ảnh khi phóng to. SVG cũng là một định dạng được sử dụng phổ biến bởi các trang web có sử dụng hình ảnh, chiếm khoảng 51%.



-Ưu điểm:

* Dung lượng có thể nhỏ
* Dễ dàng chỉnh sửa bằng CSS
* Phóng lớn mà không bị vỡ ảnh
* Hỗ trợ ảnh động
* Hỗ trợ ảnh trong suốt
* Được hỗ trợ trong mọi trình duyệt

Nhược điểm:

* Dung lượng tăng cao nếu hình có nhiều họa tiết khác nhau, phức tạp
* Không thể đọc một phần của định dạng này, phải chờ tải xong hoàn toàn

-Dùng khi:

* Logo, icon với thiết kế dạng hình học, vector
* Hình ảnh cần đáp ứng nhiều kích thước khác nhau

### # Định dạng GIF

GIF (Graphic Interchange Format) cũng là dạng bitmap như JPEG và PNG. Trong số các trang web có sử dụng hình ảnh thì có khoảng 21% trang web có sử dụng định dạng GIF.



--Ưu điểm:

* Dung lượng nhỏ
* Hỗ trợ ảnh trong suốt
* Hỗ trợ ảnh động
* Được hỗ trợ trong mọi trình duyệt

-Nhược điểm:

* Màu bị hạn chế
* Hình ảnh có thể bị vỡ
* Không chỉnh sửa được ảnh động
* Ảnh động có thể chưa chuyển động khi chưa tải xong

-Dùng khi:

* Khi cần các hình ảnh động

### # Định dạng WebP

Hình ảnh WebP là một định dạng hình ảnh bitmap do Google tạo ra, đặc biệt dành cho việc sử dụng web. Đây là một định dạng chưa phổ biến trên các trang web, chiếm khoảng 5% trong số các trang web có sử dụng hình ảnh.

-Ưu điểm:

* Dung lượng nhỏ hơn PNG, JPEG
* Hỗ trợ ảnh trong suốt
* Hỗ trợ ảnh động

-Nhược điểm:

* Hình ảnh có thể bị vỡ
* Chưa được hỗ trợ bởi mọi trình duyệt
* Giảm chất lượng khi chuyển đổi qua

### # Mã hóa Base64

Đây không phải là một định dạng ảnh. Base64 là một chương trình mã hóa chuỗi ký tự.

-Ưu điểm:

* Ảnh đi theo code HTML, CSS (không phải tải qua URL)
* Giảm số lượng kết nối, tăng tốc độ tải trang

-Nhược điểm:

* Không tái sử dụng được như URL
* Khi ảnh lớn thì chuỗi mã hóa lớn, gây nặng file HTML, CSS
* Hình ảnh không được công cụ tìm kiếm đánh chỉ mục

-Dùng khi:

* Muốn ảnh đi kèm với code HTML, không phải tải qua một URL

## 4. Sử dụng hình ảnh tối ưu

### # Tối ưu dung lượng ảnh cho trang web

Khi đã tối ưu về kích thước ảnh, nếu sử dụng định dạng PNG bạn vẫn có thể tiếp tục sử dụng **tinypng.com** để tối ưu dung lượng hình ảnh.

Sử dụng công cụ sau: <https://tinypng.com>



Cá nhân mình đã sử dụng trang web này trong khoảng 7-8 năm trở lại đây. Mỗi khi mình nhận được ảnh thiết kế từ designer dưới dạng PNG là sử dụng trang này. Mỗi lần tối ưu giảm được từ 50% - 75% dung lượng hình ảnh mà vẫn đảm bảo về mặt hiển thị.

## 5.Liên kết và hình ảnh

### # Thêm liên kết vào hình ảnh

Để thêm liên kết cho hình ảnh bạn chỉ cần sử dụng một thẻ <a> bao bọc lấy hình ảnh.



### # Sử dụng nguồn ảnh miễn phí

Tại đây là địa chỉ các trang web cung cấp ảnh chất lượng và miến phí. Danh sách dưới đây vẫn có thể được cập nhật thêm trong tương lai.

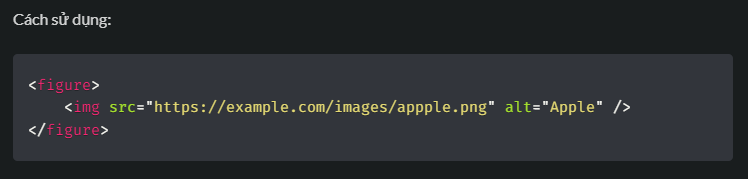
-Nguồn ảnh cực đẹp và chất:

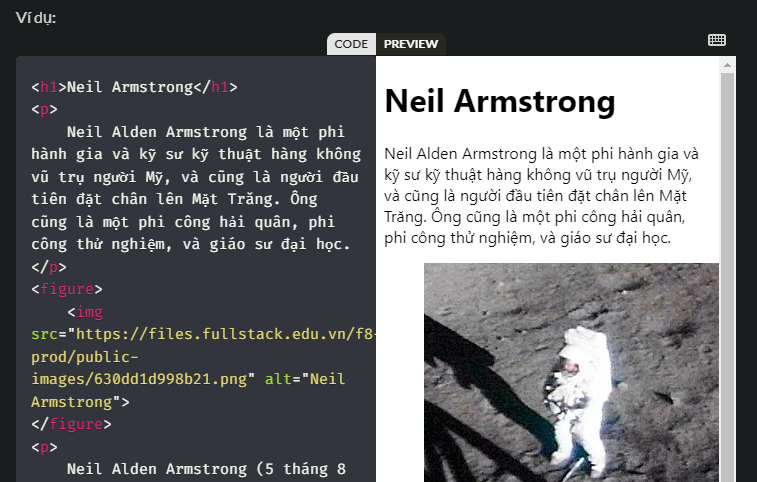
* <https://unsplash.com>
* <https://www.pexels.com>
* <https://drawkit.com>
* <https://undraw.co/illustrations>

## 6.Sử dụng hình ảnh trong html5

### # Thẻ figure

Thẻ <figure> được sử dụng để thể hiện hình ảnh trong nội dung tài liệu như hình minh họa, biểu đồ, khối code, v.v. Đây là một thẻ có từ phiên bản HTML5.



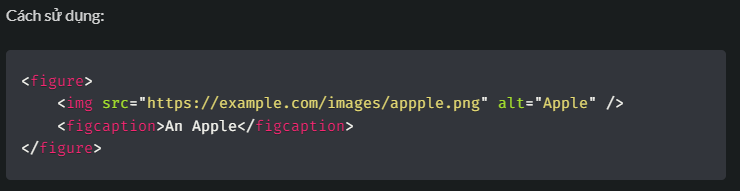


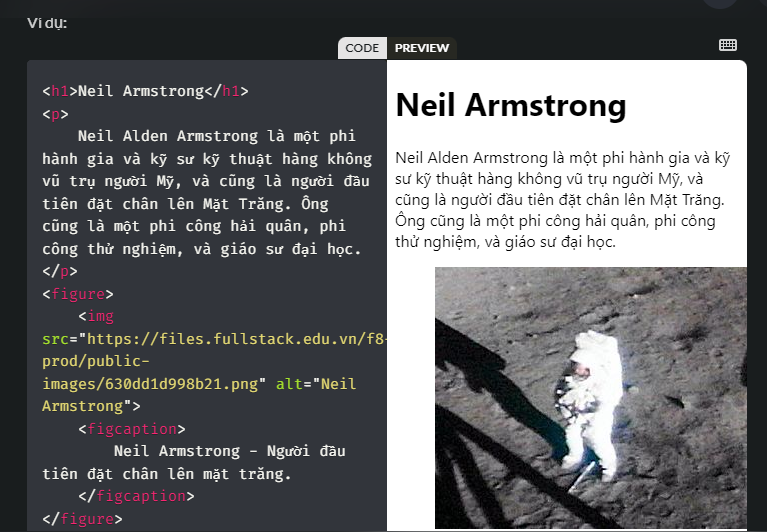
### # Thẻ figcaption

Thẻ <figcaption> dùng để bổ sung chú thích cho nội dung được thể hiện bằng thẻ <figure>. Ví dụ như chú thích cho một hình ảnh, video, biểu đồ, v.v.

-Một số lưu ý khi sử dụng thẻ <figcaption>:

* Không bắt buộc sử dụng
* Luôn nằm trong thẻ <figure>
* Đặt trước hoặc sau thẻ <img> đều được





# Phần 5. Tra Cứu Thẻ HTML

## 1.HTML Reference

Truy cập: <https://htmlreference.io>

Mục đích sử dụng:

* Tra cứu loại thẻ
* Tra cứu thuộc tính

## 2.Sử dụng W3school

Trang web w3schools.com là nơi trực tuyến về các công nghệ web. Điểm mạnh là nội dung ngắn gọn, dễ hiểu và đi kèm nhiều ví dụ.

Khi nào thì sử dụng W3Schools?

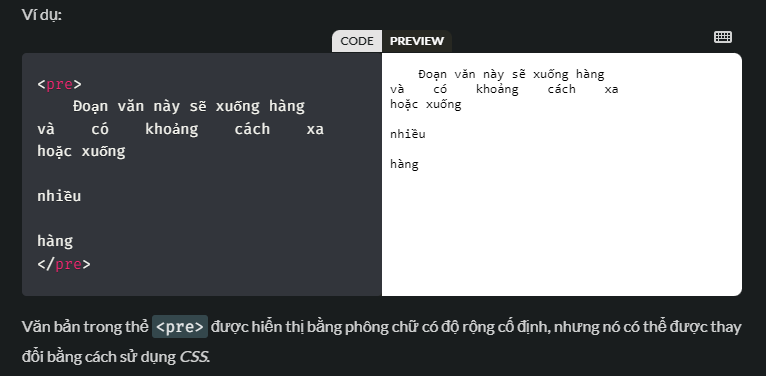
* Làm nguồn tự tham khảo thêm
* Tra cứu các thẻ HTML (bao gồm cả các thẻ đã lỗi thời)
* Tra cứu cách dùng các thuộc tính, thuộc tính nào thuộc thẻ nào, v.v

Các liên kết thường sử dụng để tra cứu HTML trên W3Schools:

* Học qua tutorial: <https://w3schools.com/html/default.asp>
* Tra cứu các thẻ: <https://w3schools.com/tags/default.asp>
* Tra cứu các thuộc tính: <https://w3schools.com/tags/ref_attributes.asp>
* Tra cứu các thuộc tính toàn cục: <https://w3schools.com/tags/ref_standardattributes.asp>

# Phần 6. Comments và blank space

## 1.Thẻ pre (giữ nguyên đoạn văn bản, dầu cách, dấu tab đều giữ nguyên)



## 2. HTML Entities

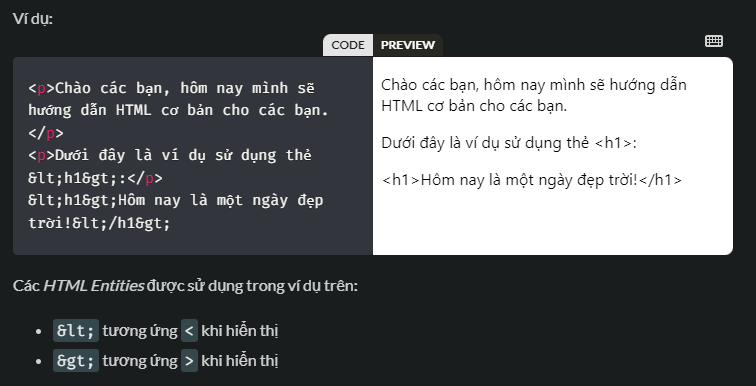
HTML Entities là các chuỗi bắt đầu bằng dấu & và kết thúc bằng dấu ;

Các bạn có thể tham khảo:

HTML Entities thông dụng (w3.org: <https://www.w3.org/wiki/Common_HTML_entities_used_for_typography>)

HTML Entities thông dụng (w3schools.com: <https://www.w3schools.com/html/html_entities.asp>)

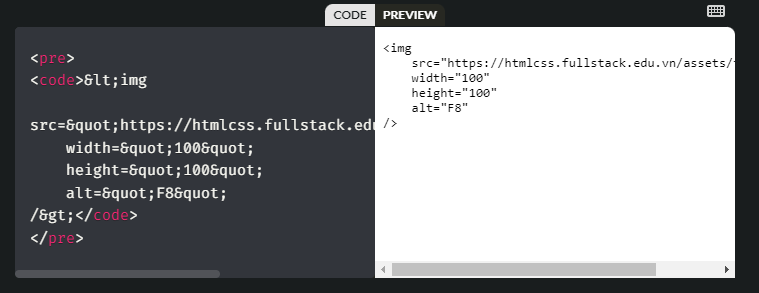
HTML Entities trong HTML4, HTML5. Link: <https://www.w3schools.com/charsets/ref_html_entities_a.asp>



## 3.Thẻ code (dùng để hiển thị đoạn code, được bao bọc bởi thẻ pre và sử dụng Entities để show những thẻ code)

Công cụ chuyển HTML sang HTML Entities (HTML Escape):

* <https://codebeautify.org/html-escape-unescape>
* <https://www.freeformatter.com/html-escape.html>



# Phần 7. Tính Hợp Lệ Của HTML

## 1.Thẻ tùy chỉnh

### Index.html

<!DOCTYPE html>

<html lang="vi">

<head>

    <meta charset="UTF-8">

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

    <title>Ví dụ thẻ tùy chỉnh</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap" rel="stylesheet">

</head>

<body>

    <!-- Ví dụ 1 -->

    <app-modal heading="Chào mừng" label="Xem lời chào">

        <p>Chào mừng các bạn đã gia nhập F8!</p>

    </app-modal>

    <!-- Ví dụ 2 -->

    <app-modal heading="Thông báo" label="Xem thông báo">

        <p>Khóa học này đã có sẵn hơn 200 bài học (bao gồm video, bài tập, thử thách, flashcards, v.v).</p>

    </app-modal>

</body>

</html>

### Main.js

/\*\*

 \* Các kiến thức sử dụng trong file này

 \* các bạn sẽ được học trong khóa JavaScript Pro

 \* sẽ được ra mắt trong đầu năm 2023.

 \*/

import { LitElement, html, css } from 'lit';

const styles = css`

    :host {

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

    }

    \* {

        box-sizing: border-box;

    }

    .wrapper,

    .overlay {

        position: fixed;

        inset: 0;

    }

    .wrapper {

        display: flex;

        z-index: 100;

    }

    .overlay {

        background-color: rgba(0, 0, 0, 0.6);

        animation: fade-in 0.3s;

    }

    header h1 {

        font-size: 24px;

        font-weight: 700;

    }

    .container {

        position: relative;

        z-index: 1;

        min-width: 320px;

        max-width: 80%;

        margin: auto;

        padding: 0 20px 20px;

        border-radius: 4px;

        background-color: #fff;

        animation: fade-in 0.3s, grow-up 0.3s;

    }

    .wrapper.closing {

        pointer-events: none;

    }

    .wrapper.closing .overlay {

        animation: fade-out 0.3s forwards;

    }

    .wrapper.closing .container {

        animation: fade-out 0.3s forwards, grow-down 0.3s forwards;

    }

    .content {

        font-size: 16px;

        line-height: 1.6;

    }

    footer {

        display: flex;

        justify-content: flex-end;

        margin-top: 32px;

    }

    button {

        border: none;

        padding: 6px 16px;

        border-radius: 4px;

        color: #fff;

        background: linear-gradient(141.34deg, #5ebbff 0%, #a174ff 91.11%)

            border-box;

        font-size: 16px;

        font-weight: 600;

        cursor: pointer;

        user-select: none;

    }

    @keyframes fade-in {

        from {

            opacity: 0;

        }

        to {

            opacity: 1;

        }

    }

    @keyframes fade-out {

        from {

            opacity: 1;

        }

        to {

            opacity: 0;

        }

    }

    @keyframes grow-up {

        from {

            transform: scale(0.5);

        }

        to {

            transform: scale(1);

        }

    }

    @keyframes grow-down {

        from {

            transform: scale(1);

        }

        to {

            transform: scale(0);

        }

    }

`;

class AppModal extends LitElement {

    constructor() {

        super();

        this.\_show = false;

        this.\_closing = false;

    }

    static get styles() {

        return styles;

    }

    static get properties() {

        return {

            heading: { type: String },

            label: { type: String },

        };

    }

    get \_modal() {

        return this.renderRoot.querySelector('#modal');

    }

    \_toggle = () => {

        if (!this.\_show) {

            this.\_show = true;

            this.requestUpdate();

        } else {

            this.\_modal.classList.add('closing');

            this.\_modal.querySelector('.container').addEventListener(

                'animationend',

                () => {

                    this.\_show = false;

                    this.requestUpdate();

                },

                { once: true },

            );

        }

    };

    render() {

        return html`

            ${this.\_show

                ? html`

                      <div id="modal" class="wrapper">

                          <div class="overlay" @click="${this.\_toggle}"></div>

                          <div class="container">

                              <header>

                                  <h1>${this.heading}</h1>

                              </header>

                              <div class="content">

                                  <slot></slot>

                              </div>

                              <footer>

                                  <button @click="${this.\_toggle}">

                                      Đóng lại

                                  </button>

                              </footer>

                          </div>

                      </div>

                  `

                : ''}

            <button @click="${this.\_toggle}">${this.label}</button>

        `;

    }

}

if (!customElements.get('app-modal')) {

    customElements.define('app-modal', AppModal);

}