# What is HTML?

HTML là viết tắt của Hyper Text Markup Language. Nó không phải là ngôn ngữ lập trình mà chỉ là ngôn ngữ đánh dấu. Nhiệm vụ của nó sẽ đánh dấu từng phần của văn bản, đại loại như dán nhãn các dòng chữ trong văn bản để các ngôn ngữ lập trình nhận diện và xử lý. Nó còn là khung sườn cho trang Web khi biểu diễn các phần của trang Web một cách có cấu trúc và logic hơn.

# Tag - Element

Chúng ta xây dựng và điều khiển các trang HTML từ các **tag**, là các thẻ mà mang một một chức năng nào đó cụ thể góp phần xây dựng nên khung sườn của trang Web. Các tag có cấu trúc như sau:

<tag name> content </tag name>

Cả phần lớn này lần lượt là **opening tag**, **content** bên trong tag và **closing tag**. Ba phần này hợp lại tạo thành một **element**, là các phần tử bên trong HTML. Một số tag không có nội dung và cũng có một số tag không có opening tag hoặc closing tag, ta gọi chúng là **single tag** hay là **self-contained tag**, tức là chúng chứa toàn bộ nội dung trong một tag duy nhất mà không cần có content ở giữa để hiển thị.

Một số tag điển hình, ví dụ như là **paragraph tag**, dùng để viết đoạn văn bản bên trong.

<p> Paragraph text </p>

Hay **anchor tag**, dùng để chèn link tới một trang Web khác.

<a> Link to another page </a>

Ngoài ra còn **heading tag**, dùng để đánh dấu tiêu đề trong dòng văn bản.

<h1> Headings </h1>

Và còn có **blockquote tag**, tag này chứa một câu quote nào đó, sẽ được đặt bên trong dấu ngoặc kép.

<blockquote> Quotes </blockquote>

Cuối cùng là **image tag**, dùng để đánh dấu một hình ảnh.

<img>

# Attribute

Thêm vào đó, các tag cũng còn có thứ bao gồm bên trong nó là các **attribute,** chúng là các thuộc tính được thêm vào để thể hiện thêm một số thông tin của cái tag đó. Một số tag có một số attribute nhất định, một số attribute có thể có ở mọi tag, gọi là **global attribute.**

Ví dụ như là **class** **attribute**, đây là một global attribute có tác dụng đánh dấu một lớp các đối tượng để CSS (Cascading Style Sheet – Dùng để tạo style cho HTML) có thể thao tác một lúc nhiều element. Các attribute được thể hiện bằng tên attribute, theo sau đó là dấu bằng và giá trị của chúng nằm trong dấu ngoặc kép.

<h1 class="Header"> Headings </h1>

Chúng ta cũng có thể có nhiều attribute trong một tag, phân biệt nhau bởi dấu cách. Ví dụ dưới đây là **style** **attribute** được thêm vào. Attribute này có tác dụng thêm CSS vào tag trong cùng một dòng, đây là inline style, cách này ngày nay không được ưa chuộng do người ta ưu tiên tách CSS ra một file riêng (outline style).

<h1 class="Header" style="color:red"> Headings </h1>

# Nested Tag

Các tag có thể lồng nhau (nested), ví dụ ta thêm **emphasize tag** vào bên trong **header tag** ở trên, làm cho một đoạn trong header được nhấn mạnh.

<h1> <em>Headings</em> line </h1>

Kết quả ở trang Web:

A white and black sign

Description automatically generated with medium confidence

Và các tag cứ thế có thể tiếp tục lồng nhau. Dưới đây ta thêm vào **strong tag** để tô đậm (bold) dòng chữ.

<p> <em>Headings <b>line</b> </em> </p>

Kết quả ở trang Web:

Text

Description automatically generated

# Web page structure

Chúng ta sẽ bắt đầu xây dựng một trang Web bằng cách tạo ra một file có tên là **index.html**. Sở dĩ chọn tên index vì nó là file chỉ mục, lúc mà Web browser quét qua toàn bộ các trang Web thì nó sẽ tìm index.html và xem như là homepage (trang chủ) của Web. Đồng thời file HTML có đuôi là “.html”.

Một dòng code mà mọi file HTML đều có là:

<!DOCTYPE html>

Dòng này có chức năng nói với Web browser rằng: “Ê, tao là tài liệu HTML đây, dưới đây tao sẽ trình bày một đống ngôn ngữ siêu văn bản đó! Hãy duyệt tao dưới dạng HTML nha”.

Tiếp theo, ta có thể có một minh họa tương đồng giữa miếng bánh sandwich và tập tin HTML. Tức là các tag lồng nhau như các tầng của bánh, và cái miếng kẹp ở ngoài cùng chính là **html tag**. Tag này sẽ chứa toàn bộ phần lõi bên trong của cái bánh sandwich, là toàn bộ nội dung của trang Web mà chúng ta đang tạo.

Bên trong html tag thì đầu tiên chúng ta sẽ đi đến **head tag**, đây là một tag được ví như sốt mayonnaise, nó không góp phần nhiều vào cấu trúc của bánh nhưng làm cho miếng bánh ngon lành hơn. Nhiệm vụ của head tag là đưa thêm thông tin về trang Web cho Web Browser nhận biết. Những thông tin này sẽ không hiển thị cho người dùng.

Trong head tag chính là một số thông tin extra, đại loại như tiêu đề của cái tab hiển thị trên Web Browser. Tiêu đề này sẽ nằm trong **title tag**. Ngoài ra còn có khai báo keyword cho trang Web để các Search Engine như Google biết mà hiển thị trang Web của chúng ta cho người dùng (cách này đã cũ). Keyword này nằm trong **meta tag** – một single tag, meta tag này còn chứa thuộc tính có tên là **charset**, giá trị “UTF-8”. Ngoài ra meta tag còn có chức năng tham chiếu đến các file CSS và JS bên ngoài.

<!DOCTYPE html>

<html>

    <head>

        <meta name="keywords" content="Bán cá"/>

        <meta charset="UTF-8" />

        <title>Cửa hàng hải sản</title>

    </head>

</html>

Sau khi xong phần trang trí bên ngoài thì tới phần thịt, rau, trứng bên trong cái bánh sandwich. Phần này chính là phần thân của toàn bộ trang Web, nằm trong **body tag**. Trong body tag ta sẽ sử dụng mọi tag có thể dùng để trình bày và đánh dấu văn bản. Các nội dung bên trong phần body sẽ được hiển thị trên trang Web.

<!DOCTYPE html>

<html>

    <head>

        <meta name="keywords" content="Bán cá"/>

        <meta charset="UTF-8" />

        <title>Cửa hàng hải sản</title>

    </head>

    <body>

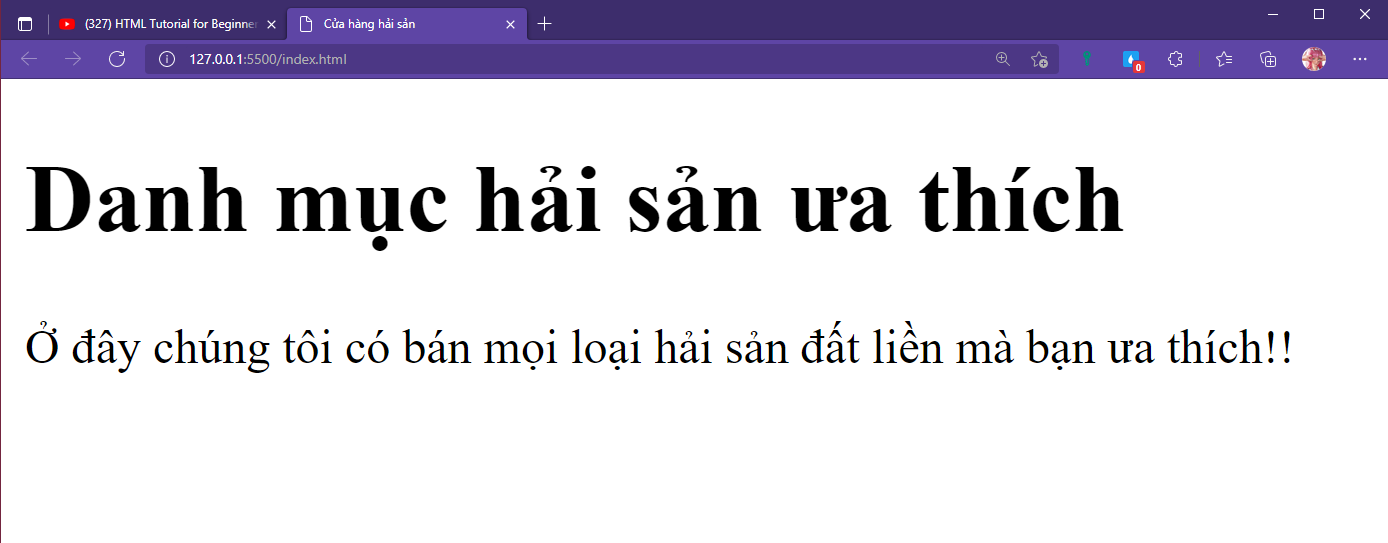
        <h1>Danh mục hải sản ưa thích</h1>

        <p>Ở đây chúng tôi có bán mọi loại hải sản đất liền mà bạn ưa thích!!</p>

    </body>

</html>

Kết quả trang Web:



# Headings and Text

Các heading tag, có thể có từ h1 đến h6, sẽ là các phân cấp tiêu đề từ lớn đến nhỏ.

<body>

        <h1>Chào mừng đến với trang bán hàng uy tín nhất hành tinh</h1>

        <p>Ở đây chúng tôi có bán mọi loại hải sản đất liền mà bạn ưa thích!!</p>

        <h1>Bán cá</h1>

        <h2>Bán mực</h2>

        <h3>Bán tôm</h3>

        <h4>Bán cua</h4>

        <h5>Bán sò</h5>

<h6>Bán ghẹ</h6>

</body>

Cần lưu tâm một điều là các heading tag nên được chọn từ 1 đến 6 sao cho phù hợp với mức độ quan trọng của header, tương tự như việc đánh dấu các section cho văn bản Word hiện tại này.

Tiếp theo ta sẽ đi qua một loạt các tag dùng trong văn bản, có thể tham khảo thêm trên mạng tùy nhu cầu. Đầu tiên là **small tag**, sẽ làm chữ nhỏ đi một chút. Tiếp theo là **mark tag**, highlight chữ. **Sub tag** và **sup tag**, tượng trưng cho subscript và superscript (chữ dịch xuống và lên). **Preformat tag**, để đánh dấu văn bản đã được format trước đó rồi, cần áp dụng những gì đã được format hiển thị lên Web Browser. Tức là nếu ta xuống dòng bên trong preformat tag thì Web Browser của ta cũng sẽ xuống dòng theo, trong khi bình thường thì không hoặc cần tag xuống dòng.

        <h1>Bán <small>cá</small></h1>

        <h2>Bán <mark>mực</mark></h2>

        <h3>Bán <sub>tôm</sub></h3>

        <h4>Bán <sup>cua</sup></h4>

        <h5>Bán

            sò</h5>

        <h5><pre>Bán

            sò</pre></h5>

<h6>Bán ghẹ</h6>

Hiển thị ở Web Browser (các Header bình thường và các Header được format bằng các tag đánh dấu đặc biệt).

Text

Description automatically generatedText

Description automatically generated

Nhận thấy cả hai dòng h5 đều cùng xuống hàng, nhưng dòng nào có preformat mới được hiển thị xuống hàng thật sự.

# Image tag

**Image tag**, là một tag hiển thị hình ảnh trên trang Web mà thường đi chung với **source attribute**. Attribute này chứa đường dẫn tới hình ảnh mà chúng ta cần chèn vào trang Web. Ngoài ra còn có **width & height attribute** để thiết lập kích thước cho ảnh, **alternative attribute** để hiển thị mô tả hình ảnh nếu như Web Browser chẳng may không hiển thị được hình ảnh. Trong code dưới đây thì width có giá trị là “100%”, tức là độ rộng hình ảnh sẽ được căn chỉnh bằng chính kích thước của cửa sổ Web Browser.

<img src="img/zero-two.jpg" alt="Hình ảnh zero-two" width="100%">

# Links

**Anchor tag** như đã nhắc ở trên là một tag để chèn liên kết vào trang Web. Trong tag này có **href attribute** có giá trị là một đường dẫn tới liên kết nào đó. Có hai loại liên kết là internal (nội) và external (ngoại). Như tên gọi thì internal sẽ liên kết đến trong cùng thư mục trang Web (ví dụ như các file HTML contact hay about), trong khi external sẽ mở hoặc đi đến một trang Web khác. Chúng ta còn có feature Download thì là liên kết đến các file, trong khi Anchor là điều hướng đến một section nào đó trong cùng trang Web (kiểu đi đến một cái bookmark trong PDF hay header nào đó trong Word).

Liên kết đến một trang Web khác cùng thư mục (internal):

<a href="contact.html">Inbox dark deal ngay!</a>

Lưu ý khi muốn tạo đường dẫn thoát ra khỏi thư mục nào đó, ta sử dụng “../” (hai dấu chấm và dấu slash).

Liên kết đến một trang Web khác bên ngoài (external):

<a href="https://www.youtube.com/">Mở Youtube chill ngay !!</a>

Và khi liên kết đến một hình ảnh nào đó, chỉ việc link đến hình ảnh đó là xong:

<a href="img/zero-two.jpg">Tải ngay ảnh hot !!</a>

Trong trường hợp văn bản trang Web quá dài, ta cần điều hướng đến một header hoặc section nào đó của trang Web một cách nhanh chóng, ta sẽ đánh dấu cái section cần điều hướng đó bằng **ID attribute**, rồi chúng ta liên kết đến ID đó bằng đường dẫn **“# + tên ID”**. Ví dụ section của ta là một heading có ID như sau:

<h2 id="02"> Hình ảnh zero-two full HD không che!</h2>

Chúng ta liên kết đến nó thông qua đường dẫn **“# + tên ID”**, tức là **“#02”**:

<a href="#02">, hoặc xem luôn cho máu</a>

Thì lúc chúng ta bấm vào dòng chữ “hoặc xem luôn cho máu” này, Web Browser sẽ điều hướng đi đến cái chỗ mà có đánh dấu ID được nhắc đến trong href attribute.

Ngoài ra chúng ta có thể dùng một loại giá trị là **“#top”** để teleport lên top ngay lập tức 😊

Để thuận tiện cho User, khi external linking, ta thường mở trang Web ở một tab mới hoặc cửa sổ mới để người dùng không đánh mất tiến trình ở trang Web hiện tại. Chúng ta có thể làm được điều đó bằng cách sử dụng **target attribute.** Giá trị của attribute này thường là **“\_blank”**, dùng để mở một tab hoặc cửa sổ mới. Ta sẽ thử áp dụng cho website linking và image linking ở trên:

<a href="https://www.youtube.com" target = "\_blank">Mở Youtube chill ngay!!</a>

<a href="img/zero-two.jpg" target ="\_blank">Tải ngay ảnh hot !!</a>

# Lists

Có ba loại danh sách trong HTML: danh sách không có thứ tự (unordered list), danh sách có thứ tự (ordered list) và danh sách định nghĩa (definition list).

Unordered list được đặt trong **ul tag** (“ul” stands for unordered list). Các dòng danh sách được đặt trong **li tag** (“li” stans for list item). Ordered list được đặt trong **ol tag** (“ol” stands for odered tag), các list item của ordered list cũng được đặt trong **li tag**. Cuối cùng là Definition list thường dùng để định nghĩa một danh sách các khái niệm, ví dụ như cái từ điển. Definition list đặt trong **dl tag**, kèm theo **dt tag** (definiton term) để hiển thị term và **dd tag** (definition description) để mô tả định nghĩa của term đó.

    <ul>

        <li>Bạch tuột</li>

        <li>Cá ngừ</li>

        <li>Cá mập</li>

        <li>Cá đuối</li>

        <li>Cá thu</li>

    </ul>

Hoặc:

    <ol>

        <li>Bạch tuột</li>

        <li>Cá ngừ</li>

        <li>Cá mập</li>

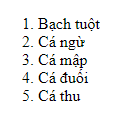
        <li>Cá đuối</li>

        <li>Cá thu</li>

    </ol>

Hình ảnh trên Web Browser:

Text

Description automatically generated

Các list item cũng có thể chứa các sub list:

<ol>

        <li>Bạch tuột</li>

        <li>Cá ngừ</li>

            <ol>

                <li>Cá ngừ một nắng</li>

                <li>Cá ngừ hai nắng</li>

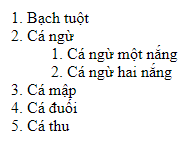
            </ol>

        <li>Cá mập</li>

        <li>Cá đuối</li>

        <li>Cá thu</li>

    </ol>



Definiton list sẽ kiểu như thế này:

        <dl>

            <dt>Dảk</dt>

            <dd>Đây là một từ tiếng anh, Dark, tức là tối. Nó bị viết theo TELEX nên trông hơi phèn. Thường dùng để biểu thị một vibe nào đó nó khá tối.

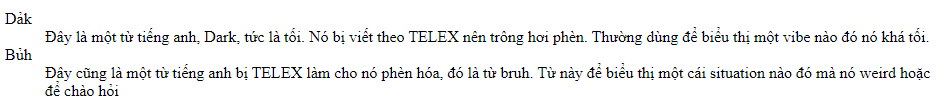
</dd>

            <dt>Bủh</dt>

            <dd>Đây cũng là một từ tiếng anh bị TELEX làm cho nó phèn hóa, đó là từ bruh. Từ này để biểu thị một cái situation nào đó mà nó weird hoặc để chào hỏi

           </dd>

        </dl>



# Hr and Br Tags

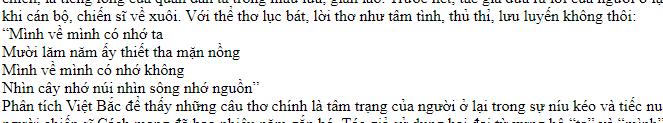
Để đoạn văn bản trong **paragraph tag** được xuống dòng, ta sử dụng **break tag**, là một single tag có tag name là br.

“Mình về mình có nhớ ta<br>

Mười lăm năm ấy thiết tha mặn nồng<br>

Mình về mình có nhớ không<br>

Nhìn cây nhớ núi nhìn sông nhớ nguồn”<br>



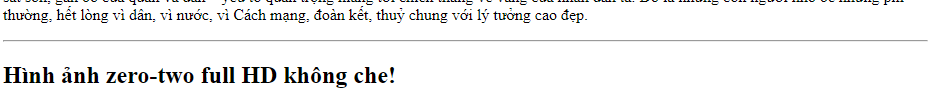
Còn để chia trang Web ra làm các phần riêng biệt rõ ràng, ta sử dụng một đường kẻ ngang giữa các phần. **Hr tag** sẽ có tác dụng tao ra dòng kẻ đó. Đây là đoạn code **hr tag** nằm giữa đoạn văn bản nào đó và một cái header.

        </p>

        <hr>

        <h2 id="02"> Hình ảnh zero-two full HD không che!</h2>

Hiển thị trên Web Browser:



# Div tag

Div stands for “division”, đây là một tag dùng để chia trang Web thành các section, tức là nhóm các phần khác nhau của trang Web thành các khối có trật tự và logic. **Div tag** có cách sử dụng rất đơn giản,chúng ta chỉ cần cho các section chúng ta muốn chia vào các div tag là xong. Về cơ bản sau khi thêm xong, sẽ không có cái gì xảy ra trên Web Browser. Như dưới đây ta đã chia phần chào mừng đến trang Web (Header) và phần chứa các link thành hai division khác nhau.

A screenshot of a computer

Description automatically generated with medium confidence

Tuy vậy kết quả hiển thị vẫn không có gì thay đổi.

Text

Description automatically generated

Nhưng mục đích của việc phân nội dung trang Web thành các division khác nhau là để sử dụng khi có ID, class và CSS. Lúc ấy, ta có thể áp dụng style cho nội dung trong từng division một cách dễ dàng và hữu hiệu. Nhìn chung, việc dùng **division tag** là để trang Web có được một cấu trúc bao gồm các phần hoặc block rõ ràng và dễ kết hợp với CSS.

# ID and Class

ID là một mã định danh độc nhất, chúng chỉ dùng cho một trang Web, ví dụ như Header, About, Contact,…

Class cũng là mã định danh cho các element (bao gồm tag và content của tag) mà có thể sử dụng nhiều lần trên một trang Web. Ví dụ như comment, ta có thể có nhiều comment.

Mã định danh (tạm dịch từ identifier). ID và Class sẽ là hai attribute cùng tên (**id attribute** và **class attribute**)

Cả ID và Class đều sử dụng cho CSS và JS.

Từ phần chia division ở trên, ta sẽ áp dụng ID cho hai division đầu là Header và Navigation (Phần chứa các link).

Text

Description automatically generated

Các phương thức contact thì ta có thể chia division và đặt class cho nó.

    <div *id*="contact">

        <h3> Liên hệ với chúng tôi tại </h3>

        <div *class*="contact-method">

            <h4>Qua email: </h4>

        </div>

        <div *class*="contact-method">

            <h4>Qua điện thoại: </h4>

        </div>

        <div *class*="contact-method">

            <h4>Qua fb: </h4>

        </div>

        <div *class*="contact-method">

            <h4>Qua tax: </h4>

        </div>

    </div>

Nói chung, việc thiết lập các ID và class là để sử dụng CSS và JS một cách thuận tiện hơn trong việc tùy chỉnh style và áp dụng các hàm lập trình.

# Adding CSS to HTML

Có ba cách để thêm CSS cho một trang HTML, nói cách khác là trang Web. Đầu tiên là inline, cách này sẽ liên kết đến CSS trong một tag nào đó của file HTML sử dụng **style attribute**. Cách thứ hai là dùng style tag ở đầu file HTML. Cách thứ ba là liên kết đến một file CSS bên ngoài.

Ở cách inline, chúng ta thêm vào **style attribute** – một global attribute cho các tag rồi sử dụng ngôn ngữ CSS bên trong giá trị của attribute đó. Giả dụ dưới đây chúng ta thêm vào màu đỏ cho dòng header đầu trang.

<h1 *style*="color:red">Trang chủ của shop bán hải sản uy tín nhất Đông Lào</h1>

Trong trường hợp cần add CSS cho lượng lớn dữ liệu thì cách này có vẻ khả thi. Chúng ta đi đến cách thứ hai là dùng **style tag** bên trong **head tag**.

    <head>

        <meta *name*="keywords" *content*="Bán cá"/>

        <meta *charset*="UTF-8" />

        <title>Cửa hàng hải sản</title>

        <style>

*#header*{

                color:red;

            }

        </style>

    </head>

Chẳng hạn ta cần add style cho lượng lớn các tag có ID **“header”** thì chúng ta chỉ cần reference đến nó bên trong **style tag**, mở ngoặc nhọn và bắt đầu viết CSS.

Tiếp theo, nếu chúng ta cần sử dụng CSS cho rất nhiều trang Web mà không chỉ đơn giản là lượng lớn các tag trong một trang Web duy nhất. Lúc này chúng ta cần dùng đến cách thứ ba, tức là liên kết với một file CSS bên ngoài.

Chúng ta sẽ sử dụng **link tag**, bao gồm **rel attribute** có giá trị là stylesheet, **type attribute** có giá trị là “text/css”, **href attribute** có đường dẫn đến file CSS mà chúng ta cần liên kết.

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

Tạm viết file CSS apply style cho h1 tag như sau:

h1{

    color: blue;

    text-decoration: underline;

    text-transform: uppercase;

}

Kết quả:



# Adding JS to HTML

Có vài cách để thêm file JS vào HTML. Cách thứ nhất là sử dụng **script tag** và viết các đoạn code JS bên trong đó. Trình biên dịch sẽ tìm kiếm chỗ nào có **srcipt tag** này và thực thi nó. **Script tag** có thể nằm ở bên trong **head tag** hoặc **body tag**. Cách thứ hai là liên kết đến một file JS bên ngoài có đuôi là “.js”. Ngoài ra cũng có thể thêm JS cho một tag nào đó (inline), nhưng không khuyến khích cách này.

Ở cách thứ nhất, compiler sẽ quét đến chỗ nào có JS rồi thực thi nó, khi nào kết thúc JS thì mới quét qua những phần dưới. Tức là nếu ta để JS ở phần head thì có thể trang Web sẽ không được hiển thị trước khi các lệnh JS hoàn thành nhiệm vụ của nó. Do đó mà ta nên có thói quen đặt JS cuối phần body của HTML, ngoại trừ một số trường hợp ngoại lệ mà JS có thể nằm ở đầu hoặc ở giữa.

Graphical user interface

Description automatically generated

Ví dụ ta có đoạn code thế này, ta sử dụng script trong head và gọi hai dòng lệnh in ra console “Hello World!” và hiện một cửa sổ pop up để thông báo dòng chữ “Hey kids, chào mừng đến với thiên đường hải sản”. Dưới đây là hình ảnh ở console.

Graphical user interface

Description automatically generated

Và hình ảnh pop up:

Graphical user interface, text, application

Description automatically generated

Cách thứ hai đó chính là liên kết với một file HTML bên ngoài. Chúng ta cũng sử dụng **script tag**, nhưng kèm theo đó là **source attribute** với giá trị là đường dẫn đến file JS.

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

Thì cũng tương tự như cách thêm JS ở trên, cách này cũng tuân theo thứ tự được quét đến của **script tag** trong file HTML.