**HTML TAG**

**HTML tags** are the hidden keywords within a web page that define how your web browser must format and display the content. Most **tags** must have two parts, an opening and a closing part. For example, <**html**> is the opening **tag** and </**html**> is the closing**tag**.

**Diferrence between block and inline tags?\**

**Display :Inline**

* Các items nằm trên cùng 1 dòng
* Chỉ chỉnh dc margin và padding left, right
* **Không thể set width và height.**

**Display : Block**

**a**

* Luôn đc xuống dòng nếu ko set width
* Các item có kiểu display: block sẽ set được width, height, margin, padding đầy đủ 4 hướng (top, bottom, right, left).

**Display : inline-Block(tạo thành thanh navbar)**

* Sẽ được sxep theo kiểu inline ( items xếp cùng 1 dòng)
* Các items sẽ có thuộc tính của display-block( width, height, margin, padding đủ 4 hướng)

**Tag attributes**

* Thuộc tính của một thẻ
* VD : thẻ <a> có thuộc tính “href”, img có thuộc tính src

**What is class? ID? Style?(.class -- #id)**

* Class : một lớp, các thẻ trong class dùng giống nhau
* ID : dùng 1 lần, Các thẻ html dùng 1 lần sẽ dc gắn id

**Insert comment**

* // ghi chus thich cho 1 dong
* <!--...--> 1 doan

**White-space**

* Dùng 1 khoảng trắng thay vì dấu cách nhiều lần, trong lập trình thì chỉ tính dấu cách đầu tiên( phải dùng ký tự thực thể &nbsp mới cách dc nhiều lần)

**HTML entities(ký tự thực thể)**

* Ký tự thực thể được sử dụng để hiển thị các biểu tượng, ký tự trong HTML.
* Dùng để hiển thị biểu tượng ký tượng mà ko bị nhầm lẫn với việc dùng 1 thẻ
* Có thực thể tên và thực thể số : VD : &nbsp;
* Thực thể tên dùng ký tự dạng chữ , thực thể số dùng ký tự dạng số

**Why are there both numerical and named character entity values?**

* Số dùng theo bảng mã ACII dựa trên bảng chữ cái La tinh dùng trong TA và các ngôn ngữ tây âu
* Tuy nhiên vì khó nhớ nên đã tạo ra ký tự bằng tên cho các nhà phát triển web

FORM

**What is form?** – là thẻ quan trọng để làm các thao tác dăng nhập, đăng ký, bình luận,( [Thẻ select](https://www.ducanhplus.com/thuc-hanh-voi-the-select-va-option-trong-html/), [radio button và checkbox](https://www.ducanhplus.com/the-radio-button-va-checkbox-trong-html/) để tạo các chọn lựa)

The HTML <form> element defines a form that is used to collect user input:

Form dùng để thu thập dữ liệu của người dùng vào input

**How many input types?**

* TextArea : Nhap du lieuvan ban
* Thẻ select, radiobutton và checkbox
* **Fieldset : Nhóm các trường liên quan trong form(dùng kèm tag legend- định nghĩa một chú thích), tạo đường bao quanh ngoài thẻ form như 1 border**
* Label : đặt tên các trường ếuu click chuột vào nhãn, sẽ đưa con trỏ chuột vào vùng <input />.

**Nút Submit**

* Dùng để tạo nút ấn

**Form action** : xác định trang web mà dữ liệu dc nhập vào được gửi đến và xử lý

**Form method** : xác định phương thức truyền

* Get : dùng khi truyền url như id….
* POST : Dùng để truyền các dữ liệu như mật khẩu tài khoản cần phải bảo mật

**Form validation:** việc kiểm tra form search, đặt điều kiện --- VD: form phải có từ 3 ký tự trở lên mới thoả mãn, form chỉ nhập số…..

* VD : Tên không được để trống
* Email đúng định dạng
* Username bao gồm các chữ cái và dấu gạch dưới, có độ dài 6-32 ký tự
* Password bao gồm các ký tự chữ cái, chử số, ký tự đặt biệt, độ dài 6-32 ký tự
* Radio giới tính yêu cầu chọn ít nhất
* Select chọn ngày, tháng, năm sinh yêu cầu chọn ít nhất một giá trị
* Có 2 phương pháp Validation

- Phía Client (jquery)

- Phía Server (Php)

HÀM HỖ TRỢ

* htmlspecialchars() : Chuyển đổi ký tự đặc biệt qua html
* trim() loại bỏ các ký tự dư thừa
* strlen() kiểm tra độ lớn chuổi
* empty() kiểm tra dữ liệu rỗng
* isset() kiểm tra tồn tại biến
* Is\_numeric () kiểm tra chuổi số,…
* Kiểm tra dữ liệu bằng biểu thức chính quy

CSS

* CSS is a language that describes the style of an HTML document.
* CSS describes how HTML elements should be displayed.

CSS Basic

**What are components of css style?( cấu tạo, hợp thanhfh )**

* Selector − an HTML tag to which the required style will be applied.
* Property - a type of attribute of an HTML tag. They can be color, size, border, etc.
* Value − assigned to properties. A color property can have a value, such as a specific color, size, border, etc. For example: red or #F1F1F1.

**How many way to define CSS?**

* **Inline** - by using the style attribute in HTML elements(dùng trực tiếp trong thẻ)
* **Internal** - by using a <style> element in the <head> section( dùng với 1 thẻ style )
* **External** - by using an external CSS file( tạo 1 file style mới)
* <link href="css/style.css" rel="stylesheet" type="text/css" media="screen,print" />

**Margin and Padding**

* padding là khoảng trống nằm giữa nội dung và viền. là khoảng cách so với thẻ con
* margin là khoảng trống nằm giữa viền và phần tử tiếp theo. là khoảng cách so với thẻ cha.

**Properties of**

* **display : xác định laoij hiển thị của thành phần ( như 1 khối, trong cùng 1 dòng , theo 1 danh sách, bảng…..)**
* display: block;
* display: inline;
* display: table;
* **position :** Thuộc tính position xác định loại của phương pháp định vị trí cho thành phần.( static : nằm theo thứ tự văn bản, relative : định vị tuyệt đói ví dụ cho 1 ảnh và 1 chữ, set relative cho ảnh thì nó sẽ di chuyển và đè lên dc chữ,
* absolute : định vị tuyệt đối theo thnhf phần bao ngoài viền ví dụ cho thẻ div có position relative thì thành phần ảnh img có thuộc tính absolute sẽ nằm trong relative mẹ mà ko thể tràn ra ngoài trừ khi set chỉ số âm,
* fixed : Fixed cũng hoạt động dựa vào một thẻ cha làm chuẩn bên ngoài. Chỉ có khác một điều là đối với Fixed thì khi bạn scrolling thì nó sẽ nằm luôn trên màn hình ngay tại vị trí chuẩn đó và đè lên trên các phần tử bên dưới.Thuộc tính position thường dùng kèm với các thuộc tính định vị trí: [left](https://hocwebchuan.com/reference/cssSection/pr_left.php), [right](https://hocwebchuan.com/reference/cssSection/pr_right.php), [bottom](https://hocwebchuan.com/reference/cssSection/pr_bottom.php), [top](https://hocwebchuan.com/reference/cssSection/pr_top.php)
* Z-**index : có 1 ảnh và 1 dòng chữ , ảnh set z-index là 20, chữ setz-index là 30 thì chữ sẽ đè lên trc còn ảnh sẽ in chìm xuống bên dưới chữ**
* **Thuộc tính overflow** xác định điều gì sẽ xảy ra nếu một thành phần box tràn nội dung.

1. Visible : text vẫn hiển thị tràn box
2. Hidden : text bị tràn bị dấu đi
3. Scroll : text bị tràn dấu đi và xuất hiện thanh scroll( cả scroll ngang và dọc)
4. Auto : giống scroll nhưng chỉ có dọc
5. Inherit : xác định thừa hưởng thành phần thuộc tính bê ngoài

* **Float** : huộc tính float xác định có hay không một thành phần được float (trôi nổi).( left right, none , inherit)
* **Flex :** là một kiểu dàn trang (layout mode)  mà nó sẽ tự cân đối kích thước của các phần tử bên trong để hiển thị trên mọi thiết bị. ( 1 div to bên ngoài, 1 div nhỏ bên trong chứ các items, chỉnh div ngoài với display : flex) ??????

**Khác nhau display none và visibility: hidden**

* Display none : ẩn đi và chỗ không gian của nó cũng bị mất( biến mất)
* visibility: hidden : ẩn đi nhưng không gian của nó vẫn ở đấy chiếm 1 chỗ( tàng hình)

**What canwe set in a background ?**

* **Background –attachment** : (fixed-scroll): Xác định thành phần nền được cố định hoặc cuộn so với trang.
* [background-color](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr02) : màu nền
* [background-image](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr03) ảnh nền
* [background-position](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr04): (left , top right bottom %, px) : vị trí hình ảnh nền thành phần
* [background-repeat](https://hocwebchuan.com/reference/cssSection/example/ex_bg.php#pr05): repeat x ( sử dụng giá trị này làm hình nền chỉ được lặp theo trục ngang. Repeat- y(sử dụng giá trị này làm hình nền chỉ được lặp theo trục dọc.), repeat( cả 2 trục) , no repeat( ko lặp lại ) : Xác định hình ảnh nền được lặp như thế nào

**How many ways we can define a color ?**

* **Color keywords : red, blue…..**
* **RGB**: là hệ màu với 3 channel Reg, Green, Blue( số từ 0-255) 0- 100%
* **RGBA**: là hệ màu với 4 channel Reg, Green, Blue và Alpha (0-255) hoặc 0% - 100%
* **HEXA** **Decimal**: là hệ màu sử dụng hệ số thập lục phân ( #0f012030)
* **HSL**: là hệ màu với 3 channel Hue, Saturation, Lightness( sử dụng theo bảng màu hình tròn có trục quay) cũng viết số
* **HSLA**: là hệ màu với 4 channel Hue, Saturation, Lightness vaf Alpha
* **Trong đó channel Alpha là channel giúp cho màu sắc trở nên trong suốt hơn (có thể nhìn thấy được phần tử bên dưới). Channel alpha có giá trị từ 0 – 1**

**CSS Unit Unit trong CSS là đơn vị đo của các thuộc tính trong CSS như margin, padding, width, height...**

Có 2 đơn vị ablsolute và relative

* Ablsolute : tuyệt đốit ( cm, mm, in: inch, px, pt, pc)
* Relative:tương đối ( thường dùng %)( em khoảng cách giữa các dòng): Tương đối so với  font-size của phần tử cha.)

**Short hand properties**

* sử dụng các thuộc tính CSS viết tắt bạn có thể làm giảm kích thước của tài liệu CSS của bạn nhiều hơn và giảm thiểu thời gian code hơn.
* VD : background: #fff url(image.gif) no-repeat top left
* Thay vì : background-color: #fff;  
  background-image: url(image.gif);  
  background-repeat: no-repeat;  
  background-position: top left

**The *cursor* property of CSS allows you to specify the type of cursor that should be displayed to the user.**

* Auto : tự thiết lập con trỏ chuột
* Crosshair : chuột dạng crosshair
* E-resize
* Help
* Move
* n-resize
* ne-resize
* nw-resize
* pointer
* progress
* s-resize
* se-resize
* sw-resize
* text
* w-resize
* wait
* inherit

**CSS FONT**

**Face of a form( dinh dang thuoc tinh form)**

* font-family : xac định tên form
* font-style : loại form
* font-weight :bold; xác định form đậm hay ko
* src : đường dẫn
* unicode-range : xác định phạm vi các ký tự unicode dc hỗ trợ
* letter spacing : khoảng cách các ký tự
* line-height : khoảng cách dòng
* text – decoration : xác định chữ có gạch dưới ko
* word – spacing : khoảng cách chữ
* font- stretch : xác định văn bản rộng hơn hay hẹp hơn(chưa hỗ trợ bởi trình duyệt)

**which properties make a font italic(nghiêng) or oblique( nghiêng nhẹ sang phải)**

* Font-style

**Increase or decrease a bold light**

* Font-weight : bold;
* Font-weight : 400;

CSS TEXT

**-text aligin ( căn văn bản)**

**-text-decoratinon : underline ( trang trí )**

**-text-transform: thiết lập các ký tự viết hoa cho văn bản.**

* None : tra về gtri ban đầu
* Capitalize : chữ đầu tien mỗi thành phần viets hoa
* Uppercase : tất cả các chữ viết hoa
* Lowercase : tất cả thành chữ thường
* Inherit : xác định thưởng hưởng thuoojctinhs từ thành phần cha

**Text – shadow** : thuoojjc tính đổ bóng thành phần

* VD : text-shadow : 1px 2px 3px #C91923;

**Tiền tố moz và webkit**

Vendor prefix là tiền tố được thêm vào bởi nhà cung cấp trình duyệt:

* -webkit là của Chrome và Safari
* -moz là của Firefox
* -o là của Opera
* -ms là của Internet Explorer  
  Những tiền tố này giúp render ra những tính năng CSS mới
* Vì khi CSS3 mới ra thì mỗi trình duyêt mỗi trợ 1 khác ( hỗ trợ properties ) để đàm bảo các thuộc tính có thể sử dụng được trên mọi trình duyệt mỗi khi nhà phát triển thêm 1 thuộc tính mới.
* Nhưng bây giờ CSS3 hoàn thiện nên trình duyệt nào cũng dc. Ko cần dùng tiền tố

CSS SELECTORS and COMBINATORS

**Selectors** -Selector là các lựa chọn dùng để truy vấn đến các thẻ HTML nhằm tác động vào chúng.

* VD : p{

property: value;

}

**CSS combinators** là mối liên hệ giữa các selector.

* VD :<div>

<p>Content</p>

</div>

* Thì thẻ div và thẻ p sẽ có mối liên hệ như sau: Thẻ div là thẻ cha và thẻ p là thẻ con, và trong CSS để truy vấn đến thẻ p thì ta có thể thông qua thẻ div như sau:
* div p{

/\* Các thuộc tính CSS\*/

}

**How many combinatior ?**

* descendant selector (space) : chọn tất cả những thẻ con trong 1 selector
* child selector (>) : chọn tất cả những thẻ con trực tiếp của một selector, Nghĩa là thẻ con nằm vô trong đúng một cấp so với selector đó.( nghĩa là nó độc lập mà ko lồng ghép 2 tag với nhau )
* adjacent sibling selector (+) : chọn tất cả các thẻ anh chị em nằm liền kề ngay sau 1 selector
* general sibling selector (~) : tương tự trên nhưng nằm ngoài selector

**What is a universal selector** ? Ký hiệu chọn toàn bộ

* (\*) sẽ chọn mọi đối tượng thuộc bất kỳ loại nào.
* VD \* {
* color: blue; /\* color of all the elements should be blue \*/

F: silver; /\* silver background is set for all the elements \*/

}

Bắt đầu từ CSS3, dấu sao \* có thể được dùng kết hợp cùng với [namespaces](https://developer.mozilla.org/vi/docs/Web/CSS/CSS_Namespaces):

* ns|\* - khớp với mọi đối tượng trong namespace *ns*
* \*|\* - khớp với mọi đối tượng
* |\* - khớp với mọi đối tượng không được khai báo namespace

**How overriding css work?(ghi đè)**

**Dựa vào độ ưu tiên mà sẽ tuỳ chỉnh overide**

**Độ ưu tiên CSS**

* Để rõ ràng hơn về độ ưu tiên này trong CSS thì mình làm câu chuyện nho nhỏ như sau cho các bạn dễ hình dung. Giả sử các bạn đang xếp hàng có 4 người mua bánh. A (anh bạn) B (Bạn) C (Chú bạn) D (Dì bạn) thì bây giờ theo thứ tự ưu tiên sẽ là A B C D
* Đầu tiên là các **elements**(thẻ) trong CSS như là thẻ p, div, section , header … thì độ ưu tiên trong CSS của nó nằm cuối ta có (0A, 0B, 0C, 1**D**)
* Tiếp theo là các **class,** **pseudo class** như .home , .content , :hover , :before , :after hoặc các **attribute**(thuộc tính) như a[target="\_blank"], input[type="text"], a[href^="http"]… thì độ ưu tiên của nó nằm kế cuối ta có (0A, 0B, 1**C**, 0D)
* Tiếp đến là các **id** như #header, #banner nó có độ ưu tiên thứ nhì ta được (0A, 1**B**, 0C, 0D)
* Và cuối cùng là **inline-style**. Nghĩa là code trực tiếp bên trong thẻ HTML luôn như này và nó có độ ưu tiên cao nhất ta được (1**A**, 0B, 0C, 0D)

CSS advance

* **Pseudo Class** trong CSS được sử dụng để viết CSS cho một trạng thái nào đó của một phần tử.VD viết CSS đổi màu các liên kết khi rê chuột vào, đổi thuộc tính một phần tử khi nhấp vào,…
* Các pseudo class được khai báo tại vùng chọn, đặt đằng sau vùng chọn và có dấu hai chấm (:) ngăn cách. Ví dụ: #link:hover (vùng chọn của #link khi rê chuột vào).
* **Một số Pseudo class thông dụng**
* :hover – Chọn trạng thái khi rê chuột vào một phần tử.
* visited – Được sử dụng cho liên kết, chọn liên kết khi đã được truy cập (dựa vào History trên trình duyệt).
* :link – Được sử dụng cho liên kết, chọn liên kết khi chưa nhấp vào.
* :active – Chọn phần tử khi họ chọn/nhấp vào.

**Pseudo element( phần tử giả )** được sử dụng để tạo một số style đặc biệt của element được chọn.

* VD : - Style cho chữ cái đầu tiên, hoặc dòng đầu tiên của phần tử.
  + Chèn nội dung trước hoặc sau nội dung của phần tử.
* **:: before pseudo-element** :  là pseudo-element được sử dụng nhiều và thông dụng nhất. pseudo-elementthường được sử dụng để thêm text, hình ảnh hay bất kỳ nội dung gì phía trước nội dung của phần tử được chọn.
* **:: after pseudo-element** :  được sử dụng để thêm nội dung nhưng là vào phía sau phần tử được chọn.
* **:: selection pseudo-element :** dùng để style cho một vùng văn bản được người dùng chọn (hay còn gọi là "**bôi đen**"). Chỉ có một số thuộc tính css khả dụng với ::selection là color, background, curso, và outline.
* **:: first-line pseudo-element** : pseudo-element ::first-line dùng để style cho dòng đầu tiên của phần tử được chọn( chỉ có thể được áp dụng cho các phần tử html là block.)
* font properties
* color properties
* background properties
* word-spacing
* letter-spacing
* text-decoration
* vertical-align
* text-transform
* line-height
* clear
* **::first-letter pseudo-element :** được sử dụng để thêmstyle đặc biệt vào chữ cái đầu tiên của văn bản.
* font properties
* color properties
* background properties
* margin properties
* padding properties
* border properties
* text-decoration
* vertical-align (only if "float" is "none")
* text-transform
* line-height
* float
* clear

RESPONSIVE

**@Media :** cho phép ta tùy chỉnh CSS cho nhiều thiết bị khác nhau từ máy tính cho đến điện thoại smart phone và các thiết bị in ấn. Thuộc tính này có cú pháp như sau:

@media not|only mediatype(all,print,screen) and (media feature) {

    CSS-Code;

}

**VD :**  Nếu chiều rộng của trình duyệt nhỏ hơn hoặc bằng 480px thì ẩn class .large

@media only screen and (max-width: 480px){

  .large{

      display: none;

  }

}

**Responsive :** Web pages can be viewed using many different devices: desktops, tablets, and phones. ... It is called**responsive** web design when you use **CSS** and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen

**Note:** A web page should look good on **any device**!

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

**Units +of responsive :**

* px. Pixels (px) are the most commonly used unit of measure because there are many standards in web design based around the pixel. ...
* % Percentages get a lot of use in responsive design and are pretty straightforward. ...
* em. This is relative to the font-size value of the element. ...
* rem.

**Why use em(** font-size value **) instead of px in design?**

* **Px** dễ sử dụng nhất, khi font size hiển thị sang 1 thiết bị khác thì px ko thay đổi theo
* **Em** : chỉnh theo element cha
* VD :

|  |  |
| --- | --- |
| 1  2  3  4  5 | <div class="parent">      <p class="child">...</p>  </div |
| 3  4  5  6 | -------------------------------  <style>     .parent { font-size: 16px; }   // 16px     .child  { font-size: 2em }    // 2\*16px = 32px  </style> |

* **Rem** : chỉnh theo root
* Nếu :root chưa được khai báo giá trị font-size thì sẽ lấy giá trị mặc định của brouser.
* -> Chúng ta nên chọn chàng rem hoặc em cho nàng font-size, khoảng cách (margin, padding…) thì rem, px, với các layouts như menu, sub-menu thì em là một gợi ý không tồi.

Layout( bố trí)

Thiết kế bố cục các thành phần chính của 1 trang web

CSS layout

* A website is often divided into **headers, menus, content and a footer:**
* **A header** is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:
* **A navigation bar** contains a list of links to help visitors navigating through your website:
* The layout in this section, often depends on the target users. The most common layout is one (or combining them) of the following:
* **1-column** (often used for mobile browsers)
* **2-column** (often used for tablets and laptops)
* **3-column layout** (only used for desktops)
* **The main content** is the biggest and the most important part of your site.

It is common with **unequal** column widths, so that most of the space is reserved for the main content. The side content (if any) is often used as an alternative navigation or to specify information relevant to the main content. Change the widths as you like, only remember that it should add up to 100% in total:

* **The footer** is placed at the bottom of your page. It often contains information like copyright and contact info:

**HTML layout elements**

* <header> - Defines a header for a document or a section
* <nav> - Defines a container for navigation links
* <section> - Defines a section in a document
* <article> - Defines an independent self-contained article
* <aside> - Defines content aside from the content (like a sidebar)
* <footer> - Defines a footer for a document or a section
* <details> - Defines additional details
* <summary> - Defines a heading for the <details> element

There are five different ways to create multicolumn layouts. Each way has its pros and cons:

* HTML tables (not recommended)
* CSS float property
* CSS flexbox
* CSS framework
* CSS grid

Different between atributes and properties