|  |  |
| --- | --- |
| **CSS UNIT** | |
| |  |  | | --- | --- | | **Absolute Lengths //đơn vị tuyệt đối** | | | **cm** | centimeters | | **mm** | millimeters | | **in** | inch (1in = 96px = 2.54cm = 72point = 6 picas) | | **px** | pixel | | **pt** | point | | **pc** | picas (1 picas = 12pt) | | **Relative Lengths //đơn vị tương đối** | | | **%** | Relative to the parent element | | **em** | Relative to the font-size of the element ( 2em = 2 times the size of the current font) | | ex | Relative to the x-height of current font | | **ch** | Relative to the *width* of the “0” (số 0) of current font-size | | **rem** | Relative to font-size of the **root** element ( 2em = 2 times the size of the root font) | | **vw** | Relative to 1% width of the viewport | | **vh** | Relative to 1% height of the viewport | | **vmin** | so sánh width và height của viewport 🡺 cái nào **nhỏ** hơn thì relative với 1% cái đó | | **vmax** | so sánh width và height của viewport 🡺 cái nào **lớn** hơn thì relative với 1% cái đó | | |
| **CSS Background** | |
| **background-color** | **Màu nền** |
| **background-color:** hexadecimal | hexadecimal w transparency | RGB | RGBA | HSL | HSLA | Color Name | Currentcolor **;**  **hexadecimal :** #RRGGBB (từ 00 -> FF) | hexadecimal with **transparency**: #RRGGBBTT (từ 00->FF)  **RGB :** **rgb( red , green , blue)** (từ 0->255 hoặc 0%->100%)  **RGBA**: **rgba( red, green, blue , alpla)** (alpha từ 0.0 -> 1.0 : đại diện Opacity :mờ đục)  **HSL :** **hsl( hue, saturation , lightness )** (hue :0->360 , saturatio : 0%->100% , lightness : 0%->100%)  **HSLA** : **hsla( hue , saturation, lightness, alpha)** (alpha từ 0.0->1.0)  **Color Name :** red, blue, coral ….  **Curentcolor**: vd : #myDIV{ color: blue ; border: 10px solid **currentcolor** ;} //curentcolor = blue, | |
| **background-image** | **Ảnh nền** |
| **background-image: url(“** img\_url **“)**  **url(“**img\_url\_1**”) , url(“**img\_url\_2**”) ;** // ảnh 1 nằm bên trên ảnh 2  **linear-gradient(** direction , color\_stop1, color\_stop2,…)  **direction:** to top, top right, to bottom, to left, 90deg,….. | |
| **background-repeat** | **Lập lại ảnh nền** |
| **background-repeat: repeat-x | repeatt-y | no-repeat ;** | |
| **background-attachment** | **Gán ảnh nền** |
| **background-attachment: scroll | fixed;**  **scroll:** thay đổi theo thanh cuộn  **fixed :** không thay đổi khi cuộn thanh cuộn | |
| **background-position** | **Xác định vị trí ảnh nền** |
| **background-posion : x y | X | initial | inherit ;**  **x , y , X : top | bottom | left | right | % | px | rem | em …..**  **background-position: X ;** // = X center;  **background-position: initial;** //nguyên bản, không ảnh hưởng bởi thuộc tính thẻ parent  **background-position: inherit;** //thừa hưởng; ảnh hưởng bới thuộc tính thẻ parent | |
| **background:** | **Shorthand của ảnh nền** |
| **background: color image repeat attachment position;**  **background**: #ffff00 url(“img.png”) no-repeat fixed right top; //có thể thiếu value nhưng phải đúng thứ tự | |
| **background-clip** | **Chỉ định vùng sơn của nền** |
| **background-clip: border-box | padding-box | content-box | initial | inherit ;**  **border-box : Default,** sơn hết từ boder cho tới content  v  **padding-box :** sơn từ phần padding tới content  **content-box :** sơn mỗi phần content | |
| **background-origin** | **Chỉ định vùng ảnh của nền**  **//border-box | padding-box | content-box | initial | inherit** |
| **background-size** | **Xác định kích thước ảnh nền** |
| **background-size: auto | *length* | cover | contain | initial | inherit ;**  **auto : Default ,** background image được thể hiền bằng kích thước gốc của ảnh  ***length:*** width height | width *( // height sẽ là auto*)  **cover :** resize ảnh nền để bao phủ toàn bộ vùng chứa, kéo dài 1 chiều tới khi phủ hết 1 cạnh.  **contain :** resize để ảnh hiển thị toàn bộ trong vùng chứa, lớn nhất có thể | |
| **CSS BORLDER** | |
| **border-…. : top&bottom&left&right | top right bottom left | top&bottom left&right ;**  **border-top: ,border-right: , border-bottom: , border-left:** | |
| **border-style** | **Hình dạng của border** |
| **border-style: dotted | dasher | solid | double | groove | ridge | inset | outset | none | hidden ;**  viền chìm viền nổi chìm nổi không viền ẩn viền  nội dung nội dung | |
| **border-width** | **Width of the four borders** |
| **border-width: size (px, pt, em,…) | thick | medium ;** | |
| **border-color** | **Color of the fours borders** |
| **border-color: name | HEX | RGB | HSL with transparent;** | |
| **SHORT HAND: border: width style color;** | |
| **border-radius** | **Bo góc của border** |
| **border-top-left-radius ; border-top-right-radius ; border-bottom-left-radius ; border-bottom-right-radius**  **border-radius: 2px; /**/ bo 4 góc=2px  **border-radius: 2px 3px;** //bo góc top-left&bottom-right=2px ; top-right&bottom-left=3px  **border-radius: 2px 3px 4px** //bo góc top-left=2px ; top-right&bottom-left=3px ; bottom-right=4px  **border-radius: 2px 3px 4px 5px** // top-left 🡪 top-right 🡪 bottom-right 🡪 bottom-left | |
| **CSS MARGINS** | |
| **margin | ( margin- top|right|bottom|right ) : auto | length | % | inherit;**  **margin : top&right&bottom&left | top right&left bottom | top&bottom right&left** | |
| **CSS PADDING** | |
| **padding | ( padding- top|right|bottom|right ) : auto | length | % | inherit;**  **padding : top&right&bottom&left | top right&left bottom | top&bottom right&left** | |
| **CSS Height and Width** | |
| **height | width: auto | length | % | initial | inherit;** | |
| **max-height** | **Maximum heigth of an element** |
| Nếu content lớn hơn max-height , nó sẽ bị OVERFLOW. Thuộc tính overflow được kích hoạt.  Nểu content nhỏ hơn max-height, thuộc tính max-height không có giá trị. | |
| **min-height** | **Minimun heigth of an element** |
| Nếu content nhỏ hơn min-height, thì min-height sẽ được áp dụng cho height property  Nếu content lớn hơn min-height, thì min-height không có giá trị | |
| **max-width** | **Maximum width of an element** |
| Nếu content lớn hơn max-width , nó sẽ thay đổi height của content  Nếu content nhỏ hơn max-width, thì max-width không có hiệu lực  //Ngăn không cho thuộc tính width lớn hơn max-width | |
| **min-width** | **Minimum of an element** |
| Nếu content nhỏ hơn min-width, thì mim-width sẽ được áp dung cho width propety  Nếu content lớn hơn min-width, thì min-width không có hiệu lực  //min-width ngăn không cho thuộc tính width nhỏ hơn min-width | |
| **CSS OUTLINE** | |
| **Outline** là đường kẻ xung quanh element, bên ngoài borders. Outline khác với border vì outline không phải một phần của element nên total width và height của element không ảnh hưởng bởi width và heigth của outline | |
| **outline-style** | **HÌnh dạng outline** |
| **outline-style: dotted | dasher | solid | double | groove | ridge | inset | outset | none | hidden ;**  viền chìm viền nổi chìm nổi không viền ẩn viền  nội dung nội dung | |
| **outline-width** |  |
| **outline-width: thin | medium | thick | *specific size* ;**  **1px 3px 5px** | |
| **outline-color** |  |
| **outline-color: name | HEX | RGB | HSL | invert;**  **invert :** Đảo ngược màu sắc ( so với background-color) đảm bảo viền được thấy được | |
| **SHORT HAND: outline : *width style color* ;** | |
| **outline-offset** | **Khoảng cách giữa outline và border** |
| **outline-offset:** *specific size***;** | |
| **CSS TEXT** | |
| **color** | **Màu chữ** |
| **text-align** | **Căn lề chữ** |
| **text-align: center | left | right | justify ; // justify:** Mỗi dòng được kéo dãì để mọi dòng đề có  chiều dài như nhau, các lề trái và phải đều thẳng hàng | |
| **direction + unicode-bidi** | **Thay đổi phương hướng của text** |
| **direction: rtl;** // right to left  **unicode-bidi: bidi-override;** //đừng tìm hiểu, nhức đầu lắm | |
| **vertical-align** | **Căn độ cao thấp so với mặt ngang** |
| **vertical-align: baseline | text-top | text-bottom | sub | super;**  **baseline : Default,** aaaaa **text-top , sub :** aaa**a**aa **text-bottom , sub:** aaa**a** | |
| **text-decoration** | **Thêm đường gạch cho text** |
| **text-decoration: overline | line-through | underline | none ;**  aaaaaaa ~~aaaaaaaa~~ aaaaaa aaaaaa //dùng xoá underline ở link | |
| **text-transform** | **Thêm cách viết HOA hoặc thường** |
| **text-transform: uppercase | lowercase | capitalize ;**  VIẾT HOA viết thường Viết Hoa Chữ Đầu | |
| **text-indent** | **Text Indentation** // thụt vào đầu dòng |
| **letter-spacing** | **Khoảng cách giữa 2 chữ cái** |
| **line-height** | **Khoảng cách giữa các dòng** |
| **word-spacing** | **Khoảng cách giữa 2 từ** |
| **text-overflow** | **Cách báo hiệu nội dung bị tràn** |
| **white-space:** nowrap;  **overflow:** hidden;  **text-overflow: clip | ellipsis | initial | inherit;**  Hello wo Hello… //elipsis : dùng dấu … đại diện cho phần tràn ra ngoài | |
| **white-space** | **Kiểm soát khoảng trắng** |
| **white-space : normal | nowrap | pre | pre-line | pre-wrap | inherit ;**  **vd: <p>** Đây là đoạn text có hai dòng.  Để làm mẫu cho thuộc tính white-space**. </p>**  **nomal :** văn bản sẽ dựa vào thuộc tính của thẻ **<p**>  //không có xuống dòng , và khoảng trắng và wrap khi window resize .  **nowrap :** văn bản có thuộc tính của thẻ <p> và không bị wrap khi **window resize .**  **pre :** văn bản có thuộc tính của thẻ <pre> // giữ nguyên xuống dòng và không wrap  **pre-line:** văn bản có thuộc tính của thẻ <pre> nhưng không có white-space giữa các chữ,  có wrap từng dòng nếu window resize nhỏ lại (vd: a b 🡪 a b)  **pre-wrap:** văn bản có thuộc tính của thẻ <pre> và có wrap từng dòng nếu window resize nhỏ lại | |
| **text-shadow** | **Hiệu ứng đổ bóng cho text** |
| **text-shadow: x y blur color; //** vd: text-shadow: 2px 2px 3px red;  **x & y** : vị trí của shadow, **blur:** độ mờ của shadow; **color:** màu shadow; | |
| **CSS FONT** | |
| **font-family** | **Chọn font family** |
| **CÓ 5 HỌ FONT CHỮ :**  **Serif :** Các chữ có nét nhọn vd **T B F R S** //tạo cảm giác sang  **San-serif :** Các chữ không có nét nhọn **T B F R S** //tạo cảm giác thân thiện, dễ đọc  **Monospace** : Các chữ có chiều rộng cố định **T B F R S** // tạo cảm giác máy móc  **Cursive**  : Các chữ dạng chữ viêt tay **T B F R S**  **Fantasy :** Các chữ trang trí , cảm giác vui tươi **T B F R S**  \*Khi khai báo font-family, nên khai một vài font dự phòng và kết thúc bằng một HỌ trong 5 phông trên  để tăng tính tương thích với các trình duyệt khác nhau, nên sử dụng các font trong **web safe fonts**  vd: **font-family:** “Times New Roman”, Times, serif**;** | |
| **font-style** | **font-style:** italic; **//** *aaaaa* |
| **font-weight** | **font-weight:** bold; **// aaaaa** |
| **font-variant** | **font-variant:** small-caps; **//** Bao binH  -> BAO BINH , VIẾT HOA hết nhưng chữ nào viết hoa rồi nó to hơn |
| **font-size** | **font-size:** px | em | vw | %..; |
| **Google Fonts** | **Sử dụng font của google** |
| **<head>**  **<link rel=”stylesheet” herf=**[**http://fonts.googleapis.com/css?family=Sofia**](http://fonts.googleapis.com/css?family=Sofia)**>**  **</head>**  Khi đó ta có thể sử dụng font Sofia nhứ sau: font-family=”Sofia, sans-serif; //font GG phải đặt trong “ “  Nếu muốn thêm nhiều font từ GG: **herf=**[**http://fonts.googleapis.com/css?family=Sofia**](http://fonts.googleapis.com/css?family=Sofia)**|Trirong|…”** | |
| **Google Effects** | **Thêm hiệu ứng cho text từ Google** |
| Gọi hiệu ứng từ Google bằng cách thêm **“ *effect=****effectname***”** vào sau GG API  **<head>**  **<link rel=”**stylesheet**” href=”**[**http://fonts.googleapis.com/css?family=Sofia**](http://fonts.googleapis.com/css?family=Sofia)**&effect=neon|outline|..” >**  **</head>**  Để sử dụng được hiệu ứng hãy đặt tên class của element bắt đầu với font-effect- :  vd: **class=”font-effect-neon”**  **<**h1 class=”**font-effect-**neon”> Tiêu đề này có hiệu ứng NEON </h1> 🡪 | |
| **SHORT HAND: font: *style variant weight size/line-height family*;**  vd: **font: italic small-caps bold 12px/30px Georgia,serif;** | |
| **CSS ICON** | |
| Để sử dụng icon vào HTML page là phải thêm thư viện icon;  <head> //Bootstrap Icons  <link rel=”stylesheet” href=”<http://maxcdn.bootstrapscdn.com/bootstrap/3.3.7/css/bootstrap.min.css>”>  </head>  Để sử dụng icon đặt tên phải sử dụng 2 element là **<i>** hoặc **<span>** và đặt tên class theo các tên đã xác định của các icon, và có thể dùng css để customized icon đó:  **<I** cl**ass=”**glyphicon glyphicon-cloud” **style=”**font-size:48px; color:red**;”> cloud </i> 🡪 cloud** | |
| **CSS LINK** | |
| **a:link** | nomarl, unvisited link |
| **a:visited** | a link the user has visited |
| **a:hover** | a link when the user mouses over it |
| **a:active** | a link the moment it is clicked |
| **a:hover phải đến sau a:link và a:visited**  **a:active phải đến sau a:hover**  **Dùng các css text , font, background , padding, cursor design cho <a>** | |
| **CSS LIST** | |
| **list-style-type** | **ul {list-style-type:** circle | square | dish .. **; }**  **ol {list-style-type:** number | upper-rowan | upper-alpha ;..**}** |
| **list-style-image** | **ul {** list-style-iamge: **url(“img\_url”); }** |
| **list-style-position** | **ol | ul {list-type-positon: inside | outside; }**  Xác định bullet points nằm trong (default) hay nằm ngoài khu vực text |
| **SHORT HAND: list-style: type position image;** | |
| **CSS TABLE** | |
| **border** | **Viền cho table** |
| **table, th, td { border: 1px solid black;} 🡪** | |
| **border-collapse: collapse** | **Hơp nhất 2 viền thành 1** |
| **table { border-collapse: collapse;} 🡪**  // border-collapse: separate; là *default* | |
| **width | height:** | **Kích thước cho table** |
| **text-align:** | **Căn lề chữ cho table** |
| **table | th | tr { text-alight: left | center | right ; }** | |
| **vertical-align:** | **căn vị trí text trong khung table** |
| **th | td { vertical-align: top | middle | bottom ; } 🡪** | |
| **padding** | **căn vị trí text trong khung table** |
| **tr:hover** | **Highlight hàng được trỏ trong table** |
| **tr:hover { background-color:** yellow**;} //tr = row :** nguyên hàng sẽ vàng khè khi bỏ chuột vào | |
| **tr:nth-child(even)** | **tạo table dạng kẻ sọc** |
| **tr:nth-child( even | odd ){ background-color: lightblue;}**  **even: odd:** | |
| **border-spacing** | **khoảng cách giữa các khung** |
| **caption-side:** | **vị trí caption của table :**  **caption-side: bottom | top;** |
| **empty-cells:** | **xử lý những ô table trống** |
| **table{ empty-cell: show | hide ; }** //show là default | |
| **table-layout:** | **bố trí bảng** |
| **table { table-layout: auto | fixed ; }**  **auto:** Chiều rộng của cột được đặt bởi nội dung rộng nhất và **không bị phá vỡ trong ô**. Content sẽ quyết định layout.  **fixed:** chiều rộng bảng và cột được quyết bằng chiều rộng của bảng và cột của ô đầu tiên. Các ô trong các hàng khác không ảnh hưởng đến độ rộng **cho dù content có bị vỡ.** Nếu hàng đầu không có chiều rộng nào thì chiều rộng cột sẽ được chia đều trên bảng.  \*\* LỢI ÍCH CỦA : **table-layout: fixed**; bảng sẽ hiển thị nhanh hơn. | |
| **CSS LAYOUT - DISPLAY** | |
| Qui tắc về **Block-level elements** và **Inline Elements:**  **Block-level element:** Luôn bắt đầu trên một dòng mới và chiếm hết width hết mức có thể  vd: <div> <h1-<h6> <p> <form> <header> <footer> <section>  **Inline Element:** Không bắt đầu trên một dòng mới và chỉ chiếm width khi cần thiết  <span> <a> <img> <hiệu ứng chữ>…..  ***\*\* An inline Elemet cannot contain a block-level element .\*\**** | |
| **display** | **Cách hiển thị** |
| **display: none | inline | block | inline-block | content | flex | grid | inline-flex | inline-table | list-item | run-in** | table | taple-caption | table-colum-group | table-header-group | table-cell | table-colum | table-row | initial **|** inherit;  **none** : element bị ẩn hoàn toàn // không hiển thị và không để lại vùng trống  **inline:** element có thuộc tính inline // bất cứ height và width đều không có hiệu lực  **block:** element có thuộc tính block-element // bắt đầu ở dòng mới và chiếm width hết mức có thể  **inline-block :** element có thuộc tính inline // nhưng có thể set height và width  **flex:** element trở thành flex-container  **inline-flex:** trở thành một inline nằm trong flex-container  **grip**: element trở thành grid-container  **inline-grip:** trở thành một inline nằm trong grid-element | |
| **visibility** | **Khả năng nhìn thấy** |
| **visibility: visible | hidden;**  Default | Ẩn nội dung // nhưng vẫn chừa lại khoảng trống trên layout  **visibility: collapse; // dùng trong <table>**  **table** {visibility: collapse;} 🡪 Table hợp nhất rồi biến mất tiêu  **tr**{visibility: collapse;} 🡪 row hợp nhất rồi mất hết, còn trơ là cái khung table lép xẹp  **td**{visibility: collapse;} 🡪 data hợp nhất rồi biến mất, còn trơ lại cái khung table + cái row empty | |
| **display: flex;** | **Hiển thị kiểu dãn** |
| **Tại display flex container:**  **display: flex;**  **flex-direction:** row | colum | row-reverse | colum-reverse;  //hướng main axis  **flex-wrap:** wrap | no-wrap | wrap-reverse;  Short Hand: **flex-flow:** flex-direction flex-wrap**;**  **justify-content:** center | flex-start | flex-end | space-between | space-around | space-evenly ;  //Canh theo **main axis: (main start) (main end) (space outside = 0) (space outside = 1/2 space inside) (space outside = space inside)**  **align-content:** stretch | center | flex-start | flex-end | space-between | space-around | space-evenly;  // Canh theo **cross axis (cross start) (cross end)**  **align-item :** stretch | center | flex-start | flex-end | | baseline ;  // Canh theo **cross axis và canh theo**  ***Tại flex item:***  **order :** vị trí của flexvd: order:1; sẽ nằm ngay flex-start  **flex-grow:** phình ra so với các phần khácvd: flex-grow:2 sẽ lớn gấp đôi flex-grow:1  **flex-shink:** co lại so với các phần khácvd: flex-shink:2 sẽ nhỏ gấp đôi flex-shink:1  **flex-basic:** cài giá trị cho item vd: 100px sẽ luôn có width=100px;  Short Hand**: flex : grow shink basic ;** (*canh theo main size, main axis*)  **align-seft:** auto | flex-start | flex-end | center | baseline | stretch;  **justify-seft:** auto | flex-start | flex-end | center | baseline | stretch; | |
| **CSS POSITION PROPERTY** | |
| **top, bottom, right, left** | **Xác định vị trí** |
| **z-index** | **Đặt thứ tự xếp lớp //đi với position: absolute;** |
| **z-index: …| -1 | -2 | 0 | 1 | 2 | ….;** | |
| **position** | **xác định loại của vị trí** |
| **position: static | relative | absolute | fixed ;**  **static:** tĩnh, không ảnh hưởng bới top, bottom, left & right  **relative:** khi set các thuộc tính top, bottom, left & rightcho element sẽ không ảnh hưởng tới layout,  các element khác sẽ dựa trên position: static của element để xắp sếp // *kiểu là mày bay đâu thì bay tụi*  *tao vẫn nằm ở vị trí như lúc mày ở position:static.*  **absolute :** tuyệt đối**,** Dùng thiết lập vị trí của một element nhưng nó sẽ luôn nằm trong một parent đang  là **relative**  **fixed :** element sẽ hiển thị luôn đi theo browser khi cuộn trang | |
| **CSS OVERFLOW** | |
| **overflow** | **Xử lý phần tràn ra** |
| **overflow: visible | hidden | scroll | auto;**  **visible : Default,** hiển thị phần tràn ra  **hidden :** ẩn phần tràn ra ngoài  **scroll:** tạo thanh cuộn  **auto:** tạo thanh cuộn khi cần thiết | |
| **overflow-x và overflow-y** | **xử lý phần tràn ra theo chiều x và y** |
| **CSS FLOAT & CLEAR** | |
| **float** | **trôi** |
| **float: none | left | right;**  **// thuộc tính trôi làm element nổi lên so** | |
| **clear** | **Làm rõ** |
| Sau khi sử dụng thuộc tính float và muốn element tiếp theo nằm dưới phần float đó ta dùng clear  vd: nếu .div1{ float:right;} thì .div{clear:right;}  **clear: left | right | bottom;** | |
| Nếu **phần tử float** cao hơn **phần tử container** thì nó sẽ bị tràn ra ngoài  oatloat  Sử dụng **overflow: auto** vào phần tử container để khắc phục    oatloat | |
| **CSS BOX-SIZING** | |
| Ở mặc định width và height của một element sẽ được tính như sau:  **width + padding + border = actual width of an element;**  **height + padding + border = actual width of an element;**  Điều này gây ra khi bạn set 2 <div> có **width=50%** nhưng chúng sẽ không nằm trên cùng 1 hàng được  Để giải quyết vấn đề này ta sử dụng:  **box-sizing: border-box;**  // lúc này **width = actual width of an element;**  **height = actual height of an element;** | |
| **CSS COMBINATOR** | |
| Mối quan hệ giữa các Selector:  **.** **Descendant Selector (space)** //con cháu, tất cả các selector nằm trong selector được chỉ định  vd :**div p{ }**  **<div>**  <p> </p>  <section> <p></p></section> //có 2 thẻ <p> là con và cháu của thẻ <div>  **</div>**  **. Child Selector ( > )** //con vd: **div > p { }**  **<div>**  <p> </p>  <section> <p></p></section> //có 1 thẻ <p> là con của thẻ <div>  **</div>**  **. Adjacent Sibling Selector ( + )** //em ruột ngay sau, vd: **div + p{ }**  <body>  <p> </p>  **<div>**  <p> </p>  <section> <p> </p></section *// có 1 thẻ <p> là em ruột ngay sau của thẻ <div> cùng mẹ là <body>*  **</div>**  <p></p>  <p></p>  </body>  **. General Sibing Selector ( ~ )** //em ruột, vd: **div ~ p { }**  <body>  <p> </p>  **<div>**  <p> </p>  <section> <p> </p></section *// có 2 thẻ <p> là em ruột của thẻ <div> cùng mẹ là <body>*  **</div>**  <p></p> **SPACE : CON CHÁU**  <section> <section> **> : CON**  <p></p> **+ : EM RUỘT NGAY SAU**  </body> **~ : TẤT CẢ EM RUỘT** | |
| **CSS Pseudo-class //lớp giả** | |
| Pseudo-class dùng xác định trạng thái đặc biệt của một phần tử  Cú pháp của Pseudo-class: **selector: pseudo-class { property : value; }**     |  |  |  | | --- | --- | --- | | **Pseudo-class** | **Example** | **Example description** | | **:active** | a:active | Chọn đường link được kích hoạt | | **:checked** | input:checked | Chọn các thẻ <input> đã được check | | **:disable** | input:disable | Chọn các thẻ <input> bị disable | | **:empty** | p:empty | Chọn các thẻ <p> mà để trống nội dung | | **:enable** | input:enable | Chọn các thẻ <input> đã được cho phép | | **:first-child** | p:first-child | Chọn các thẻ <p> là con đầu của một thẻ parent khác | | **:first-of-type** | p:first-of-type | Chọn các thẻ <p> là con đầu của một thẻ parent khác | | **:focus** | input:focus | Chọn các thẻ <input> được focus | | **:hover** | a:hover | Chọn các thẻ <a> được trỏ vào | | **:in-range** | input:in-range | Chọn các thẻ <input> có giá trị nằm trong vùng xác định | | **:invalid** | input:invalid | Chọn các thẻ input có giá trị không hợp lệ | | **:lang( )** | p:lang(*it*) | Chọn các thẻ <p> với giá trị <p lang=”it”> </p>  // *it* là định nghĩa cho cụm từ bằng ngôn ngữ nào vd: vi , en… | | **:last-child** | p:last-child | Chọn các thẻ <p> là con ÚT của ba má nó | | **:last-of-type** | p:last-of-child | Chọn các thẻ <p> là con ÚT của ba má nó | | **:link** | a:link | Chọn các thẻ <a> chưa được visited | | **:not(*selector*)** | :not(p) | Chọn bất cứ thể nào không phải là <p> | | **:nth-child(*n*)** | p:nth-child(2) | Chọn đứa con thứ 2 mà là thẻ <p>  // :first-child = :nth-child(1) | | **:nth-last-of-type(n)** | p:nth-last-of-type(2) | Đứa con kế út mà là thẻ <p> | | **:only-of-type** | p:only-of-type | Chọn đứa con duy nhất là <p> trong nhà | | **:out-of-range** | input:out-of-range | Chọn thẻ <input> có giá trị nằm ngoài vùng xác định | | **:read-only** | input:read-only | Chọn thẻ <input> có giá trị readonly | | **:root** | root | Chọn phần tử gốc của document //như body{ } | | **:target** | #news:target | Chọn các thẻ có **id=”news”** là mục tiêu hướng tới của một thẻ khác có **href=”#news”** //sau khi click hiệu ứng mới hiện | | **:valid** | input:valid | Chọn thẻ <input> có giá trị valid //vd nhập đúng kiểu email | | **:visited** | a:visited | Chọn các thẻ <a> đã được visit | |  |  |  | | **CSS Pseudo-elements** | | | | Các element giả được sử dụng để tạo kiểu cho các phần cụ thể cửa phần tử đó  Cú pháp  **selector : : pseudo-element { property : value; }** | | | | **Pseudo-Elements** | **Example** | **Example description** | | **::after** | p::after | Thêm something vào cuối content trong thẻ <p>  vd: **p::after { content : “ ! “ ;}** 🡺 cuối mỗi thẻ <p> xuất ra có thêm dấu **!** | | **::before** | p::before | Thêm something vào đầu content trong thẻ <p>  vd: **p::after { content : “ -> “ ;}** 🡺 đầu mỗi thẻ <p> xuất ra có thêm dấu **->** | | **::first-letter** | p::first-letter | Chọn chữ cái đầu trong mỗi thẻ <p> | | **::first-line** | p::first-line | Chọn dòng đầu tiền trong thẻ <p> | | **::marker** | ::marker | Chọn các *điểm đánh dấu* trong thẻ list <li> | | **::selection** | p::selection | Chọn các chữ được đánh dấu trong thẻ <p> //được bôi đen | | |
| **CSS SELECTOR** | |
| |  |  |  | | --- | --- | --- | | **Selector** | **Example** | **Example description** | | **\*** | **\*** | Select all elements | | **[ attribute ]** | **[target]** | Select all elements have attribute target //vd <a target=””> | | **[ attribute = value]** | **[color=red]** | Select all elements have aattribute: color=”red” | | **[attribute~=value]** | **[name~=mail]** | Select all elements have name attribute containing the word “mail” //vd name=”send mail”, | | **[attribute |= value]** | **[lang|=en]** | Select all elements have attribute lang=”en” or lang=”en-…” // bắt đầu bằng en- là được | | **[attribute ^= value]** | **a[href^ =”http”]** | Select all <a> has href **start** width “http” | | **[attribute $= value]** | **a[href$ =”.com”]** | Select all <a> has href **end** width “.com” | | |
| **CSS ANIMATION** | |
| **animation: name** **duration**  **timing-function** delay iteration-count direction fill-mode play-state;  **(đi kèm position:relative;)**  vd: **position:** relative;  **animation:** mymove 5s ease;  **@keyframes** mymove**{**  from {top: 0px; background-color: yellow;} ***hoặc*** 0%{ top: 0px; background-color: yellow;}  to {top: 200px; background-color: blue;} 100%{ top: 200px; background-color: blue;}  **} }**   * + **name:** xác định tên của keyframe ảnh động mà ta phải khai báo ở mục **@keyframes**;   + **duration:** xác định thời gian ảnh động hoàn thành //vd: 0.1s 2s   + **timing-function:** xác định hiệu ứng về tốc độ cho ảnh động   //một số timing-function hay sử dụng:  linear : ảnh động có tốc độ bằng nhau từ đầu đến cuối  ease: Default; fast-slow-fast  ease-in: slow start  ease-out: slow end  esse-in-out: slow start and slow end   * delay: 1s // sau 1s thì ảnh động chạy (default: 0) * iteration-count: 2 //ảnh động chạy 2 lần (default: 1) * direction: reverse - ảnh động chạy ngược lại , alternate - ảnh động chạy đi chạy lại * fill-mode: forwards - ảnh động giữ lại giá trị của last keyframe ( hiệu ứng cuối cùng, có thể ảnh hưởng bởi direction chứ không phải dòng cuối cùng của keyframes)   backwards - ảnh động giữ giá trị first keyframe và giữ nó trong suốt delay time  both – có cả 2 thuộc tính trên   * play-state: pause – dừng lại,   // vd cài div:hover{ animation-play-state: pause;} để trỏ chuột vào dừng ảnh động lại | |
| **CSS TRANSFORMS 2D – biến đổi** | |
| **transform-origin:**50% 50% 0; Default , tâm của element nằm ở giữa height và width // có thể đổi vị trí  **transform:** **translate(*x,y*) | translateX(*n*) | translateY(*n*) -- Di chuyển vị trí của element**  vd: transform: translate(50px,100px) // dịch qua phải 50px và dịch chuyển xuống dưới 100px  **scale(*x,y*) | scaleX(*n*) | scaleY(*n*) -- Tăng hoặc giảm height và width của element n lần**  vd: transform: scale( 2 , 0.5 ) // chiều width tăng 2 lần và height giảm 2 lần với tâm scale là vị trí của transform-origin  **rotale(*angle*) -- Xoay một góc *angle***  v  vd: transform: rotate ( 20deg ) // xoay element một góc 200 tại tâm xoay là vị trí của transform-origin  **skew(*x-angle,y-angle*) | skewX(*angle*) | skewY(*angle*) – Nghiêng**  vd: skew(20deg , 20deg) skewX(20deg) skewY(20deg) | |
| **CSS TRANSITIONS – kiểm soát sự thay đổi** | |
| **transition: transition-property transition-duration transition-timing-function transition-delay;**  **v**d: transition: width 2s linear 1s , heigth 3s linear 1s , transform 3s; | |
| **GRID - Hiển thị kiểu lưới** | |
| **CỘT HÀNG KHOẢNG CÁCH**    **Tại grid container:**   * **display:** grid | inline-grid;- thể hiện trình bày dạng grid * **grid-template-colums:** auto auto …. | px auto ….;- xác định số cột và witdh các cột * **justify-content:** center | start | end | space-between | space-around | space-evenly ; * **align-content:** center | start | end | space-between | space-around | space-evenly ; * **grid-colum-gap:** px * **grid-row-gap:** px * **grid-gap:** colum-gap row-gap;   **Tại gird item: Dựa vào Grid line để chia**   * **grid-colum-start:** line-start; * **grid-colum-end:** line-end; * **grid-colum:** line-start / line-end | line-start / span …. ; * **gird-row-start: line-start;** * **grid-row-end: line-end;** * **grid-row: line-start / line-end | line-start / span …. ;** * **gird-area: colum-start row -start colum-end row-end;** | |
| **VIEWPORT** | |
| **<meta name=”viewport” content=”width=device-width, initial-scale=1.0”>** | |
| **@media** | |
| **Cú pháp:**  **@media not|only mediatype and (*mediafeature* and|or|not *mediafeature*) {**  **//css-code**  **}**  Các loại **mediatype:**  **- screen :** màn hình;  **- print :** bản in;  **- speech:** chế độ đọc ( sử dụng hỗ trợ người khuết tật)  **- all - Default :** ở mặc định là cài cho cả 3 cái ở trên  Các loại **mediafeatures:**  **- min-width:**  **- max-width:** | |
| **CENTER A DIV** | |
| **1. lineheight và text-align : center**  **2. display: flex**  **3.**  **positon: absolute**  **top: 50%;**  **left: 50%;**  **transform: translateY(-50%) translateX(-50%) ;** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |