**CSS : Cascading Style Sheets**

Mục lục

[1. Phân loại CSS 2](#_Toc83990938)

[2. ID và Class 2](#_Toc83990939)

[3. Độ ưu tiên theo các cách sử dụng 2](#_Toc83990940)

[4. Đặt biến CSS 2](#_Toc83990941)

[5. Đơn vị trong CSS 3](#_Toc83990942)

[5.1 Absolute units 3](#_Toc83990943)

[5.2 Relative units 3](#_Toc83990944)

[6. Màu sắc CSS 3](#_Toc83990945)

[Xác định màu theo hệ thập lục phân – HEX 3](#_Toc83990946)

[Xác định màu theo kết hợp màu RGB 3](#_Toc83990947)

[Xác định màu HSL và HSLA 4](#_Toc83990948)

[7. CSS Backgrounds 4](#_Toc83990949)

[8. CSS Response 4](#_Toc83990950)

[9. CSS Grid 4](#_Toc83990951)

[10. CSS SASS 4](#_Toc83990952)

[11. CSS Reference 4](#_Toc83990953)

1. Phân loại CSS

* Internal: viết trên head của tứng trang html
* External: viết ra file .css riêng, nhiều trang html cùng sử dụng chung 1 file css
* Inline: định dạng css cho từng thẻ trong trang html

1. ID và Class

* ID: sử dụng CSS riêng cho từng đối tượng (vd: #heading)
* Class: sử dụng CSS riêng cho nhiều đối tượng (vd: .heading)

1. Độ ưu tiên theo các cách sử dụng

* Internal và External: cái nào mới nhất thì CSS sẽ nhận cái đó
* Inline > #id > .class > tag
* Equal specificity: nếu đều định nghĩa cùng tên thì cái nào mới nhất thì CSS sẽ nhận cái đó.
* Universal selector and inherited:
* VD: #id.class > #id > .class

1. Đặt biến CSS

* Có thể được gọi ở nhiều nơi

:root{

--text-color: red;

}

* h1, h2{

color: var(--text-color);

}

1. Đơn vị trong CSS
   1. Absolute units

* Đơn vị tuyệt đối
* Px
* Pt
* Cm
* Mm
* Inch
* Pc
  1. Relative units
* Đơn bị tương đối
* % : phụ thuộc vào thẻ chứa nó
* Rem : từ giá trị định nghĩa ở thẻ cha gần nhất chứa có, nó sẽ có giá trị bằng 1rem = giá trị cha
* Em
* vw (viewport width) bằng giá trị chiều rộng của thẻ chứa nó, giá trị cố định
* vh (viewport height) bằng giá trị chiều cao của thẻ chứa nó, giá trị cố định.
* vmin
* vmax
* ex
* ch

1. Selector

<https://hocwebchuan.com/reference/cssSection/>

1. Màu sắc CSS

- Màu HTML được chỉ định bằng tên màu xác định trước hoặc với các giá trị RGB, HEX, HSL, RGBA hoặc HSLA

## Xác định màu theo hệ thập lục phân – HEX

* Viết dấu “#” kết hợp với bao gồm chữ Latin (a,b,c,d,e,f,A,B,C,D,E,F) và số
* Rút gọn 6 ký tự giống nhau thành 3 ký tự: #ffffff => #fff
* Rút gọn 3 cặp ký tự : #ff0066 => #f06

## Xác định màu theo kết hợp màu RGB

* Sự kết hợp của 3 màu: đỏ(Red), xanh lá cây(Green), xanh dương (Blue)
* Cách viết rgb(155,167,1) : các giá trị từ 0 đến 255
* Một số giá trị đặc biệt:
  + Màu đen: rgb(0,0,0)
  + Màu đỏ: rgb(255,0,0)
  + Màu xanh lá cây: rgb(0,255,0)
  + Màu xanh dương: rgb(0,0,255)
  + Màu vàng: rgb(255,255,0)
  + Màu Trắng: rgb(255,255,255)

Xác định màu theo tên tiếng anh

## Xác định màu HSL và HSLA

* Được chỉ định bằng cách sử dụng màu sắc(hue), độ bão hoà (saturation) và độ sáng (lightness): hsl(147, 50%, 47%)
* Hue: Chỉ số màu sắc giao động giá trị từ 0-360
* 0: đỏ
* 120: xanh lá cây
* 240: xanh dương
* Độ bão hoà thể hiện theo giá trị phần trăm từ 0-100% (đậm hay nhạt)
  + 0% : màu xám
  + 100%: màu đầy đủ
* Độ sáng cũng thể hiện theo phần trăm 0-100% (tối hay sáng)
  + 0%: đen
  + 50%:vừa không tối không sáng
  + 100%: trắng
* Ở HSLA có thêm giá trị A(alpha) chỉ độ mờ của màu từ 0 đến 1
  + 0 hoàn toàn trong suốt
  + 1 không trong suốt

<https://hocwebchuan.com/reference/cssSection/css_color.php>

<https://www.w3schools.com/html/html_colors_hsl.asp>

1. CSS Backgrounds
   1. Background Color

* Chỉ định màu cho từng phần tử background-color

### Opacity / Transparency (Độ trong suốt)

* Thể hiện độ trong suốt của một phần tử giá trị từ 0-1, 1 là rõ nhất

opacity: 0.4

### Độ trong suốt với RGBA

* Sử dụng thành phần A(alpha) để điều chỉnh độ trong suốt của màu

background: rgba(0, 128, 0, 0.1);

* 1. Background Image
* Thuộc tính background-image được sử dụng để chỉ định một hình ảnh làm hình nền của một phần tử

background-image: url("paper.gif");

* 1. Background Repeat
* Mặc định, thuộc tính background-image sẽ bị lặp lại hình ảnh theo cả chiều ngang và chiều dọc
* Vậy sử dụng thuộc tính background-repeat điều chỉnh việc lặp theo chiều ngang hay chiều dọc

background-repeat: repeat-x;

background-repeat: repeat-y;

* Hoặc sử dụng giá trị no-repeat để không lặp lại

background-repeat: no-repeat;

* Sử dụng thuộc tính background-position để định dạng vị trí của hình ảnh

background-position: right top;

* 1. Background Attachment
* Sử dụng thuộc tính background-attachment để cổ định hay cho hình ảnh scroll

background-attachment: fixed;

background-attachment: scroll;

* 1. Background Shorthand
* Là cách viết gom lại các css trên 1 dòng

background: #ffffff url("img\_tree.png") no-repeat right top;

<https://www.w3schools.com/css/css_background_repeat.asp>

1. CSS Border

* Sử dụng thuộc tính border-style để tạo style cho đường viền
  + dotted 
  + dashed 
  + solid 
  + double 
  + groove 
  + ridge 
  + inset 
  + outset 
  + none No border
  + hidden Hidden border
  + dotted dashed solid double (mix) /\* top right bottom left\*/ 
  1. CSS Border Width
* Sử dụng thuộc tính border-width để tăng giảm độ đậm nhạt

border-width: 5px;

border-width: medium;

border-width: thick;

border-width: 5px 20px /\*top bottom\*/

border-width: 5px 5px 5px 5px /\* top right bottom left\*/

* 1. CSS Border Sides
* Định dạng cụ thể một vị trí trong border để css

border-top-style: dotted;  
  border-right-style: solid;  
  border-bottom-style: dotted;  
  border-left-style: solid;

* Dựa vào số lượng của giá trị border mà xác định vị trí được css ở đâu

/\* Four values: top right bottom left \*/  
  border-style: dotted solid double dashed;  
  
/\* Three values: top (2 bên left right) bottom \*/  
  border-style: dotted solid double;  
  
/\* Two values: (trên dưới) (2 bên) \*/  
  border-style: dotted solid;  
  
/\* One value: toàn bộ \*/

  border-style: dotted;

* 1. CSS Rounded Borders
* Sử dụng thuộc tính border-radius để bo tròn góc cho đương viền

border-radius: 5px;

## **Border Collapse**

|  |  |  |  |
| --- | --- | --- | --- |
| border-collapse | collapse | border-collapse: collapse; | Khoảng trống giữa các đường viền (border) của table bị loại bỏ, chỉ còn đường viền duy nhất. |
| separate | border-collapse: separate; | Khoảng trống giữa các đường viền (border) của table vẫn giữ nguyên, đây là dạng mặc định của table. |
| inherit | border-collapse: inherit; | Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài). |

<https://www.w3schools.com/css/css_border.asp>

Border Spacing

1. Margins - Padding



Ban đầu

<div>

<p>HỌC WEB CHUẨN</p>

</div>



div {

border: 1px solid red;

}

div p {

background: #66CC00;

}

* 1. Margin
* Thuộc tính margin canh lề cho thành phần

Sau khi CSS margin-top :10px;

div p {

background: #66CC00;

margin: 10px 0 0 0;

}



* 1. Padding
* Thuộc tính padding thêm vào khoảng không cho thành phần

Sau khi CSS padding-top: 20px;

div p {

background: #66CC00;

padding: 20px 0 0 0;

}

Shape

Description automatically generated with medium confidence

Các side (phân bố vị trí) của margin và padding giống như border side

* 1. Giá trị inherit

1. Height và Witdth
2. Box Shadow

* Chỉnh màu ẩn đằng sau một box

#example1 {  
  box-shadow: 5px 10px;  
}  
  
#example2 {  
  box-shadow: 5px 10px #888888;  
}

Graphical user interface, text, application

Description automatically generated

1. Box-sizing

* Theo mặc định, chiều rộng và chiều cao của một phần tử được tính như sau:
  + Width + padding + border = chiều rộng thực của phần tử
  + Height + padding + border = chiều cao thực của phần tử
* Xác định lại chiều rộng và chiều cao cho thành phần :

p {

background: #cccccc;

border: 5px solid #cc0000;

height: 100px;

width: 200px;

}

p.contentBox {

box-sizing: content-box;

-moz-box-sizing: content-box;

-webkit-box-sizing: content-box;

}

p.borderBox {

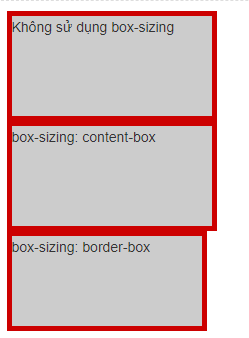
box-sizing: border-box;

-moz-box-sizing: border-box;

-webkit-box-sizing: border-box;

}

* Thấy giá trị border-box, chiều rộng và chiều cao hiện tại của thành phần đã bao gồm luôn phần border



1. Chữ hoa lên, thường xuống

text-transform: uppercase;

1. Transition
2. scroll-behavior

* Click thẻ a scroll tới section khác
* Có 4 giá trị:
  + Auto: nhảy đến lập tức
  + Smooth: kéo đến
  + Inherit
  + Initial

1. Thuộc tính transform

* Thuôc tính transform xác định một chuyển động 2 chiều, 3 chiều, có thể xoay, tỷ lệ, di chuyển, nghiêng,…

<https://hocwebchuan.com/reference/cssSection/css3/pr_transform.php>

1. fill thuộc tính màu của svg

* Đổ màu cho svg
* .hero .glide\_\_arrow svg{
* height: 1.8rem;
* width: 1.8rem;
* fill: var(--primaryColor);
* }

1. Flex

* Đặt độ dài linh hoạt cho mặt ngang
* Giá trị mặc định: 0 1 auto %
* Flex viêt tắt của:
  + Flex-grow
  + Flex-shrink
  + Flex-basis

1. Thuộc tính display

Code: file /display\_property

1. Thuộc tính visibility

* Xác định thành phần có được nhìn thấy hay không

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **giá trị** | **Ví dụ** | **Mô tả** |
| visibility | collapse | visibility: collapse; | Làm "sụp đổ" các hàng và cột của thành phần [table](https://hocwebchuan.com/reference/tag/tag_table.php), giá trị này chỉ sử dụng cho thành phần table. |
| hidden | visibility: hidden; | Thành phần sẽ không được nhìn thấy. |
| visible | visibility: visible; | Hiển thị thành phần, đây là dạng mặc định. |
| inherit | visibility: inherit; | Xác định thừa hưởng thuộc tính từ thành phần cha (thành phần bao ngoài). |

1. CSS Response

1. CSS Grid

display: grid;

grid-template-columns: auto auto auto auto;//số cột trên hàng

grid-column-gap: 10px;//khoảng cách giữa các cột

grid-row-gap: 25px//khoảng cách giữa các hang

* Gộp 2 khoảng cách thành: 10px 25px

1. CSS SASS
2. CSS Reference