[CSS: 3](#_Toc83650403)

[I. CSS Basic 3](#_Toc83650404)

[1. Overview 3](#_Toc83650405)

[1.1. What’s CSS 3](#_Toc83650406)

[1.2. Type of Css 5](#_Toc83650407)

[2. CSS Basic 6](#_Toc83650408)

[2.1. CSS Selector 6](#_Toc83650409)

[2.2. CSS Variables 9](#_Toc83650410)

[a. How to declare 9](#_Toc83650411)

[b. How to use 9](#_Toc83650412)

[c. How to access variables with JavaScript 10](#_Toc83650413)

[d. Easily change variable values when responsive 11](#_Toc83650414)

[e. Also we have custom fallback values of variables 11](#_Toc83650415)

[2.3. CSS Units 12](#_Toc83650416)

[a. Relative Units 12](#_Toc83650417)

[b. Absolute Units 13](#_Toc83650418)

[c. Viewport Units 13](#_Toc83650419)

[d. Browser Support 14](#_Toc83650420)

[2.4. CSS Text 14](#_Toc83650421)

[a. Text color 14](#_Toc83650422)

[b. Text-decoration 15](#_Toc83650423)

[c. Text-transform 16](#_Toc83650424)

[d. Text-align 16](#_Toc83650425)

[e. Text-indent 17](#_Toc83650426)

[2.5. CSS Links 18](#_Toc83650427)

[2.6. CSS Font 20](#_Toc83650428)

[a. Một số thuộc tính chính của font 20](#_Toc83650429)

[b. Trang web font miễn phí 22](#_Toc83650430)

[2.7. CSS List 23](#_Toc83650431)

[a. Một số thuộc tính chính của list 23](#_Toc83650432)

[2.8. CSS Table 25](#_Toc83650433)

[a. Các thuộc tính CSS định dạng bảng 26](#_Toc83650434)

[border 26](#_Toc83650435)

[text-align 30](#_Toc83650436)

[2.9. CSS Padding 33](#_Toc83650437)

[2.10. CSS Border 34](#_Toc83650438)

[2.11. CSS Margin 37](#_Toc83650439)

[2.12. CSS Background 38](#_Toc83650440)

[a. CSS thiết lập màu nền cho background 39](#_Toc83650441)

[b. CSS thiết lập hình nền cho backround 40](#_Toc83650442)

[c. Cho phép lặp hoặc không lặp background 40](#_Toc83650443)

[d. Thiết lập vị trí hiển thị cho background 42](#_Toc83650444)

[e. Thiết lập background đứng im khi scroll (fixed background) 43](#_Toc83650445)

[f. Sử dụng thuộc tính background nâng cao 44](#_Toc83650446)

[g. Sử dụng background-clip 44](#_Toc83650447)

[h. Cấu hình kích thước cho background (background-size) 44](#_Toc83650448)

[i. Thuộc tính background-origin trong CSS3 45](#_Toc83650449)

[k. CSS3 Multiple backgrouds 47](#_Toc83650450)

[j. CSS Background shorthand 49](#_Toc83650451)

[2.13. CSS Function 49](#_Toc83650452)

[a. Var Function 49](#_Toc83650453)

[b. Radial-gradient() Function 50](#_Toc83650454)

[c. Rgba() Function 51](#_Toc83650455)

[d. Rgb() Function 52](#_Toc83650456)

[e. Calc() Function 52](#_Toc83650457)

[F. Attr() Function 53](#_Toc83650458)

# 

# CSS:

# I. CSS Basic

## 1. Overview

## 1.1. What’s CSS

CSS là chữ viết tắt của cụm từ tiếng Anh "Cascading Style Sheet", dịch ra tiếng Việt thì thực sự mình cũng không thể dịch cho chuẩn được, nhưng về ý nghĩa thì CSS dùng để trình bày bổ cục của một Website dựa vào các thẻ HTML. Nói nôm na thì CSS sẽ tác động thay đổi cách hiển thị mặc định của các thẻ HTML nhằm thân thiện với người dùng hơn hay nhằm một mục đích cụ thể nào đó.

Có một ví dụ trực quan để diễn tả nhiệm vụ của CSS như sau, giả sử bạn xây nhà và bạn sẽ sử dụng gạch để xây dựng ngôi nhà. Nhưng để trang trí cho ngôi nhà thì bạn sẽ sử dụng một số đồ nội thất khác như sơn màu, ... thì lúc này sơn màu sẽ tác động làm thay đổi color mặc định của gạch. Nói ví von thi màu sơn chính là CSS và gạch chính là HTML.

CSS đóng một vai trò rất quan trọng khi bạn xây dựng giao diện Website. Thông thường bạn sẽ  sử dụng một chương trình thiết kế như Photoshop để tạo giao diện, sau đó các Frontend Developer sẽ dựa vào file đó kết hợp với HTML & CSS để convert thanh một file có đuôi mở rộng là .html, file File này sẽ chứa code **HTML và CSS**.

Có 3 cách để tích hợp CSS vào với HTML.

Cách 1: Internal

Internal CSS code được đặt trong mục <head> của một trang nhất định. Classes và IDs có thể được chỉ định tới CSS, nhưng nó chỉ hoạt động trên trang có định đó.  Classes và IDs có thể được dùng để dẫn tới code CSS, nhưng chỉ hoạt động trên trang đó. Style CSS được nhúng bằng cách này sẽ cần phải được tải về mỗi khi trang được load, vì vậy có thể dẫn đến mất nhiều thời gian để duyệt website. Tuy nhiên, trong một số trường hợp internal stylessheet rất hữu dụng. Một ví dụ là bạn muốn gửi một trang mẫu – vì tất cả đều chỉ nằm trong một trang – xem preview sẽ dễ hơn nhiều. Internal CSS được đặt trong thẻ <style></style>. Một ví dụ của internal stylesheet:

|  |
| --- |
| <head>  <style type="text/css">  p {color:white; font-size: 10px;}  .center {display: block; margin: 0 auto;}  #button-go, #button-back {border: solid 1px black;}  </style>  </head> |

Cách 2: External

Có lẽ là cách dễ nhất để thêm Style CSS vào website của bạn là liên kết với một file .css ở bên ngoài. Bằng cách này bất kỳ thay đổi nào bạn thực hiện trên CSS sẽ thay đổi trên toàn website. Một cách để làm là đặt link CSS trong mục <head> của trang:

|  |
| --- |
| <head>   <link rel="stylesheet" type="text/css" href="style.css" />  </head> |

Trong đó  **style.css** chứa tất cả các style rules. Ví dụ:

|  |
| --- |
| .xleftcol {    float: left;    width: 33%;    background:#809900;  }  .xmiddlecol {    float: left;    width: 34%;    background:#eff2df;  } |

Cách 3: Inline

Inline CSS được sử dụng cho một thẻ HTML xác định. <style> attribute được dùng để style một HTML tag xác định. Sử dụng CSS bằng cách này không được khuyến khích, vì mỗi tag HTML cần được styles độc lập. Quản lý website sẽ rất khó nếu bạn chỉ sử dụng inline CSS. Ví dụ, trong trường hợp bạn không có quyền truy cập tới file CSS hoặc cần áp dụng style cho một element duy nhất. Ví dụ của một trang HTML với inline CSS sẽ giống như sau:

|  |
| --- |
| <!DOCTYPE html>  <html>  <body style="background-color:black;">  <h1 style="color:white;padding:30px;">Hostinger Tutorials</h1>  <p style="color:white;">Something usefull here.</p>  </body>  </html> |

## 1.2. Type of Css

CSS có nhiều kiểu nhưng gom lại thì chúng ta sẽ có các thể loại CSS thông dụng như sau:

* **Background**: CSS tùy chỉnh hình nền
* **Text**: CSS tùy chỉnh cách hiển thị đoạn text
* **Font**: CSS tùy chỉnh kích thước, kiểu chữ
* **Link**: CSS tùy chỉnh link
* **List**: CSS tùy chỉnh danh sách
* **Table**: CSS tùy chỉnh bảng
* **Box model**: Mô hình box model kết hợp padding, margin, border.

## 2. CSS Basic

## 2.1. CSS Selector

Có rất nhiều loại selector nhưng trong bài này chúng ta tìm hiểu một vài cách về **DOM selector** nhé. Trong bài này chúng ta chỉ tìm hiểu ba cách chính đó là:

* CSS Selector phân cấp
* CSS Selector ID
* CSS Selector Class

**Selector phân cấp**

Phân cấp nghĩa là sẽ dựa vào cấp cha để tìm cấp con.

Example:

Html:

|  |
| --- |
| <div>      <p>          Nội dung sẽ có màu đỏ vì nó nằm trong thẻ p.      </p>  </div>  <p>      Nội dung không có màu vì nó nằm ngoài thẻ p.  </p> |

Css:

|  |
| --- |
| div p{  color: red  } |

**Selector ID**

Bạn lưu ý là trong một trang web ID là duy nhất nhé, nghĩa là nếu bạn định nghĩa hai ID giống nhau trong 1 layout thì không đúng chuẩn giao diện của W3C.

Giả sử bạn có nhiều thẻ div và bạn muốn viết CSS cho một thẻ DIV nào đó thôi thì bạn có thể chọn giải pháp là Selector theo ID của HTML. Chúng ta sử dụng dấu thăng (#) để đại diện cho ID.

Html:

|  |
| --- |
| <div id="active">  ACTIVE  </div>  <div>  NON-ACTIVE  </div> |

Css:

|  |
| --- |
| #active{  background: red  } |

**Selector class**

Với ID là duy nhất thì class ngược lại, nghĩa là bạn có thể cho nhiều thẻ HTML có cùng tên class, điều này khá tiện lợi cho CSS. Ví dụ bạn cần style cho một số thẻ div nào đó thôi thì nếu bạn dùng ID thì không hay lắm vì phải viết nhiều lần, chính vì vậy ta sẽ chọn class. Selector cho class sẽ là dấu chấm (.).

Html:

|  |
| --- |
| <div class="bg-yellow">  Yellow  </div>  <div>  NONE  </div>  <div class="bg-yellow">  Yellow  </div>  <div class="bg-yellow">  Yellow  </div> |

Css:

|  |
| --- |
| **.**bg-yellow{  background: yellow  } |

## 2.2. CSS Variables

### a. How to declare

Để khai báo một biến, trước tiên bạn cần quyết định phạm vi của biến đó sẽ tồn tại. Đối với global scope bạn có thể sử dụng :root, hoặc bạn cũng có thể tạo các biến cục bộ, tên biến phải bắt đầu bằng 2 dấu gạch ngang (--) và được phân biệt chữ hoa và chữ thường.

|  |
| --- |
| :root {  --main-color: #ffeead;  --main-background: #ff0000;  } |

### b. How to use

Để truy cập một biến, bạn cần sử dụng hàm var () và truyền tên của biến làm tham số.

|  |
| --- |
| .title {  color: var(--main-color);  background-color: var(--main-background);  } |

Kết quả của đoạn mã ở trên giống với SASS hoặc LESS khi được compiled. Tuy nhiên, so với các chương trình tiền xử lý, CSS variables có những lợi ích nhất định:

* Được hỗ trợ trực tiếp bởi trình duyệt, không phải biên dịch.
* Các biến được chia theo tầng (cascading). Cũng như CSS selectors, thuộc tính tùy biển có thể được quy định lại bởi những luật ở tầng thấp hơn.
* Giúp mã nguồn dễ đọc và có ý nghĩa hơn, nâng cao tính tùy biến và khả năng bảo trì.
* Hỗ trợ hầu hết các trình duyệt hiện tại
* Bạn cũng có thể kết hợp với hàm calc() khi sử dụng CSS variables:

|  |
| --- |
| :root {  --default-font-size: 1.1rem;  }  h1 {  font-size: calc(var(--default-font-size) \* 5); /\* 5.5rem \*/  } |

### c. How to access variables with JavaScript

Do CSS variables tồn tại trong DOM, có thể được truy xuất và thay đổi bằng JavaScript. Tính năng này mở ra những cơ hội mới rất hữu ích khi lập trình frontend.

|  |
| --- |
| var root = document.querySelector(':root');  var rootStyles = getComputedStyle(root);  var mainColor = rootStyles.getPropertyValue('--main-color');  console.log(mainColor);  --> '#ffeead' |

Thay đổi giá trị của biến bằng Javascript

|  |
| --- |
| root.style.setProperty('--main-color', 'red') |

### d. Easily change variable values when responsive

Chúng ta có thể thay đổi giá trị của biến khi sử dụng @media, @document, hay @support…

|  |
| --- |
| :root {  --main-font-size: 16px;  }  media all and (max-width: 600px) {  :root {  --main-font-size: 12px;  }  } |

### e. Also we have custom fallback values of variables

Sử dụng hàm var (), bạn có thể xác định nhiều giá trị fallback khi biến đã cho chưa được xác định, điều này có thể hữu ích khi làm việc với Custom Elements và Shadow DOM.

|  |
| --- |
| .el-one {  color: var(--my-var, red); /\* Red if --my-var is not defined \*/  }  .el-two {  background-color: var(--my-var, var(--my-background, pink)); /\* pink if my-var and --my-background are not defined \*/  } |

## 2.3. CSS Units

CSS có nhiều đơn vị đo, phổ biến nhất là pixels, nhưng bên cạnh đó còn khá nhiều đơn vị đo khác tuy không được biết đến nhiều nhưng đôi khi chúng rất hữu dụng trong vài trường hợp cụ thể.

Trong bài viết này đề cập đến **relative units**, **absolute units** và **viewport units**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Media** | **Recommended** | **Occasional use** | **Infrequent use** | **Not recommended** |
| Screen | em, rem, % | px | ch, ex, vw, vh, vmin, vmax | cm, mm, in, pt, pc |
| Print | em, rem, % | cm, mm, in, pt, pc | ch, ex | px, vw, vh, vmin, vmax |

### a. Relative Units

**Relative units** là loại đơn vị sẽ có giá trị tương đối so với độ dài của thuộc tính.

Ngược lại với **Absolute units** như pixels, points hay centimeters, chúng ta có thể xác định kích thước theo **relative units** như %, em hoặc rem.

Đối với hầu hết các trình duyệt, font-size mặc định sẽ là 16px, ta có thể lấy giá trị này làm giá trị chuẩn để tính toán (vd: 16px tương đương với 1em, 1rem hoặc 100%)

|  |  |
| --- | --- |
| **Unit** | **Description** |
| % | Có giá trị tương đối so với phần tử cha |
| em | Tương đối so với font-size của phần tử cha (vd 2.5em tức là font sẽ lớn hơn normal font 2.5 lần) |
| rem | Tương đối so với phần tử gốc. Phần tử gốc ở đây là thẻ html |
| ch | Tương đối so với width của phần tử "0" |
| ex | Tương đối so với x-height của font hiện tại |

### b. Absolute Units

**Absolute units** là loại đơn vị có giá trị không thay đổi theo kích thước màn hình, hướng và các biến thể khác, tuy nhiên chỉ chính xác tuyệt đối khi output có độ phân giải đủ cao.

**Absolute units** sẽ cần thiết trong các trường hợp yêu cầu kích thước phần tử phải chính xác 100% và không được thay đổi. Chúng cũng khá hữu dụng nếu ta muốn cố định một khoảng nào đó để tránh các trường hợp khoảng đó trở nên quá rộng hoặc quá hẹp.

|  |  |  |
| --- | --- | --- |
| **Unit** | **Description** |  |
| cm | centimeters | 1 cm = 1 cm |
| mm | millimeters | 10 mm = 1 cm |
| in | inches | 1 in = 96px = 2.54 cm |
| px | pixels | 1 px = 1/96th of 1 in |
| pt | points | 1 pt = 1/72 of 1 in |
| pc | pica | 1pc = 12 pt |

### c. Viewport Units

Các **Viewport Units** đại diện cho một tỉ lệ % của khung nhìn (viewport) của trình duyệt hiện tại.

Sự khác biệt so với **Percentage units** là các **Viewport units** luôn luôn được tính bằng tỉ lệ phần trăm kích thước viewport của trình duyệt, trong khi đó **Percentage units** lại kế thừa kích thước của các phần tử cha của chúng.

|  |  |
| --- | --- |
| **Unit** | **Description** |
| vw | 1% chiều rộng của viewport |
| vh | 1% chiều cao của viewport |
| vmin | 1vw hoặc 1vh, tùy theo số nào nhỏ hơn |
| vmax | 1vw hoặc 1vh, tùy theo số nào lớn hơn |

### d. Browser Support

Các đơn vị đo được hỗ trợ trong từng phiên bản của trình duyệt, trích từ w3school:

## 2.4. CSS Text

CSS cung cấp cho chúng ta một số thuộc tính định dạng text như màu sắc, hiển thị chữ in hoa hoặc in thường, vị trí hiển thị, ... Những thuộc tính này khá hay và rất quan trọng vì nó quyết định tính thẩm mỹ cho các dòng chữ trong website.

Trong bài chúng ta tìm hiểu một số thuộc tính định dạng cho text như sau:

* color
* text-decoration
* text-transform
* text-align
* text-indent

### a. Text color

Ta sử dụng thuộc tính color để gán màu sắc cho chữ.

Trong CSS thì ta sử dụng 3 cách biểu diễn color như sau:

* HEX: có định dạng #ma\_mau
* RGB: có định dạng rgb(xxx,yyy,zzz)
* Tên màu: ví dụ red, blue

Example:

|  |
| --- |
| .color-red{  color: red;  }  .color-blue{  color: blue;  }  .color-pink{  color:#d624d0  } |

### b. Text-decoration

Để gạch chân cho chữ thì bạn sử dụng thuộc tính text-decoration với giá trị:

* underline: gạch chân
* none: bỏ gạch chân

Example:

|  |
| --- |
| .gach-chan{  text-decoration: underline;  text-decoration-color: #28a4c9;  } |

### c. Text-transform

Nếu bạn muốn một thẻ nào đó luôn luôn hiển thị in hoa hoặc luôn luôn hiển thị in thường mặc dù ta nhập vào là hoa hay thường thì sử dụng thuộc tính text-transform với các thuộc tính:

* uppercase: chuyển đổi in hoa
* lowercase: chuyển đổi in thường
* none: không chuyển đổi gì cả (mặc định)

|  |
| --- |
| .upper{  text-transform: uppercase;  }  .lower{  text-transform: lowercase;  } |

### d. Text-align

Nếu bạn muốn đoạn text hiển thị ở gữa màn hình hoặc bên trái, bên phải thì sử dụng thuộc tính text-align với các giá trị:

* center : hiển thị ngay giữa
* left: hiển thị bên trái
* right: hiển thị bên phải
* justify : hiển thị canh đều so với lề phải và lề trái

Example:

|  |
| --- |
| .show-center{      text-align: center;  }  .show-left{      text-align: left;  }  .show-right{      text-align: right;  }  .show-justify{      text-align: justify;  } |

### e. Text-indent

Nếu bạn muốn chữ hiển thị ở một vị trí nào đó tính từ góc trái trên cùng (tính theo pixels) thì bạn dùng thuộc tính text-indent. Thuộc tính này thường dùng để gắn trong nội dung thẻ a của logo và sau đó dùng nó để ẩn text đi.

Ta sử dụng text-indent và giá trị của nó là số pixel lớn hơn kích thước màn hình (tính theo số âm).

Example:

|  |
| --- |
| .hide-text{  text-indent: -9999px  } |

## 2.5. CSS Links

[Thẻ a](https://freetuts.net/dung-the-a-trong-html-de-tao-links-335.html) đóng vai trò rất quan trọng vì nhiệm vụ của nó là giúp chuyển trang giữa các file trong hệ thống website. Không những vậy mà bên [SEO](https://freetuts.net/thu-thuat/thu-thuat-seo-onpage-va-toi-uu-onpage-danh-cho-coder-1732t.html) nó cũng có tác dụng chuyển hướng trang để các công cụ tìm kiếm có thể lọc toàn bộ website của bạn.

Mỗi website sẽ có nhiều thẻ a và bạn muốn bắt mắt với người dùng thì buộc phải style cho nó như xác định chiều cao, độ lớn chữ, màu sắc, hover, ... Và trong bài này mình sẽ trình bày tất cả các vấn đề này.

**Nội dung như sau**:

* Color
* Text-decoration
* Background
* Style(hover, visited, link, active)

a. Color

|  |
| --- |
| a{  color: blue  } |

b. Text decoration

Thông thường khi bạn tạo thẻ a thì theo mặc định nó có gạch chân nên để tắt gạch chân thì bạn sử dụng thuộc tính text-decoration:none.

|  |
| --- |
| a.non-textdecoration{  color: red;  text-decoration: none;  } |

c. Background

Cũng tương tự các phần trên ta sẽ sử dụng một thuộc tính chuyên về [CSS background](https://freetuts.net/thiet-lap-mau-nen-voi-css-background-347.html) đó là background.

|  |
| --- |
| a{  background: blue;  color:#FFF; /\*white\*/  } |

d. Style

Các sự kiện này xay ra khi chúng ta dùng chuột thao tác lên nó.

* hover: Khi bạn hover chuột qua nó sẽ có tác dụng
* visited: khi ạn click vào the a thì trạng thái của thẻ a đó là visited.
* active: Khi ạn click vào thẻ a nhưng nhấn giữ chuột
* link: thẻ a nào bạn chưa click lần nào thì nó sẽ có tác dụng

Example:

|  |
| --- |
| a:hover{  color: red;  text-decoration: none;  }  a:visited{  color: yellow;  }  a:link{  color: red;  }  a:active{  color: yellow;  } |

## 2.6. CSS Font

Trong bài này mình sẽ tìm hiểu về font trong CSS và các thuộc tính liên quan.

Trước khi đi vào nội dung chính thì mình lấy một ví dụ cho các bạn dễ hiểu, đó là trong phần mềm soạn thảo Word ta có thể tùy chỉnh nhiều thuộc tính cho font như: in đậm, in nghiêng, kiểu chữ, kích thước. Ta gọi đây là những thông số của font, và bạn hoàn toàn có thể làm điều này trong CSS.

Để biết cách sử dụng font trong CSS như thế nào thì mình cùng điểm qua các thuộc tính chính của font nhé.

### a. Một số thuộc tính chính của font

CSS hỗ trợ rất nhiều thuộc tính font để định dạng chữ. Tuy nhiên hôm nay mình chỉ giới thiệu các thuộc tính hay sử dụng đó là: font-family, font-size, font-weight, font-style.

**font-family**

font-family là thuộc tính quyết định loại font nào sẽ được áp dụng.

Example:

|  |
| --- |
| p {  font-family: "Times New Roman", sans-serif, serif;  } |

**font-size**

font-size là thuộc tính dùng để thiết lập kích cỡ của chữ.

Example:

|  |
| --- |
| p.f12 {  font-size: 12px;  }  p.f15 {  font-size: 25px;  } |

**font-weight**

font-weight là thuộc tính dùng để thiết lập độ đậm của chữ.

font-weight có thể sử dụng một trong hai loại sau:

* Bằng số: 100, 200, 300, 400, 500, 600, 700,800, 900
* Bằng chữ: normal, bold, lighter....

Trong đó:

* normal: chữ sẽ hiển thị bình thường, đây cũng là giá trị mặc định và tương đương với 400.
* bold: chữ sẽ hiển thị  đậm, tương đương 700.
* bolder: chữ cái sẽ đậm hơn phần tử cha của nó.
* lighter: chữ cái sẽ nhạt hơn chữ cái của phần tử cha.

Độ đậm sẽ tăng dần từ 100 đến 900.

Example:

|  |
| --- |
| p.normal {  font-weight: normal;  }  p.bold {  font-weight: bold;  }  p.f100 {  font-weight: 100;  }  p.f900 {  font-weight: 900;  } |

**font-style**

font-style là thuộc tính dùng để thiết lập chữ in thường hoặc in nghiêng, đơn giản vậy thôi :)

Example:

|  |
| --- |
| p.normal {  font-style: normal;  }  p.italic {  font-style: italic;  } |

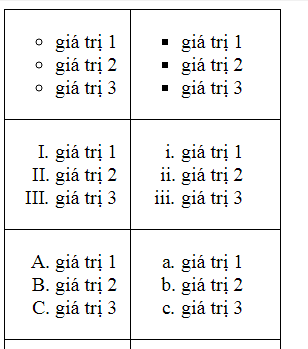
### b. Trang web font miễn phí

Sau đây là một số trang web có font miễn phí. Chúng cho chúng ta có các nhìn tổng quan về từng loại font sẽ được hiển thị trên trang web như thế nào.

* Google Fonts:
* Font Squirrel:
* Font Library:
* The League Of Move Able Type
* Exljbris:

## 2.7. CSS List

Mình sẽ lấy hình minh họa cho các bạn dễ hiểu, và mình cũng không biết giải thích khái niệm của danh sách như thế nào nữa. :)



List trong HTML có 2 loại chính đó là:

* Danh sách có thứ tự (ordered list). Đánh dấu mục của danh sách có thứ tự mặc định là số và bắt đầu bằng 1.
* Danh sách không có thứ tự (unordered list). Đánh dấu mục của danh sách không có thứ tự mặc định là dấu chấm màu đen.

CSS có những thuộc tính như list-style-type, list-style-image, list-style-position cho phép thay đổi biểu tượng đánh dấu danh mục của danh sách. Vậy cách sử dụng những thuộc tính này như thế nào, mình cùng xem tiếp phần 2 nhé.

### a. Một số thuộc tính chính của list

**list-style-type**

list-style-type là thuộc tính cho phép chỉ định loại đánh dấu mục của danh sách nào sẽ được hiển thị, ví dụ như là hình tròn, hình vuông, chữ la mã ...

list-style-type có rất nhiều giá trị, tuy nhiên mình chỉ giới thiệu những giá trị hay sử dụng sau:

* circle: giá trị hiển thị là hình tròn
* square:  giá trị hiển thị là hình vuông
* upper-roman: giá trị hiển thị là chữ la mã hoa như I, II, II, IV ...
* lower-roman: giá trị hiển thị là chữ la mã thường như i, ii, iii, iv ...
* upper-latin: giá trị hiển thị là chữ cái latin hoa như A, B, C ...
* lower-alpha: giá trị hiển thị là chữ cái alphabet như a, b, c ...
* none: không hiển thị gì cả
* decimal: giá trị hiển thị là số như 1, 2, 3 ...
* decimal-leading-zero: giá trị hiển thị là số nhưng sẽ thêm số 0 phía trước số, nếu số < 10 như 01, 02, 03 ...

Example:

|  |
| --- |
| **.**circle {  list-style-type: circle;  }  .square {  list-style-type: square;  }  .upper-roman {  list-style-type: upper-roman;  }  .lower-roman {  list-style-type: lower-roman;  }  .upper-latin {  list-style-type: upper-latin;  }  .lower-alpha {  list-style-type: lower-alpha ;  }  .none {  list-style-type: none;  }  .decimal {  list-style-type: decimal;  } |

**list-style-image**

list-style-image là thuộc tính dùng hình ảnh (image) để làm đánh dấu mục của danh sách.

list-style-position có 2 giá trị chính đó là:

* outside: đánh dấu mục của danh sách sẽ nằm bên ngoài các phần tử của danh sách
* inside: đánh dấu mục của danh sách sẽ nằm bên trong các phần tử của danh sách.

Example:

|  |
| --- |
|  |

**list-style-position**

**list-style**

## 2.8. CSS Table

### a. Các thuộc tính CSS định dạng bảng

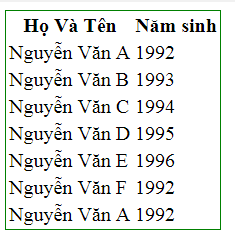
### border

border là thuộc tính dùng để thiết lập đường viền cho bảng (table), dòng (tr), ô dữ liệu (td) hay tiêu đề (th).

Example:

|  |
| --- |
| table {  border: 1px solid green;  } |

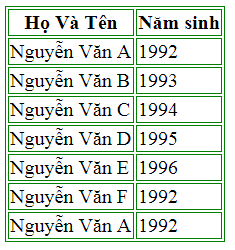
Result:



Example:

|  |
| --- |
| table, td, th {  border: 1px solid green;  } |

Result:



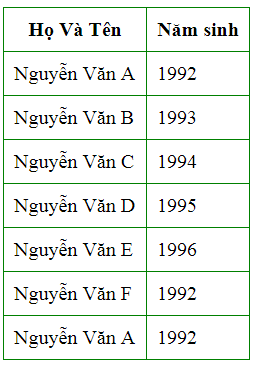
**padding**

padding là thuộc tính để thiết lập khoảng trắng giữa nội dung (text) của td, th.

Example:

|  |
| --- |
| td, th {  padding: 10px;  } |

Result:



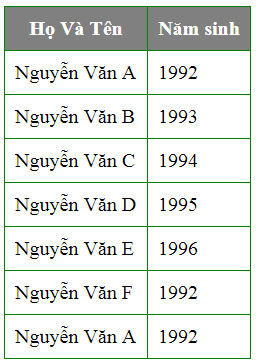
color, background-color

Color, background-color là 2 thuộc tính cho phép mình có thể thiết lập màu chữ (color), màu nền (background-color) cho td, th.

Example:

|  |
| --- |
| th {  background-color: gray;  color: white;  } |

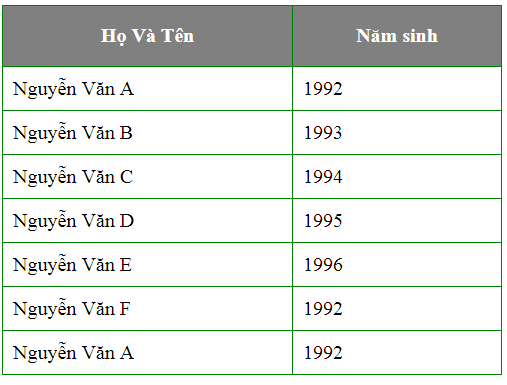
Result:



**width, height**

width, height là 2 thuộc tính cho phép mình thiết lập độ rộng và độ cao cho table, tr, th.

|  |
| --- |
| table {  width: 500px;  }  th {  height: 40px;  } |

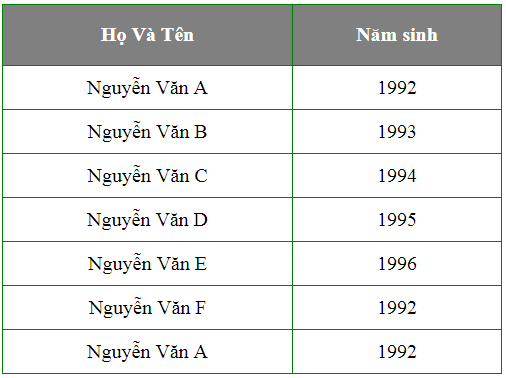


### text-align

text-align là thuộc tính cho phép mình chỉnh nội dung sang trái (left), phải (right) hoặc giữa (center) theo chiều ngang.

Example:

|  |
| --- |
| td {  text-align: center;  } |

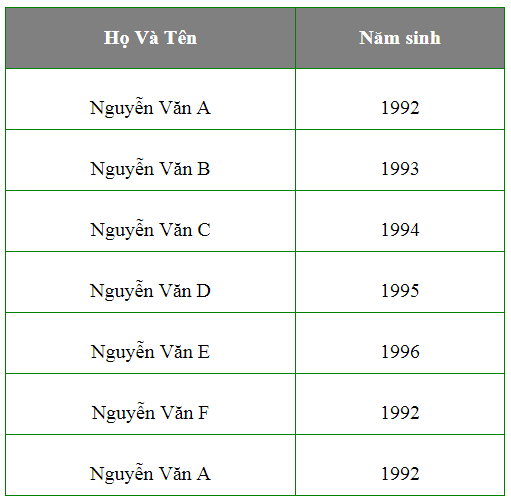


**vertical-align**

vertical-align là thuộc tính cho phép mình chỉnh nội dung ở trên (top), ở dưới (bottom) hay ở giữa (center) theo chiều dọc.

|  |
| --- |
| td {  text-align: center;  vertical-align: bottom;  height: 40px;  **}** |

Result:



**Hover**

hover cho phép mình thiết lập các thuộc tính CSS lên table, tr, th, td khi rê chuột vào chúng.

|  |
| --- |
| tr:hover {  background-color: green;  color: white;  } |

**nth-child**

nth-child là thuộc tính cho phép mình thiết lập các thuộc tính CSS lên dòng chẵn (even) hay dòng lẻ (odd).

Example:

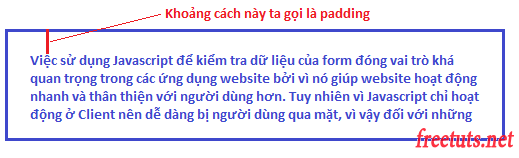
|  |
| --- |
| tr:nth-child(even) {background-color: yellow;}  tr:nth-child(odd) {background-color: green;} |

Result:



## 2.9. CSS Padding

Dùng để tạo khoảng giữa thẻ HTML và nội dung của nó, xem hình dưới đây:



Chúng ta có 5 thuộc tính của padding:

* **Padding-left: 20px** : khoảng cách lề trái so với nội dung bên trong 20px
* **Padding-top:20px** : khoảng cách lề trên so với nội dung bên trong 20px
* **Padding-right: 20px** : khoảng cách lề so với nội dung bên trong phải 20px
* **Padding-bottom: 20px** : khoảng cách so với nội dung bên trong lề dưới 20px
* **Padding : 20px** : tất cả lề trên, lề dưới, lề trái, lề phải so với nội dung bên trong đều có khoảng cách 20px

**Một lưu ý khác**: Nếu padding có 2 giá trị thì giá trị đầu được hiểu là bên trên và dưới còn giá trị thứ 2 được hiểu là trái và phải. Còn nếu có 3 giá trị thì giá trị đầu tiên là bên trên giá trị thứ 2 là bên trái và phải còn giá trị cuối cùng là bên dưới. Nếu còn 4 giá trị, giá trị đầu tiên là bên trên giá trị thứ 2 là bên phải giá trị thứ 3 là bên dưới còn giá trị cuối cùng là bên trái.

## 2.10. CSS Border

Border là thuộc tính CSS dùng để tạo đường viền bao quanh nội dung của phần tử HTML, nó nằm giữa padding và margin. Ở bài này các bạn khoan hãy quan tâm đến padding, margin là gì nha, mình sẽ có một bài khác nói về padding và margin.

**Thuộc tính của border:**

Bài này mình sẽ trình bày 4 thuộc tính quan trọng của border là: border-style, border-width, border-color và border-radius.

**Border-style:**

border-style là thuộc tính để thiết lập loại border nào sẽ được hiển thị.

Những style chính là border hổ trợ như sau:

* **dotted** - border sẽ hiển thị là những dấu chấm
* **dashed** - border sẽ hiển thị nét đứt
* **solid** - border sẽ hiển thị đường thẳng liền mạch
* **double** - border sẽ hiển thị 2 đường thẳng
* **groove** - border sẽ hiển thị dạng rãnh 3D.
* **ridge** - border sẽ hiển thị dạng viền 3D.
* **inset** - border sẽ hiển thị dạng viền trong 3D.
* **outset** - border sẽ hiển thị dạng viền đầu 3D.
* **none** - sẽ không có border
* **hidden** - border sẽ  bị ẩn.

**Border-width**

border-width là thuộc tính để thiết lập độ rộng của border. Các bạn có thể sử dụng CSS Unit như pt, px, em, rem ...  hoặc là có thể dùng 3 giá trị sau: thin, medium, thick.

**border-color**

border-color là thuộc tính để thiết lập màu sắc cho border.

Color có thể có giá trị là name, rgb, hex... Để tìm hiểu rõ hơn color trong CSS như thế nào mời các bạn xem bài [color trong CSS](https://freetuts.net/tim-hieu-ve-color-trong-css-1628.html), bài này mình sẽ không nhắc lại color trong CSS nữa.

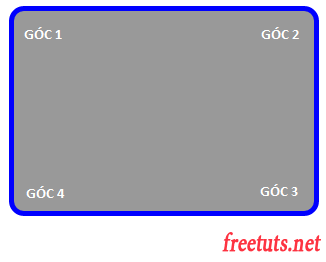
**Border-radius**

border-radius là để thiết lập bo tròn cho border.

Để tạo bo tròn đường viền bằng CSS3 thì ta sử dụng những cú pháp sau:

|  |
| --- |
| border-radius: 15px;  border-radius: 15px 15px;  border-radius: 15px 15px 15px;  border-radius: 15px 15px 15px; |

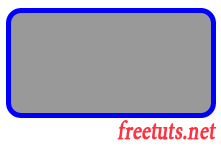
Trước khi vào vấn đề thì mình xin định nghĩa bốn góc như sau:



Và sau đây là các trường hợp sử dụng.

**Một tham số**

Trường hợp này sẽ có tác dụng cả bốn góc.



**Hai tham số**

Trường hợp này:

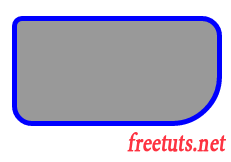
* Tham số đầu tiên là **GÓC 1** VÀ **GÓC 3**
* Tham số thứ hai là **GÓC 2** và **GÓC 4**.



**Ba tham số**

Trường hợp này:

* Tham số đầu tiên là **GÓC 1**
* Tham số thứ hai là **GÓC 2** và **GÓC 4**
* Tham số thứ ba là **GÓC 3**



**Bốn tham số**

Tương ứng với bốn góc

**Border Shorthand**

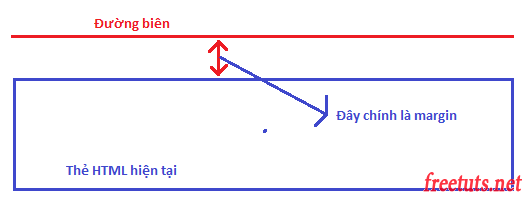
Border shorthand là cách viết ngắn gọn cho 3 thuộc tính border-width, border-style và border-color.

Cú pháp CSS:

Tuy nhiên thứ tự (width style color) mình có thể thay đổi được nha. 2 thuộc tính width và color thì không bắt buộc, còn thuộc tính style sẽ bắt buộc vì nếu không có thuộc tính style thì mấy thuộc tính khác cũng không được áp dụng.

## 2.11. CSS Margin

Dùng để tạo khoảng cách giữa hai thẻ HTML, xem hình dưới đây:



Chúng ta có 5 cách sử dụng như sau:

* **margin-left: 20px** : khoảng cách lề trái 20px
* **margin-top:20px** : khoảng cách lề trên 20px
* **margin-right: 20px** : khoảng cách lề phải 20px
* **margin-bottom: 20px** : khoảng cách lề dưới 20px
* **margin : 20px** : tất cả lề trên, lề dưới, lề trái, lề phải đều có khoảng cách 20px

**Một lưu ý khác**: Tương tự như padding. Nếu margin có 2 giá trị thì giá trị đầu được hiểu là bên trên và dưới còn giá trị thứ 2 được hiểu là trái và phải. Còn nếu có 3 giá trị thì giá trị đầu tiên là bên trên giá trị thứ 2 là bên trái và phải còn giá trị cuối cùng là bên dưới. Nếu còn 4 giá trị, giá trị đầu tiên là bên trên giá trị thứ 2 là bên phải giá trị thứ 3 là bên dưới còn giá trị cuối cùng là bên trái.

## 2.12. CSS Background

Trong bài này chúng ta tìm hiểu cách **thiết lập màu nền cho các thẻ HTML**, đây là một thuộc tính khá quan trọng mà bạn cần phải nắm vững vì hầu hết các trang web đều có sử dụng thuộc tính này để **tạo background** với các hình ảnh hoặc màu sắc (color). Không đi đâu xa mà tại website freetuts.net bạn thấy hai bên trái phải ngoài cùng có màu nền xám xám, đó là mình sử dụng thuộc tính background-image để thiết lập đấy.

Trong bài này chúng ta sẽ tìm hiểu một số thuốc tính backgrounds sau:

* background
* background-color
* background-image
* background-repeat
* background-attachment
* background-position

### a. CSS thiết lập màu nền cho background

Để thiết lập màu nền cho background thì ta sử dụng thuộc tính CSS backgroud-color hoặc background với giá trị của nó là màu sắc của background. Bạn có thể sử dụng mã màu hoặc tên màu bằng tiếng anh đều được.

Thẻ mà ta có thể thiết lập background toàn trang là thẻ body nên ta chỉ cần gán CSS cho thẻ body là được.

|  |
| --- |
| body{  background: red;  } |

Hoặc

|  |
| --- |
| body{  background: #fb0320;  } |

Hoặc

|  |
| --- |
| body{  background-color: #fb0320;  } |

### b. CSS thiết lập hình nền cho backround

Thiết lập hình nền thì ta sử dụng thuộc tính **CSS backgrond** hoặc background-image với tham số truyền vào là URL của hình ảnh.

|  |
| --- |
| body{  background: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  } |

Hoặc

|  |
| --- |
| body{  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  } |

### c. Cho phép lặp hoặc không lặp background

Như ở ví dụ phần 2 bạn thấy background là logo và nó lặp nhiều lần, vậy **làm thế nào không cho lặp background**? Ta sẽ sử dụng thuộc tính background-repeat và thuộc tính này gồm các giá trị:

* no-repeat : không lặp
* repeat : cho phép lặp
* repeat-x : lặp theo chiều ngang
* repeat-y : lặp theo chiều đứng

Vẫn còn vài thuộc tính nữa nhưng thông thường chúng ta sử dụng 4 thuộc tính này là đủ rồi.

**no-repeat**

|  |
| --- |
| body{  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  background-repeat: no-repeat;  } |

**Repeat**

|  |
| --- |
| body{  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  background-repeat: repeat;  } |

**repeat-x**

|  |
| --- |
| body{  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  background-repeat: repeat-x;  } |

**repeat-y**

|  |
| --- |
| body{  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  background-repeat: repeat-y;  } |

### d. Thiết lập vị trí hiển thị cho background

Trường hợp bạn sử dụng background không lặp và bạn muốn background hiển thị ở một ví trí nào đó như center, left, right, ... thì bạn sử dụng thuộc tính background-position. Cấu trúc của nó là:

|  |
| --- |
| background-position: position1 position2 |

Trong đó position1 và position2 gồm các giá trị sau:

* bottom: ở dưới
* left: bên trái
* right: bên phải
* center: ở giữa
* top: ở trên

Ví dụ: Hiện thị background ở giữa

|  |
| --- |
| body  {  height: 300px;  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  background-repeat: no-repeat;  background-position: center;  } |

Ví dụ: Hiện thị background  góc trái dưới

|  |
| --- |
| body  {  height: 300px;  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  background-repeat: no-repeat;  background-position: left bottom;  } |

### e. Thiết lập background đứng im khi scroll (fixed background)

Nếu màn hình dài quá thì khi bạn lăn chuột background sẽ kéo theo nên nếu bạn muốn nó đứng im thì có thể sử dụng thuộc tính background-attachment. Thông thường chúng ta sử dụng hai thuộc tính:

* fixed: sẽ đứng im
* scroll: sẽ di chuyển theo khi kéo

Ví dụ: Cho background đứng im khi lăn chuột

|  |
| --- |
| body  {  height: 1000px;  background-image: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png');  background-repeat: no-repeat;  background-attachment: fixed;  } |

### f. Sử dụng thuộc tính background nâng cao

Nếu bạn cảm thấy các thông số thiết lập background quá dài thì có thể sử dụng thuộc tính background ở dạng ghi tắt.

Ví dụ: ghi tắt với background

|  |
| --- |
| body  {  height: 1000px;  background: url('https://freetuts.net/upload/config/images/hoc-lap-trinh-online.png') no-repeat bottom fixed;  } |

### g. Sử dụng background-clip

Thuộc tính background-clip chỉ ra vùng vẽ của nền. Nó có thể nhận các giá trị:

* border-box mặc định, nền vẽ ngoài biên viền.
* padding-box nền vẽ cả ở vùng padding (không viền)
* content-box nền chỉ vẽ ở vùng nội dung phần tử

### h. Cấu hình kích thước cho background (background-size)

Thường thì chúng ta không thể thay đổi kích thước của background được nhưng với CSS3 thì hoàn toàn có thể bằng cách sử dụng thuộc tính background-size.

|  |
| --- |
| div  {  width: 620px;  height: 350px;  background:  url(https://freetuts.net/upload/tut\_post/images/2015/10/23/478/house.png) no-repeat center center,  url(https://freetuts.net/upload/tut\_post/images/2015/10/23/478/background.jpg) no-repeat center center;  background-size: 200px 200px, 300px 300px;  } |

Nếu chúng ta thiết lập kích thước cố định cho background như vậy thì không thể chạy tốt trong thiết kế responsive được vì responsive co giãn nên không có định chiều cao và chiều rộng. Lúc này ta sẽ lựa chọn sử dụng hai giá trị chuẩn sau đây:

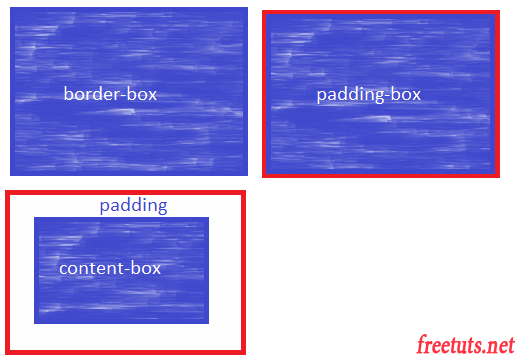
* **contain**: background sẽ có tác dụng như thẻ img, nghĩa là nó sẽ co giãn theo chiều rộng và chiều cao theo đúng tỉ lệ của hình ảnh.
* **cover**: Nếu chiều rộng và chiều cao của thẻ HTML lớn hơn hình ảnh thì nó sẽ giãn ra (full)

### i. Thuộc tính background-origin trong CSS3

Thuộc tính background-origin trong CSS3 dùng để xác đinh nơi mà **background hình ảnh** sẽ hiển thị. Nó có ba giá trị sau đây:

* **border-box**: biên của background tính luôn border ngoài cùng
* **padding-box**: biên của background tính từ vị trí padding (sát lề border)
* **content-box**: biên của backgroudn tính từ vị trí có thể sử dụng

Xem hình mô tả sau để dễ hiểu hơn.



Ví dụ

|  |
| --- |
| div{  width: 250px;  height: 100px;  margin: 20px auto;  border: solid 20px;  padding: 50px;  background: url(https://freetuts.net/upload/tut\_post/images/2015/10/23/478/hoa-hong-dep.gif) no-repeat top left;  }  #div1{  background-origin: border-box;  }  #div2{  background-origin: padding-box;  }  #div3{  background-origin: content-box;  } |

### k. CSS3 Multiple backgrouds

CSS3 cho phép thêm nhiều backgrounds cho một thẻ HTML bằng cách sử dụng thuộc tính background-image.

Nếu cú pháp của thuộc tính background là:

|  |
| --- |
| background : url(link-to-file.png) |

Thì cú pháp của multiple background là:

|  |
| --- |
| background :  url(link-to-file1.png),  url(link-to-file2.png)  ...; |

Lúc này để thiết lập các thuộc tính cho từng image thì ta sử dụng cấu trúc như sau:

|  |
| --- |
| background-position: top left, right bottom;  background-repeat: no-repeat, no-repeat |

Nghĩa là dùng dấu phẩy để thiết lập cho từng background tương ứng ở trên.

|  |
| --- |
| div  {  width: 620px;  height: 350px;  background-image:  url(https://freetuts.net/upload/tut\_post/images/2015/10/23/478/house.png),  url(https://freetuts.net/upload/tut\_post/images/2015/10/23/478/background.jpg);  background-position: center center, center center;  background-repeat: no-repeat, no-repeat  } |

Nếu bạn cảm thấy code trên dài dòng quá thì có thể sử dụng thuộc tính backgroud như ví dụ sau:

|  |
| --- |
| div  {  width: 620px;  height: 350px;  background:  url(https://freetuts.net/upload/tut\_post/images/2015/10/23/478/house.png) no-repeat center center,  url(https://freetuts.net/upload/tut\_post/images/2015/10/23/478/background.jpg) no-repeat center center;  } |

### j. CSS Background shorthand

Đây là cách viết ngắn ngọn của background. Thay vì viết dài như ở dưới.

|  |
| --- |
| body {  background-color: #ffffff;  background-image: url("img\_tree.png");  background-repeat: no-repeat;  background-position: right top;  } |

Mình có thể viết gắn hơn như ở dưới.

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

ở trên được viết tắt theo thứ tự là. Color, image, repeat, attachment và position.

## 2.13. CSS Function

### a. Var Function

Var function được dùng để thêm một giá trị vào biến của CSS.

Cú pháp:

|  |
| --- |
| var(--name, value) |

Example:

|  |
| --- |
| :root {  --main-bg-color: coral;  --main-txt-color: blue;  --main-padding: 15px;  }  #div1 {  background-color: var(--main-bg-color);  color: var(--main-txt-color);  padding: var(--main-padding);  }  #div2 {  background-color: var(--main-bg-color);  color: var(--main-txt-color);  padding: var(--main-padding);  } |

### b. Linear-gradient() Function

Hàm linear-gradient () đặt một vị trí tuyến tính làm hình nền.

Để tạo một vị trí tuyến tính, bạn phải xác định ít nhất hai điểm dừng màu. Điểm dừng màu là những màu bạn muốn tạo ra các chuyển tiếp mượt mà giữa các màu. Bạn cũng có thể đặt điểm bắt đầu và hướng (hoặc góc) cùng với hiệu ứng gradient.

Cú pháp:

|  |
| --- |
| background-image: linear-gradient(direction, color-stop1, color-stop2, ...); |

Example:

|  |
| --- |
| .grad {  background-image: linear-gradient(to right, red , blue);  }  .grad {  background-image: linear-gradient(to bottom right, red , blue);  }  .grad {  background-image: linear-gradient(180deg, red, blue);  }  .grad {  background-image: linear-gradient(to right, red,orange,yellow,green,blue,indigo,violet);  } |

### c. Rgba() Function

Hàm rgba có vai trò pha màu bằng cách kết hợp(red, green, blue, mức độ). Điều đặc biệt khi kết hợp hàm này với img nó sẽ cho một background có thể nhìn rõ ảnh trên background đó.

Giá trị màu RGBA là phần mở rộng của các giá trị màu RGB với alpha - chỉ định độ mờ của màu.

Cú pháp:

|  |
| --- |
| rgba(red, green, blue, alpha) |

Example:

|  |
| --- |
| .p1 {background-color:rgba(255,0,0,0.3);}  .p2 {background-color:rgba(0,255,0,0.3);}  .p3 {background-color:rgba(0,0,255,0.3);} |

### d. Rgb() Function

Rgb() khác với Rgba() ở chỗ nó không có alpha tức là nó không thể tạo ra màu trong suốt được, nó sẽ tạo ra màu đặc hoàn toàn.

Cú pháp:

|  |
| --- |
| rgb(red, green, blue) |

Example:

|  |
| --- |
| .p1 {background-color:rgb(255,0,0);}  .p2 {background-color:rgb(0,255,0);}  . p3 {background-color:rgb(0,0,255);} |

### e. Calc() Function

Sử dụng Calc() để tính toán độ rông(width) trong thẻ div.

Cú pháp:

|  |
| --- |
| calc(expression) |

Example:

|  |
| --- |
| .div1 {  position: absolute;  left: 50px;  width: calc(100% - 100px);  border: 1px solid black;  background-color: yellow;  padding: 5px;  text-align: center;  } |

### F. Attr() Function

