

## BÀI 3: CÁC ĐỊNH DẠNG CSS CƠ BẢN

### 3.1. Giới thiệu

CSS là viết tắt của cụm từ "Cascading Style Sheet", nó là một ngôn ngữ quy định cách trình bày của các thẻ HTML trên trang web. Là ngôn ngữ đang được sử dụng rất nhiều trong lập trình web, có thể nói CSS ra đời đã tạo nên một cuộc cách mạng ở các website trên toàn thế giới.

Có thể thấy CSS có thể quản lý toàn bộ các thành phần (giao diện) của trang web thông qua 1 file CSS, điều khiển chính xác hơn cách trình bày Layout. Quan trọng nhất là tính tái sử dụng để áp dụng cho nhiều thành phần giống nhau. Sử dụng được các kỹ thuật tiên tiến, phức tạp mà HTML không làm được.

Để học CSS cần có kiến thức về HTML, nó không thật sự cần thiết nếu bạn chỉ dùng CSS để trình bày cho một trang HTML có sẵn, nhưng bạn vẫn cần biết ý nghĩa một số thẻ HTML, nó sẽ có ích khi bạn viết CSS. Tuy nhiên, nếu bạn muốn tự thiết kế, trình bày một trang web của riêng mình thì tùy theo quy mô trang web, bạn cần phải học thêm cả HTML, XHTML, Javascript và một số ngôn ngữ lập trình web khác.

Để viết mã CSS có thể sử dụng một trình soạn thảo đơn giản như Notepad, Wordpad trong Windows hay Pico trong Linux, Simple Text trong Mac, ... tuy nhiên được sử dụng phổ biến nhất hiện nay vẫn là công cụ Macro Dreamweaver. Ngoài ra, để có thể hiển thị tốt nhất cách thể hiện của một trang HTML kết hợp với CSS nên sử dụng trình duyệt phiên bản mới nhất.

### 3.2. Một số quy ước về cách viết CSS

#### 3.2.1. Cú pháp CSS

Để định màu nền cho một trang web là xanh nhạt (light cyan) chúng ta dùng đoạn mã lệnh sau:

+ Trong HTML: `<body bgcolor="#00BFF3">`

+ Trong CSS: `body {background-color:#00BFF3; }`

Nhìn qua ví dụ trên ít nhiều chúng ta cũng thấy được mối tương đồng giữa các thuộc tính trong HTML và CSS, cho nên nếu bạn đã học qua HTML thì cũng sẽ rất dễ dàng tiếp thu CSS.

**Cú pháp CSS cơ bản:**

**Tên\_thẻ{Thuộc\_tính:giá\_trị;}**

Hoặc

**.Tên\_class{Thuộc\_tính:giá\_trị;}**

Hoặc

**#Tên\_ID{Thuộc\_tính:giá\_trị;}**

Trong đó:

- + Tên thẻ: Là tên một thẻ HTML mà người lập trình muốn CSS tương tác;
- + Tên class: Tên lớp bất kỳ do người lập trình định nghĩa được sử dụng nhiều lần trên trang HTML;
- + Tên ID: Tên của một phần tử ID bất kỳ chỉ được sử dụng 1 lần duy nhất trên trang HTML.

Ví dụ trong HTML ta có đoạn mã như:

```
<input name="Search" type="Text" value="Key Word">
```

Để áp dụng thuộc tính CSS cho riêng ô tìm kiếm này chúng ta sẽ dùng thẻ `input[name="Search"]`.

Ngoài việc viết tên thẻ cụ thể, cũng có thể dùng một thẻ đại diện như `*{ color:red }` sẽ tác động đến tất cả các thành phần có trên trang web làm cho chúng có text màu đỏ.

- + Thuộc tính: Chính là các thuộc tính quy định cách trình bày. Ví dụ: `background-color`, `font-family`, `color`, `padding`, `margin`,...

Mỗi thuộc tính CSS phải được gán một giá trị. Nếu có nhiều hơn một thuộc tính cho một thẻ thì chúng ta phải dùng một dấu `;` (chấm phẩy) để phân cách các thuộc tính. Tất cả các thuộc tính trong một selector sẽ được đặt trong một cặp ngoặc nhọn sau thẻ.

*Ví dụ:*

```
body { background:#FFF; color:#FF0000; font-size:14pt }
```

Để dễ đọc hơn, bạn nên viết mỗi thuộc tính CSS ở một dòng. Tuy nhiên, nó sẽ làm tăng dung lượng lưu trữ CSS của bạn.

*Ví dụ:*

```
body {  
    background:#FFF;  
    color:#FF0000;  
    font-size:14pt  
}
```

Đối với một trang web có nhiều thành phần có cùng một số thuộc tính, chúng ta có thể thực hiện gom gọn lại như sau:

```
h1 { color:#0000FF;  
    text-transform:uppercase }  
h2 {  
    color:#0000FF;  
    text-transform:uppercase;  
}  
h3 {
```

```

        color:#0000FF;
        text-transform:uppercase;
    }
=> h1, h2, h3 {
        color:#0000FF;
        text-transform:uppercase;}

```

Đối với một giá trị có khoảng trắng, bạn nên đặt tất cả trong một dấu ngoặc kép.

*Ví dụ:* font-family:"Times New Roman"

Đối với các giá trị là đơn vị đo, không nên đặt một khoảng cách giữa số đo với đơn vị của nó. Nó sẽ làm CSS của bạn bị vô hiệu trên Mozilla Firefox hay Netscape.

*Chú thích trong CSS:* Cũng như nhiều ngôn ngữ web khác. Trong CSS, chúng ta cũng có thể viết chú thích cho các đoạn code để dễ dàng tìm, sửa chữa trong những lần cập nhật sau. Chú thích trong CSS được viết như sau /\* Nội dung chú thích \*/

*Ví dụ:*

```

/* Màu chữ cho trang web */
body {
    color:red
}

```

Nhóm phần tử với class cho phép định dạng style của các đối tượng (table, td, div, span ...) Có thể sử dụng lặp đi lặp lại nhiều lần trong cùng một file HTML. Để thiết lập phần tử là class ta sử dụng dấu . (chấm) để thực hiện khai báo.

Ví dụ chúng ta có một đoạn mã HTML sau đây :

```

<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
    <li>Hà Nội</li>
    <li>TP. Hồ Chí Minh</li>
    <li>Đà Nẵng</li>
    <li>Thừa Thiên Huế</li>
    <li>Khánh Hòa</li>
    <li>Quảng Ninh</li>
    <li>Tiền Giang</li>
</ul>

```

Yêu cầu đặt ra là làm thế nào để tên các thành phố là màu đỏ và tên các tỉnh là màu xanh da trời. Để giải quyết vấn đề này chúng ta sẽ dùng một thuộc tính HTML gọi là class để tạo thành 2 nhóm là thành phố và tỉnh. Ta sẽ viết lại đoạn HTML thành:

```

<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
    <li class="tp">Hà Nội</li>
    <li class="tp">TP. Hồ Chí Minh</li>
    <li class="tp">Đà Nẵng</li>

```

```

        <li class="tinh">Thừa Thiên Huế</li>
        <li class="tinh">Khánh Hòa</li>
        <li class="tinh">Quảng Ninh</li>
        <li class="tinh">Tiền Giang</li>
    </ul>

```

Với việc dùng class để nhóm các đối tượng như trên thì công việc của chúng ta sẽ trở nên đơn giản hơn nhiều:

```

li .tp {color:FF0000}
li .tinh {color:0000FF}

```

Cũng giống như **class** nhưng **ID** chỉ được sử dụng một lần nếu sử dụng nhiều lần một ID thì vẫn được tuy nhiên nó không đúng chuẩn của w3c và sẽ gặp rắc rối khi dùng ID trong javascript. Việc sử dụng phần tử ID ta sử dụng dấu # để thực hiện khai báo cho ID.

*Ví dụ:* Cũng với đoạn HTML như ví dụ về class. Nhưng yêu cầu đặt ra là Hà Nội sẽ có màu đỏ sậm, TP. Hồ Chí Minh màu đỏ, Đà Nẵng màu đỏ tươi còn các tỉnh màu xanh da trời. Để giải quyết vấn đề này chúng ta sẽ sử dụng thuộc tính HTML là ID để nhận dạng mỗi thành phố và dùng class để nhóm các tỉnh. Đoạn HTML của chúng ta bây giờ sẽ là:

```

<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
    <li id="hanoi">Hà Nội</li>
    <li id="hcmc">TP. Hồ Chí Minh</li>
    <li id="danang">Đà Nẵng</li>
    <li class="tinh">Thừa Thiên Huế</li>
    <li class="tinh">Khánh Hòa</li>
    <li class="tinh">Quảng Ninh</li>
    <li class="tinh">Tiền Giang</li>
</ul>

```

Và đoạn CSS cần dùng sẽ là:

```

#hanoi { color:# 790000 }
#hcmc { color:#FF0000 }
#danang { color:#FF00FF }
.tinh { color:#0000FF }

```

**Lưu ý:** Không nên đặt tên ID và class với ký tự đầu là chữ số, nó sẽ không làm việc cho Firefox.

### 3.2.2. Đơn vị CSS

Trong CSS hỗ trợ các loại đơn vị là đơn vị đo chiều dài và đơn vị đo góc, thời gian, cường độ âm thanh và màu sắc. Tuy nhiên, sử dụng phổ biến nhất vẫn là đơn vị đo chiều dài và màu sắc. Sau đây là bảng liệt kê các đơn vị chiều dài dùng trong CSS.

Đơn vị chiều dài:

Đơn vị	Mô tả	Đơn vị	Mô tả
%	Phần trăm	Pt	Point (1pt = 1/72 inch)
in	Inch (1 inch=2,54cm)	Pc	Pica (1 pc = 12 pt)
cm	Centimeter	px	Pixels (điểm ảnh trên màn hình máy tính)
mm	Milimeter	ex	1 ex bằng chiều cao của chữ x in thường của font hiện hành. Do đó, đơn vị này không những phụ thuộc trên kích cỡ font chữ mà còn phụ thuộc vào loại font chữ vì cùng 1 cỡ 14px nhưng chiều cao của font Time New Roman và font Tahoma khác nhau
em	1 em tương đương kích thước font chữ hiện hành, nếu font chữ hiện hành là 14px thì 1em = 14px. Đây là đơn vị đo rất hữu ích cho trang web		

Đơn vị màu sắc:

Đơn vị	Mô tả
Color-name	Tên màu bằng tiếng Anh. Ví dụ: green, blue, red, black ...
RGB (r,g,b)	Màu RGB tương ứng với 03 giá trị R, G, B (red, green, blue) có giá trị từ 0 đến 255 kết hợp với nhau tạo ra màu.
RGB (%r,%g,%b)	Màu RGB với 03 giá trị R,G,B có giá trị từ 0 đến 100% kết hợp với nhau để tạo ra màu.
Hexadecimal RGB	Mã màu RGB với cách thể hiện bằng chữ số thập lục phân (hexa) với 2 ký tự đầu dành cho màu đỏ (Red), 02 ký tự kế tiếp dành cho màu xanh lá cây (Green) và 02 ký tự cuối dành cho màu xanh da trời (Blue). Ví dụ: #FFFFFF là màu trắng, #000000 là màu đen, #FF0000 là màu đỏ.

### 3.2.3. Vị trí đặt CSS

**Cách 1:** Nội tuyến (kiểu thuộc tính) đây là 1 phương pháp nguyên thủy nhất để nhúng CSS vào 1 trang HTML. Bằng cách nhúng vào từng thẻ HTML muốn áp dụng và dĩ nhiên trong trường hợp này chúng ta sẽ không cần thẻ trong cú pháp.

Lưu ý: Nếu bạn muốn áp dụng nhiều thuộc tính cho nhiều thẻ HTML khác nhau thì không nên dùng cách này.

Ở ví dụ sau chúng ta sẽ tiến hành định nền màu trắng cho trang và màu chữ xanh lá cho đoạn văn bản như sau:

VD:

```
<html>
  <head><title>Vi du</title></head>
```

```

<body style="background-color:#FFF;">
    <p style="color:green"> Học lập trình thật là
    thú vị </p>
</body>
</html>

```

### **Cách 2:** Bên trong (thẻ style)

Thật ra nếu nhìn kỹ chúng ta cũng nhận ra đây chỉ là một phương cách thay thế cách thứ nhất bằng cách rút tất cả các thuộc tính CSS vào trong thẻ style (để tiện cho công tác bảo trì, sửa chữa).

Cũng ví dụ làm trang web có màu nền trắng, đoạn văn bản chữ xanh lá, chúng ta sẽ thể hiện như sau:

```

<html>
    <head><title>vi du </title></head>
    <style type="text/css">
        body {background-color:#FFF}\
        p {color:#00FF00}
    </style>
    <body>
        <p> Hoc lap trinh that la thu vi </p>
    </body>
</html>

```

**Lưu ý:** Thẻ style nên đặt trong thẻ head.

Đối với những trình duyệt cũ, không thể nhận ra thẻ <style>. Theo mặc định, thì khi một trình duyệt không nhận ra một thẻ thì nó sẽ hiện ra phần nội dung chứa trong thẻ. Như ở ví dụ trên, nếu trình duyệt không hỗ trợ thẻ style thì 2 dòng CSS: body {background-color:#FFF } p { color:#00FF00 } sẽ hiện ra trên trình duyệt.

Để tránh tình trạng này, bạn nên đưa vào thêm dấu <!-- ở trước và --> ở sau khối code CSS. Như ví dụ trên sẽ viết lại là:

```

<style type="text/css">
    <!-- body { background-color:#FFF }
    p { color:#00FF00 } -->
</style>

```

### **Cách 3:** Bên ngoài (liên kết với một file CSS bên ngoài)

Tương tự như cách 2, thay vì đặt tất cả mã CSS trong thẻ style chúng ta sẽ đưa chúng vào trong 1 file CSS (có đuôi là .css) bên ngoài là liên kết nó vào trong web bằng thuộc tính href trong thẻ link. Đây là cách làm tối ưu nhất, được khuyến cáo, nó đặc biệt hữu ích cho việc đồng bộ hay bảo trì một website lớn sử dụng cùng một kiểu.

Đầu tiên chúng ta sẽ tạo ra một file vidu.html có nội dung như sau:

```

<html>
  <head><title>Vi du</title>
    <link      rel="stylesheet"      type="text/css"
    href="style.css">
  </head>
<body>
  <p>Hoc lap trinh that thu vi</p>
</body>
</html>

```

Sau đó, ta tạo 1 file style.css với nội dung:

```

body{background-color: #FFF }
p{color: #00FF00}

```

### 3.2.4. Sự ưu tiên

Trước khi thực thi CSS cho 1 trang web, trình duyệt sẽ đọc toàn bộ CSS mà trang web có thể được áp dụng, bao gồm CSS mặc định của trình duyệt, file CSS bên ngoài, CSS nhúng vào thẻ <style> và các CSS nội tuyến. Sau đó, trình duyệt sẽ tổng hợp lại vào 1 CSS ảo, và nếu thuộc tính CSS giống nhau thì thuộc tính nào nằm sau sẽ được ưu tiên sử dụng. Vậy thứ tự sẽ là : CSS nội tuyến (cách 1), CSS bên trong (cách 2), CSS bên ngoài (cách 3), CSS mặc định của trình duyệt.

Ví dụ: Trong một trang web có liên kết tới **file style.css** có nội dung như sau:

```

p {
    color:#333;
    text-align:left;
    width:500px
}

```

Trong **thẻ <style>** giữa thẻ <head> cũng có một đoạn CSS liên quan:

```

p {
    background-color:#FF00FF;
    text-align:right;
    width:100%;
    height:150px
}

```

Trong phần nội dung trang web đó cũng có sử dụng **CSS nội tuyến**:  
 <p style="height:200px; text-align:center; border:1px solid #FF0000; color:#000" >

Khi thực thi CSS trình duyệt sẽ đọc hết tất cả các nguồn chứa style rồi sẽ tổng hợp lại vào một CSS ảo và nếu có sự trùng lặp các thuộc tính CSS thì nó sẽ lấy thuộc tính

CSS có mức ưu tiên cao hơn. Như ví dụ trên chúng ta sẽ thấy CSS cuối cùng mà phần tử p nhận được là:

```
p {  
    background-color:#FF00FF;  
    width:100%;  
    height:200px;  
    text-align:center;  
    border:1px solid #FF0000;  
    color:#000  
}
```

Vậy, có cách nào để thay đổi độ ưu tiên cho một thuộc tính nào đó? Thật ra thì trong CSS đã có sẵn một thuộc tính giúp chúng ta thực hiện điều này, đó chính là thuộc tính !important. Chỉ cần bạn đặt thuộc tính này sau một thuộc tính nào đó theo cú pháp selector {property:value !important} thì trình duyệt sẽ hiểu đây là một thuộc tính được ưu tiên. Bây giờ, chúng ta cùng xét lại ví dụ trên nhưng có đặt thêm một số thuộc tính !important vào xem kết quả như thế nào nhé.

```
p {  
    width:500px;  
    text-align:left !important;  
    color:#333 !important  
}
```

```
p {  
    background-color:#FF00FF;  
    width:100%;  
    height:150px !important;  
    text-align:right;  
}
```

```
<p style="text-align:center; height:200px; border:1px  
solid #FF0000; color:#000" }
```

Phần CSS sẽ tác động lên thuộc tính p là:

```
p {  
    background-color:#FF0000;  
    width:100%;  
    height:150px !important;  
    text-align:left !important;  
    border:1px solid #FF0000;  
    color:#333 !important }
```



**Lưu ý:** Cùng một thuộc tính cho một selector thì nếu cả hai thuộc tính đều đặt !important thì thuộc tính sau được lấy.

### 3.3. Màu chữ và màu nền

**Thuộc tính background-color:** giúp định màu nền cho một thành phần trên trang web. Các giá trị mã màu của background-color cũng giống như color nhưng có thêm giá trị transparent để tạo nền trong suốt.

Ví dụ sau đây sẽ chỉ cho chúng ta biết cách sử dụng thuộc tính background-color để định màu nền cho cả trang web, các thành phần h1, h2 lần lượt là xanh lơ, đỏ và cam.

```
body {background-color:cyan }
h1 {background-color:red}
h2{background-color:orange}
```

#### **Thuộc tính background-image**

Việc sử dụng ảnh nền giúp trang web trông sinh động và bắt mắt hơn. Để chèn ảnh nền vào một thành phần trên trang web, chúng ta sử dụng thuộc tính background-image.

Bây giờ chúng ta sẽ cùng làm một ví dụ minh họa để xem thuộc tính background-image sẽ hoạt động ra sao. Đầu tiên hãy tìm một tấm ảnh mà bạn thích, ở đây tôi sẽ lấy tấm ảnh logo của Khoa Công nghệ thông tin Trường Đại học Công nghiệp Hà Nội.

Sau đó, chúng ta sẽ viết CSS để đặt logo này làm ảnh nền trang web như sau:

```
h1 {background-color:red}
h2 {background-color:orange}
p { background-color: FDC689}
```

Như các bạn đã thấy chúng ta sẽ phải chỉ định đường dẫn của ảnh trong cặp ngoặc đơn sau url. Do ảnh đặt trong cùng thư mục với file style3.css nên chúng ta chỉ cần ghi cntt-hau1.jpg. Nhưng nếu chúng ta tạo thêm một thư mục img trong thư mục thì chúng ta sẽ phải ghi là

```
background-image:url(img/cntt-hau1.jpg)
```

Đôi khi nếu không chắc chắn bạn có thể dùng đường dẫn tuyệt đối cho ảnh.

#### **Thuộc tính background-repeat**

Nếu sử dụng một ảnh có kích thước quá nhỏ để làm nền cho một đối tượng lớn hơn thì theo mặc định trình duyệt sẽ lặp lại ảnh nền để phủ kín không gian còn thừa. Thuộc tính background-repeat cung cấp cho chúng ta các điều khiển giúp kiểm soát trình trạng lặp lại của ảnh nền. Thuộc tính này có 4 giá trị:

- + repeat-x: Chỉ lặp lại ảnh theo phương ngang.
- + repeat-y: Chỉ lặp lại ảnh theo phương dọc.
- + repeat: Lặp lại ảnh theo cả 2 phương, đây là giá trị mặc định.
- + no-repeat: Không lặp lại ảnh.

Bây giờ, chúng ta hãy thêm thuộc tính background-repeat này vào ví dụ trên:

```
body {
```

```
background-image:url (cntt-hau1.png);
background-repeat:no-repeat; }
```

Các bạn xem, có phải ảnh nền đã không bị lặp lại như trong ví dụ trước, hãy thử thay đổi qua lại giữa các giá trị và xem kết quả tạo ra.

### Thuộc tính background-attachment

Background-attachment là một thuộc tính cho phép bạn xác định tính cố định của ảnh nền so với nội dung trang web. Thuộc tính này có 2 giá trị:

- scroll: Ảnh nền sẽ cuộn cùng nội dung trang web, đây là giá trị mặc định.
- fixed: Cố định ảnh nền so với nội dung trang web. Khi áp dụng giá trị này, ảnh nền sẽ đứng yên khi bạn đang cuộn trang web.

### Thuộc tính background-position

Theo mặc định ảnh nền khi được chèn sẽ nằm ở góc trên, bên trái màn hình. Tuy nhiên với thuộc tính background-position bạn sẽ có thể đặt ảnh nền ở bất cứ vị trí nào (trong không gian của thành phần mà nó làm nền).

Background-position sẽ dùng một cặp 2 giá trị để biểu diễn tọa độ đặt ảnh nền. Có khá nhiều kiểu giá trị cho thuộc tính position. Như đơn vị chính xác: centimeters, pixels, inches,... hay các đơn vị qui đổi như %, hoặc các vị trí đặt biệt như top, bottom, left, right.

Giá trị một số thuộc tính của Background:

GIÁ TRỊ	Ý NGHĨA
Background-position: 5cm 2cm	Ảnh được định vị 5cm từ trái qua và 2cm từ trên xuống.
Background-position: 20% 30%	Ảnh được định vị 20% từ trái qua và 30% từ trên xuống.
Background-position: bottom left	Ảnh được định vị ở góc trái phía dưới

**Thuộc tính background rút gọn:** Khi sử dụng quá nhiều thuộc tính CSS sẽ gây khó khăn cho người đọc, công tác chỉnh sửa cũng như tốn nhiều dung lượng ổ cứng cho nên CSS đưa ra một cấu trúc rút gọn cho các thuộc tính cùng nhóm.

Chúng ta có thể nhóm lại đoạn CSS sau:

```
background-color:transparent;
background-image: url (cntt-hau1.png);
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
```

Thành một dòng ngắn gọn:

```
background:transparent url (cntt-hau1.png) no-repeat
fixed right bottom;
```

Từ ví dụ trên chúng ta có thể khái quát cấu trúc rút gọn cho nhóm background:

```
background:<background-color> | <background-image> |  
<background-repeat> | <background-attachment> |  
<background-position>
```

Theo mặc định thì các thuộc tính không được đề cập sẽ nhận các giá trị mặc định. Chúng ta sẽ bỏ qua hai thuộc tính background-attachment và background-position ở dòng mã trên đi:

```
background:transparent url(cntt-hau1.png) no-repeat;
```

Hai thuộc tính không được chỉ định sẽ đơn thuần được thiết lập tới giá trị mặc định mà chúng ta điều biết là scroll và top left.

### 3.4. Font chữ

**Thuộc tính font-family:** có công dụng định nghĩa một danh sách ưu tiên các font sẽ dùng để hiển thị một thành phần trang web. Theo đó, thì font đầu tiên được liệt kê trong danh sách sẽ được dùng để hiển thị trang web. Nếu như trên máy tính truy cập chưa cài đặt font này thì font thứ hai trong danh sách sẽ được ưu tiên... cho đến khi có một font phù hợp.

Có hai loại tên font được dùng để chỉ định trong font-family: family-names và generic families.

+ Family-names: Tên cụ thể của một font. Ví dụ: Arial, Verdana, Tohama,...

+ Generic families: Tên của một họ gồm nhiều font. Ví dụ: sans-serif, serif,...

Khi lên danh sách font dùng để hiển thị một trang web ta sẽ chọn những font mong muốn trang web sẽ được hiển thị để đặt ở các vị trí ưu tiên. Tuy nhiên, có thể những font này sẽ không thông dụng lắm nên cũng cần chỉ định thêm một số font thông dụng như Arial, Tohama hay Times New Roman. Thực hiện theo cách này thì sẽ đảm bảo trang web của bạn có thể hiển thị tốt trên bất kỳ hệ thống nào.

Ví dụ sau sẽ viết CSS để quy định font chữ dùng cho cả trang web là Times New Roman, Tohama, sans-serif và font chữ dùng để hiển thị các tiêu đề h1, h2, h3 sẽ là Arial, Verdana và các font họ serif.

```
body { font-family:"Times New Roman",Tohama,sans-serif }  
h1, h2, h3 { font-family:arial,verdana,serif }
```

Mở trang web trong trình duyệt và kiểm tra kết quả. Chúng ta thấy phần tiêu đề sẽ được ưu tiên hiển thị bằng font Arial, nếu trên máy không có font này thì font Verdana sẽ được ưu tiên và kế đó sẽ là các font thuộc họ serif.

**Chú ý:** Đối với các font có khoảng trắng trong tên như Times New Roman cần được đặt trong dấu ngoặc kép.

**Thuộc tính font-style:** Định nghĩa việc áp dụng các kiểu in thường (normal), in nghiêng (italic) hay xiên (oblique) lên các thành phần trang web. Trong ví dụ bên dưới chúng ta sẽ thử thực hiện áp dụng kiểu in nghiêng cho thành phần h1 và kiểu xiên cho h2.

Ví dụ:

```
h1 {font-style:italic; }
```

```
h2 {font-style:oblique; }
```

**Thuộc tính font-variant:** Được dùng để chọn giữa chế độ bình thường và small-caps của một font chữ. Một font small-caps là một font sử dụng chữ in hoa có kích cỡ nhỏ hơn in hoa chuẩn để thay thế những chữ in thường. Nếu như font chữ dùng để hiển thị không có sẵn font small-caps thì trình duyệt sẽ hiện chữ in hoa để thay thế.

Trong ví dụ sau, chúng ta sẽ sử dụng kiểu small-caps cho phần h1:

```
h1 {font-variant:small-caps}
```

**Thuộc tính font-weight:** mô tả cách thức thể hiện của font chữ là ở dạng bình thường (normal) hay in đậm (bold). Ngoài ra, một số trình duyệt cũng hỗ trợ mô tả độ in đậm bằng các con số từ 100 - 900.

*Ví dụ:* `p{font-weight:bold}`

**Thuộc tính font-size:** Kích thước của một font được định bởi thuộc tính font-size. Thuộc tính này nhận các giá trị đơn vị đo hỗ trợ bởi CSS bên cạnh các giá trị xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger. Tùy theo mục đích sử dụng của website có thể lựa chọn những đơn vị phù hợp. Ví dụ trang web phục vụ chủ yếu là những người già, thị lực kém hay những người dùng sử dụng các màn hình máy tính kém chất lượng có thể cân nhắc sử dụng các đơn vị qui đổi như em hay %. Như vậy, sẽ đảm bảo font chữ trên trang web luôn ở kích thước phù hợp.

Ở ví dụ sau trang web sẽ có kích cỡ font là 20px, h1 là  $3em = 3 \times 20 = 60px$ , h2 là  $2em = 40px$ .

```
body {font-size:20px}
```

```
h1{font-size:3em }
```

```
h2 { font-size:2em }
```

**Thuộc tính font rút gọn:** Tương tự như các thuộc tính background, chúng ta cũng có thể rút gọn các thuộc tính font lại thành một thuộc tính đơn như ví dụ sau:

```
h1 {  
    font-style: italic;  
    font-variant:small-caps;  
    font-weight: bold;  
    font-size: 35px;  
    font-family: arial,verdana,sans-serif;  
}
```

Thành

```
h1 {  
    font: italic bold 35px arial,verdana,sans-serif;  
}
```

Cấu trúc rút gọn cho các thuộc tính nhóm font:

Font :<font-style> | <font-variant> | <font-weight> | <font-size> |<font-family>

### 3.5. Text

**Thuộc tính color:** Để định màu chữ cho một thành phần nào đó trên trang web, chúng ta sử dụng thuộc tính color. Giá trị của thuộc tính này là các giá trị màu CSS hỗ trợ. Ví dụ sau sẽ viết CSS để định màu chữ chung cho một trang web là đen, cho tiêu đề h1 màu xanh da trời, cho tiêu đề h2 màu xanh lá, chúng ta sẽ làm như sau:

```
body {color:#000}  
h1 {color:#0000FF }  
h2 {color:#00FF00 }
```

**Thuộc tính text-indent:** Cung cấp khả năng tạo ra khoảng thụt đầu dòng cho dòng đầu tiên trong đoạn văn bản. Giá trị thuộc tính này là các đơn vị đo cơ bản dùng trong CSS.

Trong ví dụ sau sẽ định dạng thụt đầu dòng một khoảng 30px cho dòng văn bản đầu tiên trong mỗi đoạn văn bản đối với các thành phần <p>. Ví dụ: p {text-indent:30px}

**Thuộc tính text-align:** Thêm các canh chỉnh văn bản cho các thành phần trong trang web. Cũng tương tự như các lựa chọn canh chỉnh văn bản trong các trình soạn thảo văn bản thông dụng như MS Word, thuộc tính này có tất cả 4 giá trị: left (canh trái – mặc định), right (canh phải), center (canh giữa) và justify (canh đều).

Trong ví dụ sau sẽ thực hiện canh phải các thành phần h1, h2 và canh đều đối với thành phần <p>

```
h1, h2 {text-align:right}  
p {text-align:justify}
```

**Thuộc tính letter-spacing:** Được dùng để định khoảng cách giữa các ký tự trong một đoạn văn bản. Muốn định khoảng cách giữa các ký tự trong thành phần h1, h2 là 7px và thành phần <p> là 5px, chúng ta sẽ viết CSS sau:

```
h1, h2 {letter-spacing:7px}  
p {letter-spacing:5px}
```

**Thuộc tính text-decoration:** thêm các hiệu ứng gạch chân (underline), gạch xiên (line-through), gạch đầu (overline), và một hiệu ứng đặc biệt là văn bản nhấp nháy (blink). Ví dụ sau sẽ định dạng gạch chân cho thành phần h1, gạch đầu thành phần h2:

```
h1 {text-decoration:underline}  
h2 {text-decoration:overline}
```

**Thuộc tính text-transform:** Là thuộc tính qui định chế độ in hoa hay in thường của văn bản mà không phụ thuộc vào văn bản gốc trên HTML. Thuộc tính này có tất cả 4 giá trị: uppercase (in hoa), lowercase (in thường), capitalize (in hoa ở ký tự đầu tiên trong mỗi từ) và none (không áp dụng hiệu ứng - mặc định).

Trong ví dụ dưới đây sẽ định dạng cho thành phần h1 là in hoa, h2 là in hoa đầu mỗi ký tự.

```
h1 {text-transform:uppercase}
```

```
h2 {text-transform:capitalize}
```