**CSS (10) - Định dạng màu**

**1         Định dạng màu & nền**

* 1. Định dạng màu

**Xác định màu**

Có hai cách phổ biến để xác định một màu, một là dùng các tên màu định sẵn và hai là dùng giá trị số (mỗi màu có một giá trị số tương ứng).

**Tên màu**

 Lập trình viên không thể sử dụng các tên màu tùy ý, mà chỉ được sử dụng các tên màu đã được đặc tả sẵn trong CSS. CSS3 hỗ trợ 140 tên màu.

Ví dụ, một chỉ dẫn CSS có dùng tên màu:

p { color: gray; } /\* văn bản trong p sẽ có màu xám \*/

Xem thêm các tên màu tại địa chỉ:

<https://www.w3schools.com/colors/colors_names.asp>

**Giá trị số**

Xác định màu bằng tên tuy đơn giản nhưng lại bị giới hạn về số lượng. Vì vậy, mọi người hay sử dụng giá trị số để xác định một màu, đó là giá trị RGB.

Với RGB, chúng ta có thể xác định hàng triệu màu khác nhau.

Ngoài hệ màu RGB, CSS cũng hỗ trợ hệ màu HSL.

**Hệ màu RGB**

Máy tính tạo ra các màu khác nhau bằng cách kết hợp ba màu cơ bản (với tỉ lệ nhất định) là: đỏ (red), xanh lá cây (green) và xanh da trời (blue). Tên gọi RGB là viết tắt của ba chữ Red, Green và Blue.

Mỗi chỉ số trong tổ hợp RGB có giá trị nằm trong khoảng từ 0 tới 255. Nếu ba chỉ số có giá trị là 0, chúng ta sẽ có màu đen (black). Nếu ba chỉ số đều có giá trị là 255, chúng ta sẽ có màu trắng (white).

Ví dụ màu lavender sẽ có bộ ba giá trị là (200, 178, 230).

**Lấy màu**

Cách dễ nhất để lấy được giá trị RGB của một màu là dùng công cụ có tên là Color Picker có sẵn của Google.

**Cách viết giá trị màu RGB**

**Dùng giá trị hệ 10**

Bắt đầu là kí hiệu rgb, sau đó là ba giá trị hệ 10 của ba màu cơ bản (RGB), mỗi giá trị nằm trong khoảng từ 0 tới 255, ví dụ,

color: rgb(200, 178, 230);

**Dùng giá trị %,**

Bắt đầu là kí hiệu rgb, sau đó là ba giá trị % của ba màu cơ bản, giá trị % được tính bằng cách lấy “giá trị hệ 10”/255\*100%, ví dụ,

color: rgb(78%, 70%, 90%);

**Dùng giá trị hệ 16**

Bắt đầu là kí hiệu #, sau đó là ba giá trị hệ 16 của ba màu cơ bản, mỗi giá trị màu được biểu diễn bằng hai kí số hệ 16, ví dụ,

color: #C8B2E6;

color: #FFCC00 hoặc color: #993366

thì có thể viết rút gọn thành,

color: #FC0 hoặc color: #936

**Màu RGBa**

Màu RGBa cho phép xác lập một màu bất kì cùng với độ mờ của màu. Chữ a là viết tắt của alpha, được sử dụng để thiết lập độ mờ của màu, nếu a có giá trị 0: tương ứng với màu trong suốt hay mờ nhiều nhất (transparent), nếu a có giá trị 1: tương ứng với màu ít mờ nhất hay đậm đặc nhất (opaque).

* 1. Màu chữ & màu nền

**Màu chữ**

Khi thiết lập màu chữ, màu đường viền (border) cũng được thiết lập. Để thiết lập màu chữ, sử dụng thuộc tính color.

Color : giá trị màu (tên hoặc số)

Nếu thuộc tính border-color cũng được thiết lập thì nó sẽ thay thế màu của đường viền đang được thiết lập bằng thuộc tính color. Nghĩa là thuộc tính border-color có độ ưu tiên cao hơn thuộc tính color.

**Màu nền**

Sử dụng thuộc tính background-color để thiết lập màu nền cho các phần tử HTML.

background-color: giá trị màu (tên hoặc số) | transparent

– Kế thừa: không

Khung nền (background, hoặc canvas) bao gồm: vùng chứa nội dung, khoảng đệm phía trong đường viền (padding), nền chứa đường viền, không bao gồm vùng đệm phía ngoài đường viền (margin).

Mặc dù màu nền không được kế thừa xuống các hậu duệ, tuy nhiên, do màu nền mặc định của các phần tử là trong suốt (transparent), nên màu nền của phần-tử-cha cũng được xem như là màu nền của các phần tử hậu duệ

**Giới hạn vùng nền**

Như đã biết, vùng nền (background) mặc định của một phần tử sẽ được tính từ viền ngoài của đường biên (border). Tuy nhiên, chúng ta có thể sử dụng thuộc tính background-clip để giới hạn vùng nền theo các tiêu chí khác nhau.

Thuộc tính background-clip:

– Giá trị: border-box | padding-box | content-box

– Mặc định: border-box

– Kế thừa: không

* 1. Làm mờ một phần tử

Opacity có nghĩa là độ mờ.

Opacity: giá trị số (trong khoảng từ 0 đến 1)

– Mặc định: 1

– Kế thừa: không

Giá trị của thuộc tính opacity càng nhỏ thì phần tử càng mờ, giá trị 0 là mờ nhất, giá trị 1 là rõ nhất