# Font

## Font Size

Ta sẽ bàn qua một chút về đơn vị tuyệt đối (Absoulute) và đơn vị tương đối (Relative). Đơn vị tuyệt đối ở đây trong ví dụ sẽ là **pixel (px)** và tương đối sẽ là **em’s (em)** và **percentage (%)**.

Pixel là đơn vị điểm ảnh. Em’s và percentage là các đơn vị phụ thuộc vào tag cha của nó. Với em’s thì một em’s sẽ bằng 1 lần kích thước của tag cha. Nếu n em’s sẽ là n\*kích thước của tag cha. Tương tự percentage là lấy % của tag cha.

Để khai báo style cho kích cỡ chữ, ta dùng cú pháp sau:

selector{

    font-size: #unit;

}

Với # là giá trị (một số nào đó), và unit là đơn vị theo sau. Ví dụ ta apply kích cỡ chữ là 14 pixel cho nội dung bên trong tag <dd>

dd{

    font-size:14*px*;

}

Chúng ta có thể sử dụng đơn vị tương đối, là các đơn vị kế thừa. Chẳng hạn ta có đoạn code dưới đây, <ul> là con của #header. Thì <ul> sẽ có font-size là 2 \* 12, do kế thừa từ #header. Tức là font-size của <ul> sẽ là 24px.

*#header*{

    font-size:12*px*;

}

*#header* ul{

    font-size:2*em*;

}

Đoạn code dưới đây là tương tự đoạn code ví dụ ở trên. Nếu ta lấy 50% thì font-size của <ul> sẽ là 6.

*#header*{

    font-size:12*px*;

}

*#header* ul{

    font-size:200*%*;

}

## Font Weight

Font Weight chính là độ dày của nét chữ, chẳng hạn như Bold là làm chữ dày hơn. Cú pháp style font weight là:

selector{

    font-weight: value;

}

Value có thể là vài giá trị sau đây:

Timeline

Description automatically generated with medium confidence

Tùy từng font – family mà vài giá trị trong “lighter, normal, bold” có thể sẽ không có.

Khi dùng các số làm value, 100 – 300 là khoảng của lighter, 600 – 700 là của normal và 900 sẽ là của bold.

## Font Family

Họ Font chữ trong CSS được declare là:

selector{

    font-family: FontName;

}

Ví dụ:

p{

    font-family:arial;

}

Thường thì value của “font-family” sẽ có nhiều tên font khác nhau cùng kiểu, đề phòng trường hợp người dùng chưa cài font được declare, các font tiếp theo sẽ đáp ứng. Lưu ý là các font phải cùng kiểu dáng kích cỡ để backup cho nhau. Chẳng hạn như:

p{

    font-family:arial, helvetica, sans-serif;

}

# Text

## Text Decoration

Để định kiểu cho một element nào đó, ta dùng cú pháp sau:

selector{

    text-decoration: value;

}

Ví dụ như chúng ta muốn các heading 1 được gạch dưới:

h1{

    text-decoration: underline;

}

Các value sau text-decoration sẽ có nhiệm vụ tương ứng với tên của chúng. Vài giá trị của text-decoration có thể có là:

A picture containing timeline

Description automatically generated

Ngoài ra còn có value “inherit” kế thừa từ element cha của nó.

## Text Transform

Text Transform là thay đổi giữa các dạng: viết hoa, viết thường, viết hoa đầu từ,… Cú pháp để declare là:

selector{

    text-transform: value;

}

Chẳng hạn như muốn header 1 có các từ được viết hoa chữ cái đầu:

h1{

    text-transform: capitalize;

}

## Text Colour

Có hai dạng màu của chữ, đó là màu chữ và màu background. Để định dạng màu chữ ta dùng cú pháp:

selector{

    color: value;

}

Còn định dạng màu nền:

selector{

    background-color: value;

}

Value là tên một màu sắc nào đó, như black, white, red, aquamatic hay green. Value của hai declaration này có thể là mã Hex, chẳng hạn như #0000000 là màu đen hoặc #800000 là màu hạt dẻ.

# Styling Links

Chúng ta đã đi qua một số selector và một số style cơ bản, cùng áp dụng chúng vào việc styling các link nào. Trước tiên ta muốn link mặc định sẽ có màu sắc nhẹ nhàng, không gạch dưới và được in đậm. Ta dùng đoạn code sau:

a{

    color: aqua;

    text-decoration: none;

    font-weight:bold;

}

Kết quả sẽ là:



Sau đó ta muốn link có hiệu ứng hover, khi trỏ vào thì đổi màu chữ, màu nền và được gạch dưới.

a*:hover*{

    color:white;

    background-color:black;

    text-decoration: underline;

}

Kết quả khi hover vào sẽ là:



# Distance

Có hai loại khoảng cách cần lưu tâm là letter spacing và line height, khoảng cách giữa các chữ và giữa các dòng.

## Letter Spacing

Để declare khoảng cách giữa các chữ (ký tự, không phải từ), ta sử dụng cú pháp:

selector{

    letter-spacing: #unit;

}

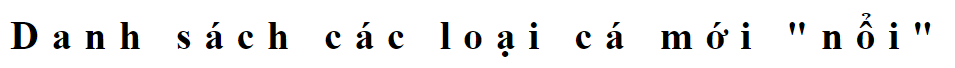
Ví dụ như:

h1{

    letter-spacing: 10*px*;

}

Sẽ có kết quả là:



Ngoài ra chúng ta cũng có thể sử dụng **“word-spacing”** để thiết lập khoảng cách giữa các từ.

## Line Height

Còn về khoảng cách dòng, cú pháp:

selector{

    line-height: #unit;

}

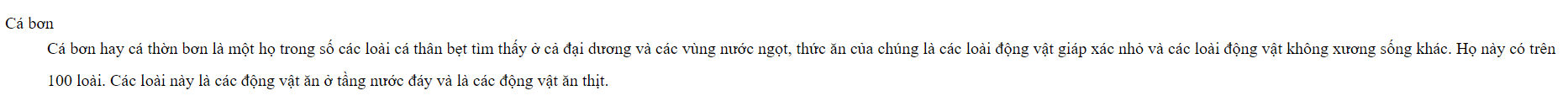
Ví dụ như là:

dd{

    line-height: 30*px*;

}

Kết quả ở trang Web là:



## Paragraph Spacing

Để thiết lập khoảng cách cho các đoạn văn bản, ta có thể dùng một trick thuộc Box Model (nói sau). Đó là set viền cho block của element văn bản này dày thêm một chút để khoảng cách giữa các đoạn văn trở nên xa nhau hơn.

Giả sử ta có hai đoạn văn bản nhỏ như dưới đây:

        <div *class*="contact-method">

            <h4>Qua email: </h4>

            <p>Email công sở</p>

            <p>Email cá nhân</p>

        </div>

Đoạn code thiết lập có thể như sau:

p{

    margin-bottom: 32*px*;

}

Có thể nhận thấy rõ ràng là viền dưới của block này đã dày thêm, tạo cảm giác như hai đoạn văn bản rời xa nhau.

A picture containing chart

Description automatically generated

# Color

Về mã màu, thường sẽ có định dạng là **#abcdef**. Với ab là thuộc khoảng màu đỏ, cd là màu xanh lá và ef là màu xanh dương. Tức chúng là thông số thể hiện cho màu RGB, chẳng hạn **#550000** thì là màu đỏ, vì các giá trị ở cdef đều là 0. Lấy một ví dụ khác chẳng hạn như **#00ff00,** thì là màu xanh lá. Và số từ 1 đến f (hệ thập lục phân) biểu thị độ sáng của màu đó, càng về f thì càng sáng, càng về 0 thì càng đậm. Nếu mọi giá trị là 0 hoặc gần 0 (khoảng 1 đến 2), thì là **màu đen**. Ngược lại, nếu mọi giá trị là f hoặc gần f (d đến e), thì là **màu trắng**.

Ngoài cách này ra, ta có thể sử dụng hàm **rgb(a, b, c)**. Với tham số là bộ số a,b,c có giá trị từ 0 đến 255.

# Opacity

Để tùy chỉnh độ trong suốt (hoặc độ đục) của một element, ta dùng property **opacity**, có value từ 0 đến 1. Với 0 là hoàn toàn trong suốt (hoàn toàn không đục) và 1 là hoàn toàn rõ ràng (mặc định hay hoàn toàn đục).

*.box*{

    border: 1*px* solid #000;

    width: 500*px*;

    height: 80*px*;

    background-color: aquamarine;

    opacity: 0.5;

}

Độ trong suốt của nó được so sánh với box bên dưới.

Shape, rectangle

Description automatically generated

Nếu như có thể để hai box trùng nhau, box nào trong HTML liệt kê sau mà có opacity nhỏ hơn box dưới thì có khả năng nhìn xuyên thấu.

Diagram, venn diagram

Description automatically generated

Property **opacity** không chỉ làm background trở nên trong suốt hoặc đục hơn, mà nó còn ảnh hưởng đến content bên trong nó. Cũng ví dụ trên thì sẽ có kết quả là:

Shape, rectangle

Description automatically generated

Nếu muốn content không bị trở nên trong suốt cùng với background, chúng ta có thể sử dụng hàm **rgba(x,y,z,w)** cho màu nền. Với x,y,z là mã màu RGB còn w là độ trong suốt.

*.box*{

    border: 1*px* solid #000;

    width: 500*px*;

    height: 80*px*;

    background-color: rgba(255, 255, 255, 0.5);

}

Shape, rectangle

Description automatically generated

# Gradients

Các ví dụ của dải màu gradient:

Để tạo ra màu gradient, chúng ta vẫn dùng property opacity, nhưng dùng một hàm đặc biệt là **linear-gradient()**. Linear Gradient là dải màu sẽ chuyển đổi hoặc giao thoa từ trên xuống dưới. Có thể có các loại gradient khác linear. Chúng ta sẽ sử dụng một ví dụ để minh họa cách dùng.

*.box*{

    border: 1*px* solid #000;

    width: 500*px*;

    height: 80*px*;

    background: linear-gradient(45*deg*, violet, aquamarine);

}

Tham số đầu tiên là direction của gradient, có thể là **to left** (từ phải qua trái), **to right** (trái qua phải), **to bottom** (trên xuống dưới,… Thậm chí direction có thể là **45deg** (nghiêng 45 độ). Hai tham số sau là màu bắt đầu và màu kết thúc theo direction. Đoạn code trên bắt đầu từ màu xanh nước biển bên phải và chuyển thành màu hồng tím bên trái.

Shape, rectangle

Description automatically generated

Độ nghiêng phụ thuộc vào trục của gradient như sau:

Diagram

Description automatically generated

Chúng ta có thể làm một trick cho button trên trang Web, bằng cách thay đổi direction màu gradient khi hover vào button. Ta có code sau:

*.box:hover*{

    border: 1*px* solid #000;

    width: 500*px*;

    height: 80*px*;

    background: linear-gradient(45*deg*,aquamarine, violet);

}

Trước khi và sau khi hover:

Shape, rectangle

Description automatically generated

Shape, rectangle

Description automatically generated

# Box Shadow

Để tạo bóng cho một element, ta dùng property **box-shadow**.

*.box*{

    border: 1*px* solid #000;

    width: 500*px*;

    height: 80*px*;

    background: linear-gradient(45*deg*,violet, aquamarine);

    box-shadow: 3*px (horizontal)* 4*px (vertical)* 5*px (spread)* rgba(0, 0, 0, 0.7);

}

Hai value đầu là vị trí chiều ngang và chiều dọc, thể hiện độ xa của bóng so với box của element. Value tiếp theo là độ mờ của bóng, giá trị càng lớn thì càng mờ. Value kế cuối là độ lan (spread), cho biết bóng của box sẽ lan rộng bao xa. Cuối cùng là màu sắc của bóng, thường thì chúng ta dùng hàm **rgba** vì đa số các bóng đều có độ trong suốt nhất định.

Shape, rectangle

Description automatically generated

Có thể dùng trang web bên dưới để preview bóng của box.

Graphical user interface, application, website

Description automatically generated