# The Box Model

Box Model là cách mà một element thể hiện chính nó trên trang Web theo không gian. Một Box Model điển hình sẽ có dạng như thế này.

A picture containing diagram

Description automatically generated

## Content

Phần đầu tiên là phần nội dung, nó sẽ nằm bên trong một khung nhỏ, đánh dấu là khung xanh da trời. Kích thước phần content này sẽ được khai báo thông qua “width” và “height”. Chẳng hạn như:

*.box*{

    width: 80*px*;

    height: 20*px*;

}

Shape

Description automatically generated

## Padding

Tiếp theo là **padding** (đệm), đây là phần thuộc element.

Cú pháp thiết lập phần đệm chẳng hạn này là:

*.box*{

    padding: 30*px*;

}

Lúc này thì mọi cạnh của **padding** đều có giá trị là 30. Cũng có thể chỉ thiết lập một số cạnh nhất định như top, right, bottom hoặc left.

*.box*{

    padding-top: 30*px*;

}

A picture containing shape

Description automatically generated

Đồng thời, chúng ta cũng có thể liệt kê bốn cạnh trong cùng một hàng khai báo:

*.box*{

    padding: 30*px* 20*px* 30*px* 20*px*;

}

Theo thứ tự lần lượt sẽ là: top, right, bottom và left.

A picture containing diagram

Description automatically generated

Hoặc chỉ đơn giản là:

*.box*{

    padding: 30*px* 20*px* ;

}

Value đầu dành cho top và bottom, value sau là right và left.

Ngoài ra, nếu chỉ có ba value, value còn lại sẽ tự động match với value đi cặp với nó. Chẳng hạn như:

*.box*{

    padding: 30*px* 20*px* 10*px* ;

}

Thì cạnh left sẽ có giá trị là 20, vì nó cặp với right.

A picture containing diagram

Description automatically generated

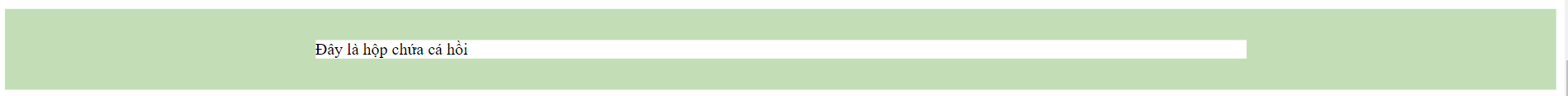
Xét trường hợp chúng ta muốn đặt content ở giữa một block của element (không bao gồm margins). Chúng ta sẽ thiết lập giá trị của right và left của **padding** có đơn vị là **percentage** và giá trị bằng nhau**.**

*.box*{

    padding: 30*px* 20*%*;

}

Phần màu xanh lá cây ở hình dưới chính là phần **padding**, left và right có kích thước là 20% của chiều rộng của thẻ chứa nó.



## Border

Border chính là viền của một Box nào đó, nó thuộc element. Khác với **padding** ở chỗ, border còn cần thêm một vài thông tin trong phần khai báo của chúng thì nó mới xác định rõ. Ví dụ như ta muốn viền của Box là đường liền nét, độ dày 1px và có màu đen.

*.box*{

    border: 1*px (width)* solid (style) #000 (color);

}

Chúng ta cũng có thể khai báo cho từng cạnh như **padding**, và cũng có thể khai báo cho từng thuộc tính nhỏ chẳng hạn như độ dày, kiểu viền và màu sắc (theo thứ tự ở trên). Một đoạn code cho ra kết quả như ví dụ trên là:

*.box*{

    border-width: 1*px*;

    border-style: solid;

    border-color: #000;

}

Đối với từng cạnh, ví dụ như cạnh top, CSS sẽ là:

*.box*{

    border-top-width: 1*px*;

    border-top-style: solid;

    border-top-color: #000;

}

Loại viền có thể có nhiều loại, đơn cử như một số loại sau đây:

Text

Description automatically generated with medium confidence

## Margins

Phần **margin** chính là phần để phân biệt các block của các element với nhau, chúng không thuộc một element như **padding** và **border**. Sẽ có bốn cạnh là top, right, bottom và left. Tương tự như **padding** thì **margin** cũng có các cách khai báo để thiết lập cách cạnh đồng thời hoặc riêng lẻ.

### Vertical Margins Collapse

Xét một trường hợp như sau:

//HTML

 <div *class*="box">Đây là hộp chứa cá hồi</div>

 <div *class*="box">Đây là hộp chứa cá thu</div>

//CSS

*.box*{

    margin: 30*px* 20*px* ;

}

Ta biết box 1 có **margin** bottom là 30px, và box 2 có margin top là 30px. Theo lẽ thường thì cả hai box này cách nhau 60px. Nhưng không phải, giá trị của nó là 30px. Đây gọi là hiện tượng Vertical Margin Collapse (chồng lề dọc). Xảy ra khi có hai box giống nhau đứng liền kề, hai viền của hai box ấy sẽ chồng vào nhau, dẫn đến khoảng cách giữa hai box lúc này chỉ có 30px mà thôi. Nếu một trong hai có margin lớn hơn, khoảng cách giữa chúng vẫn sẽ là kích thước của margin lớn nhất.

### Auto Value

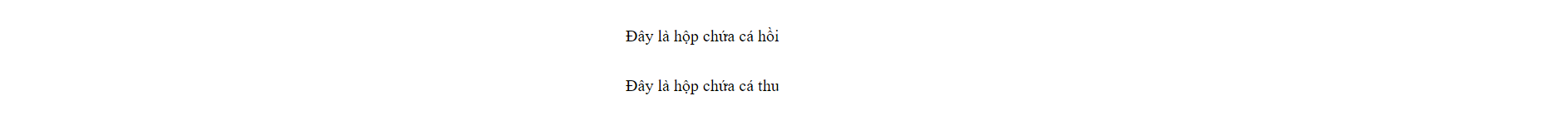
Nếu ta muốn đặt một box vào giữa trang Web, ta sẽ dùng giá trị của cặp right – left là **“auto”.** Để làm điều này thì bắt buộc chúng ta phải thiết lập kích thước chiều rộng cho content.

*.box*{

    width: 300*px*;

    margin: 30*px* auto;

}



Cũng có thể sử dụng đơn vị là **percentage** để centering box. Khi ta set box có width là 50%, phần lề trái và phải còn lại mỗi cái sẽ là 25%.

*.box*{

    width: 50*%*;

    margin: 30*px* 25*%*;

}

# Block Element

<div> là một **block element**, nghĩa là khi apply style cho nó, kích thước của nó sẽ chiếm trọn một dòng/hàng trên trang Web. Ngoài ra còn có một số tag khác thuộc **inline element**, các element không chiếm trọn một dòng mà chỉ sử dụng phần diện tích vừa đủ để bao quanh content bên trong đó mà thôi. Ảnh dưới đây sẽ minh họa cho điều này:

A picture containing table

Description automatically generated

Một số **block element** điển hình là <p>, <h1-6>, <ol>, <ul>,<pre>, <div>, <form>, <hr>, <table>.

## Display: block

Nếu muốn một **inline element** trở thành dạng **block element**, ta chỉ cần thêm property **display** và có giá trị là **“block”.**

a[*href*]

{

    margin: 10*px*;

    border: 1*px* solid #000;

    padding: 30*px* 20*px*;

    display: block;

}

Background pattern

Description automatically generated with low confidence

Lúc chưa sử dụng thì:

A picture containing text

Description automatically generated

## Display: inline-block

Thông thường thì các **inline element** sẽ không tuân theo quy luật margin như **block element**. Chúng chỉ tính margin left và right, ở ví dụ trên thì lúc là khoảng cách chiều ngang giữa hai box là 20px, không có hiện tượng chồng margin.

Đồng thời margin top và bottom cũng sẽ được bỏ qua. Điều này dẫn đến một hiện tượng khác:

Diagram

Description automatically generated with low confidence

Do là **inline**, nên chúng sẽ tiếp tục theo sau các dòng của <p>, mà không quan tâm đến Box có margin top và bottom là bao nhiêu. Để khắc phục, chúng ta sẽ biến chúng thành các **inline element** nhưng tuân theo quy luật của **block element**. Bằng cách đổi value của property **display** thành **“inline-block”.**

a[*href*]

{

    margin: 50*px*;

    border: 1*px* solid #000;

    padding: 30*px* 20*px*;

    display: inline-block;

}

Diagram

Description automatically generated with low confidence

Value **“inline-block”** cũng có thể dùng cho block **element**.

# Rounded Corners

## Border-radius

Để bo tròn góc của một Box, ta sử dụng property **border-radius** và có value là một số nào đó kèm theo đơn vị. Giá trị của số càng lớn thì góc của Box càng bo tròn.

selector{

    border-radius: #unit;

}

Chẳng hạn như:

*.box*{

    margin: 30*px*;

    padding: 20*px*;

    border: 1*px* solid #000;

    border-radius: 10*px*;

    width: 80*px*; height: 50*px*;

}

Kết quả cho đoạn code hộp cá hồi là:

Text, letter

Description automatically generated

Cũng có thể liệt kê bốn loại corner trong một lần khai báo, theo thứ tự đại diện cho top-left, top-right, bottom-left, bottom-right. Mỗi cái có thể có giá trị khác nhau. Chẳng hạn như ta dùng đoạn code dưới đây:

*.box*{

    margin: 30*px*;

    padding: 20*px*;

    border: 1*px* solid #000;

    border-radius: 40*px* 20*px* 10*px* 5*px*;

    width: 80*px*; height: 50*px*;

}

Kết quả ở dưới cho thấy cạnh top-left bị cong nhiều nhất, sau đó đến top-right và dần thẳng khi đến bottom-left.

Text

Description automatically generated

Value có thể khai báo kiểu đối xứng theo cặp, top-left và bottom-right, top-right và bottom-left.

Text

Description automatically generated

## Making Circle

Chúng ta cũng có thể tạo ra một hình tròn thông qua cách này. Bằng cách set **width** và **height** như nhau, sau đó cho radius bằng nửa giá trị của **width** và **height**. Lưu ý là ta cần xóa đi các khai báo cho margin và padding.

*.box*{

    background-color: #ccc;

    border-radius: 40*px*;

    width: 80*px*;

height: 80*px*;

}

Kết quả có được sẽ là:

Text

Description automatically generated

# Background

## Color

Một property thường dùng cho background là **background-color**.

*.box*{

    width: 100;

    height: 50;

    background-color: yellow;

}

Đây là kết quả:

A picture containing text

Description automatically generated

## Image

Property thứ hai sẽ là **background-image**, bên trong url() là đường dẫn đến hình ảnh đó.

*.box*{

    width: 1500*px*;

    height: 1000*px*;

    background-image: url(../img/sample.jpg);

}

## Repeat

Ảnh nền sẽ chiếm diện tích từ border trở vào trong (không tính phần margin). Nếu như kích thước ảnh bé so với toàn bộ kích thước của element, các hình ảnh sẽ được repeat sao cho vừa đủ kích thước element. Để cho hình ảnh không bị repeat, ta có thể dùng property **background-repeat** và set value là **“no-repeat”**.

Graphical user interface

Description automatically generated with low confidence

*.box*{

    width: 1500*px*;

    height: 1000*px*;

    background-image: url(../img/sample.jpg);

    background-repeat: no-repeat;

}

Ngoài ra, như ta thấy ở trên thì hình ảnh khi bị repeat sẽ mất hình, để khắc phục điều này, ta dùng value của **background-repeat** là **“round”**.

*.box*{

    width: 1000*px*;

    height: 500*px*;

    background-image: url(../img/sample.jpg);

    background-repeat: round;

}

Hình ảnh được giãn ra cho vừa đủ kích thước của element.

Graphical user interface, text, chat or text message

Description automatically generated

## Position

Để thiết lập vị trí cho background trong một element, ta có thể sử dụng property **background-position**. Value của property này có thể là **top, left, right, bottom, center, inherit**.

*.box*{

    border: 1*px* solid #000;

    width: 1000*px*;

    height: 500*px*;

    background-image: url(../img/sample.jpg);

    background-repeat: no-repeat;

    background-position: center;

}

Graphical user interface, text, application

Description automatically generated

Cũng có thể kết hợp hai giá trị như **top left, top right, bottom left và bottom right**. Và dùng đơn vị số (pixel, %) với value đầu là khoảng cách từ viền left và value sau là viền top.

*.box*{

    border: 1*px* solid #000;

    width: 1000*px*;

    height: 500*px*;

    background-image: url(../img/sample.jpg);

    background-repeat: no-repeat;

    background-position: 10*%* 20*%*;

}

Graphical user interface, text, application, chat or text message

Description automatically generated

## Size

Tùy chỉnh kích thước của hình ảnh bằng cách dùng property **background-size**. Value đầu là width và value sau là height.

*.box*{

    border: 1*px* solid #000;

    width: 1000*px*;

    height: 500*px*;

    background-image: url(../img/zero-two.jpg);

    background-repeat: no-repeat;

    background-position: center;

    background-size: 1000*px* 500*px*;

}

## Shorthand form

Có thể gom các property này gọn lại ở bên trong property **background**. Theo thứ tự là **url**, **reapeat**, **position** và **size**. Nhưng chúng ta tách biệt **size** ra để khỏi bị rối. Đồng thời luôn declare **color** sau **background** để style này được apply mà ko bị background đè lên. Minh họa cho dạng rút gọn là:

*.box*{

    border: 1*px* solid #000;

    width: 1000*px*;

    height: 500*px*;

    background: url(../img/zero-two.jpg) no-repeat center;

    background-color: aquamarine;

    background-size: 1000*px* 1000*px*;

}

## Multiple Background

Các background có thể chồng lên nhau, bằng cách dùng property **background-image** và value có nhiều giá trị. Mỗi giá trị cách nhau bởi dấu phẩy. Chẳng hạn như sau:

*.box*{

    border: 1*px* solid #000;

    width: 1000*px*;

    height: 500*px*;

    background-image: url(../img/web.jpg), url(../img/coding.jpg);

}

Các hình ảnh sẽ hiển thị theo thức tự được declare, cái nào declare trước sẽ nằm ở trên, declare cuối sẽ nằm dưới cùng.

A picture containing graphical user interface

Description automatically generated

Đặc biệt, những property khác khi sử dụng sẽ phải thêm dấu phẩy (“,”) để ngăn cách value cho các hình ảnh riêng biệt. Nếu muốn các ảnh cùng apply một value thì không cần.

*.box*{

    border: 1*px* solid #000;

    width: 1000*px*;

    height: 500*px*;

    background-image: url(../img/web.jpg), url(../img/coding.jpg);

    background-repeat: no-repeat;

    background-position: center;

    background-size: 500*px*, 100*%* (bằng kích thước của element);

}

Kết quả sẽ là:

Graphical user interface, application

Description automatically generated