## **Background**

**background-color**

.left { background-color: #ffdb3a; }

**background-image**

.left { background-image: url('ire.png'); }

/\* Or use multi images \*/

background-image: url('khaled.png'), url('ire.png');

**background-repeat**

Giá trị của của thuộc tính có thể là: repeat-x, repeat-y, repeat, space, round, no-repeat.



**background-size**

Thuộc tính background-size định nghĩa kích thước của hình nền. Giá trị của nó có thể là kích thước chiều dài và rộng hoặc là tỉ lệ phần trăm. Từ khóa có sẵn cho thuộc tính là contain và cover. Giá trị contain sẽ co dãn hình ảnh để phù hợp với khung. giá trị cover, ở một mặt khác nó sẽ kéo dãn hình ảnh sao cho vừa với khung mà ko gây sai lệch tỉ lệ.



Đối với các giá trị chiều dài và giá trị phần trăm, chúng ta có thể xác định cả chiều rộng và chiều cao của ảnh nền. Tỷ lệ phần trăm giá trị được tính toán liên quan đến kích thước của phần tử.

**background-attachment**

Thuộc tính background-attachment dùng để kiểm soát hình nền liên quan đến các khung hình và các thành phần. Ba giá trị là: fixed, local, scroll.

Fixed nghĩa là hình ảnh nền được cố định vào khung nhìn và không di chuyển, ngay cả khi người dùng đang di chuyển dọc theo khung.

Local là hình nền nên được cố định vào vị trí của nó trong phần tử. Nếu phần tử có một cơ chế di chuyển và hình nền được đặt lên hàng đầu, khi người dùng cuộn xuống phần tử, hình nền sẽ di chuyển ra khỏi tầm nhìn.

Scroll có nghĩa là các hình nền cố định và sẽ không di chuyển ngay cả với các nội dung của các phần tử của nó.

**background-position**

Từ khóa có sẵn: top, right, bottom, left và center. Chúng ta có thể sử dụng các từ khóa trong bất kỳ sự kết hợp, và nếu chỉ có một từ khóa được quy định.

Đối với chiều dài và tỷ lệ phần trăm giá trị, chúng ta cũng có thể xác định vị trí dọc theo trục x và trục y. Tỷ lệ phần trăm giá trị là liên quan đến các yếu tố có chứa.

.left { background-position: 20px 70px; /\* Others same as .top-left \*/ }

.right { background-position: 50%; /\* Others same as .top-left \*/ }

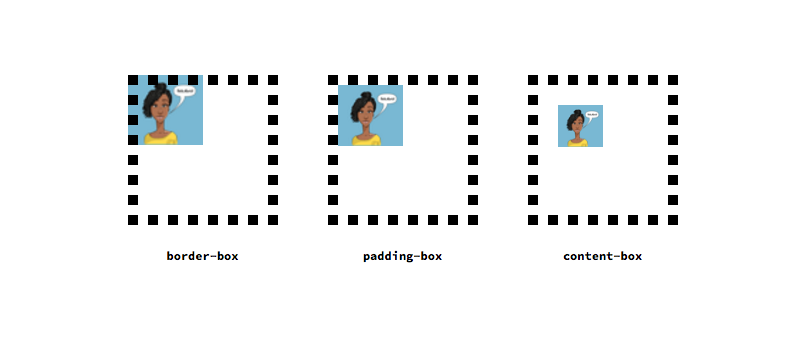
**background-origin**

Thuộc tính background-origin quy định cụ thể trong đó diện tích các mô hình hộp hình nền phải được bố trí theo.

Giá trị mặc định là border-box, khi mà vị trí hình ảnh ở sát viền của khung,

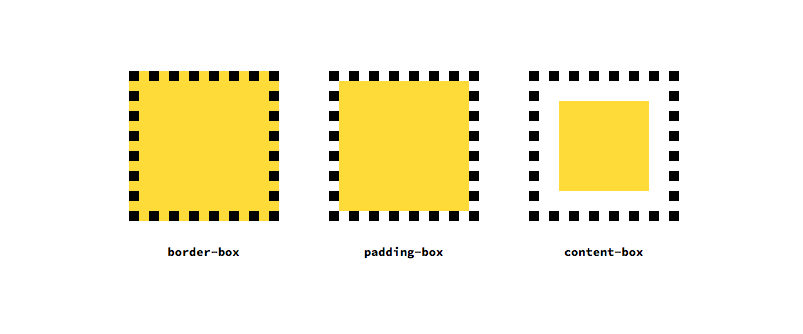
padding-box khi mà hình ảnh ở trong viền của khung và

content-box khi mà hình ảnh ở trong khung



**background-clip**

Thuộc tính background-clip xác định khu vực sơn nền, đó là khu vực mà nền có thể sơn lên, giống như background-origin, nó được dựa trên các lĩnh vực mô hình hộp.



## **Thuộc tính position**

Thuộc tính position hiện tại có những giá trị thường được dùng sau:

* **relative**: Giá trị này thường được sử dụng để thiết lập vị trí của phần tử mà không gây ảnh hưởng tới việc hiển thị ban đầu cũng như các phần tử khác
* **absolute**: Giá trị này dùng để thiết lập vị trí của một phần tử theo phần tử cha có giá trị thuộc tính position là relative hoặc absolute
* **fixed**: Giá trị này giúp cho phần tử luôn cố định một chỗ khi chúng ta scroll trình duyệt
* **static**: Đây là giá trị hiển thị mặc định của thuộc tính position trong CSS.

Và đi kèm với thuộc tính position thì đó là các thuộc tính dùng để căn chỉnh vị trí cho phần tử

* **top**
* **bottom**
* **right**
* **left**

## **Media CSS**

@media not|only mediatype and (media feature) {

CSS-Code;

}

**Trong đó mediatype gồm các thuộc tính hay sử dụng sau:**

* **all:** Dùng cho mọi thiết bị
* **print:** Dùng cho máy in
* **screen:** Dùng cho máy tính và các thiết bị smart phone

Thực tế vẫn còn nhiều nữa nhưng với lập trình web thì chúng ta thường sử dụng ba thuộc tính đó thôi. Và trước khi đi vào tìm hiểu các thuộc tính thì ban phải phân biệt hai khái niệm sau:

* **Device**: Là thiết bị sử dụng website như Laptop, Desktop, Iphone, ..
* **Viewport**: Là kích thước hiển thị của giao diện.

<meta name="viewport" content="width=device-width, initial-scale=1">

**Và media featured thì gồm các thuộc tính sau:**

* **aspect-ratio**: Tỉ lệ giữa chiều rộng và chiều cao của viewport
* **min-aspect-ratio**: Tỉ lệ tối thiểu giữa chiều rộng và chiều cao của viewport
* **max-aspect-ratio**: Tỉ lệ tôi đa giữa chiều rộng và chiều cao của viewport
* **color**: Số bits cho mỗi màu sắc của device
* **color-index:** Số lượng màu sắc mà device có thể hiển thị
* **device-aspect-ratio**: Tỉ lệ giữa chiều rộng và chiều cao của device
* **max-device-aspect-ratio**: Tỉ lệ tối đa giữa chiều rộng và chiều cao của device
* **min-device-aspect-ratio**: Tỉ lệ tối thiểu giữa chiều rộng và chiều cao của device
* **device-height**: Chiều cao của device
* **device-width**: Chiều rộng của device
* **height**: Chiều cao của viewport
* **width**: Chiều rộng của viewport
* **max-width**: Chiều rộng tối đa của viewport
* **min-width**: Chiều rộng tối thiểu của viewport
* **max-height**: Chiều cao tối đa của viewport
* **min-height**: Chiều cao tối thiểu của viewport
* **min-device-width**: Chiều rộng tối thiểu của device
* **max-device-width**: Chiều rộng tối đa của device
* **min-device-height**: Chiều cao tối thiểu của device
* **max-device-height**: Chiều cao tối đa của device
* **orientation**: Định hướng của khung nhìn (xoay hoặc không xoay thiết bị)
* **resolution**: Độ phân giải của thiết bị đầu ra (sử dụng dpi hoặc dpcm)

## **Đơn vị em và rem**

**Đơn vị tương đối** : Là các đơn vị đo lường được sử dụng trong CSS ở mức tương đối, nó sẽ được thay đổi bởi các thành phần khác. Các đơn vị tương đối bao gồm:

* %
* em
* rem

**1rem bằng bao nhiêu px? (Recommend)**

Kích thước của các phần tử sử dụng đơn vị rem sẽ được quy đổi sang pixels phụ thuộc vào kích thước font chữ của phẩn tử gốc của trang (phần tử html) Ví dụ: html có font-size: 10px; vậy thì một phần tử trong đó có width: 10rem; đổi ra sẽ có width: 100px

**1em bằng bao nhiêu px?**

Đối với **em**, việc quy đổi sang px không dựa vào thuộc tính font-size của phần tử gốc html mà sẽ dựa vào font-size của phần tử chính phần tử hiện tại.