

# Hiệu ứng chuyển đổi transition trong CSS3

Sử dụng các loại chuyển đổi thuộc tính để biến đổi hiển thị phần tử như dịch chuyển, xoay, thu phóng phần tử

## Hiệu ứng chuyển động, chuyển đổi

CSS3 tạo ra hiệu ứng chuyển đổi bằng cách thay đổi thuộc tính từ giá trị này sang giá trị khác trong một khoảng thời gian. Với các tham số:

- **transition-property** chỉ ra thuộc tính cần chuyển đổi
- **transition-duration** khoảng thời gian chuyển đổi diễn ra
- **transition-timing-function** bước thay đổi
- **transition-delay** khoảng thời gian dừng cho mỗi hiệu ứng chuyển đổi

Ví dụ sử dụng thuộc tính **transition**, hiệu ứng xảy ra trong 5s, với transition-timing-function là **ease-in** cho biết bắt đầu chậm rồi nhanh dần

```
transition: transform 5s ease-in;
```

Hiệu ứng chuyển đổi có thể sử dụng cho nhiều thuộc tính CSS: background-color, width, height, opacity ...

Ví dụ sau tạo ra hiệu ứng thay đổi chiều rộng box khi chuột di vào, ra phần tử. Hiệu ứng diễn ra trong 3s

```
<style>
  div.examp1 {
    width: 50px;
    height: 50px;
    background: #32CD32;
    transition: width 3s;
  }
  div.examp1:hover {
    width: 250px;
  }
</style>
<div class="examp1"></div>
```



## transition-timing-function

Thuộc tính này chỉ ra tốc độ thay đổi khi chuyển đổi

- **ease** bắt đầu chậm, rồi tăng dần (mặc định)
- **ease-in** chậm, tăng dần, dừng đột ngột
- **ease-out** chậm dần đều
- **ease-in-out** giống **ease** nhưng mịn hơn
- **linear** thay đổi đều đều

Ngoài ra có hàm **cubic-bezier()** để định nghĩa theo hàm đồ thị hàm cubic-bezier, ví dụ

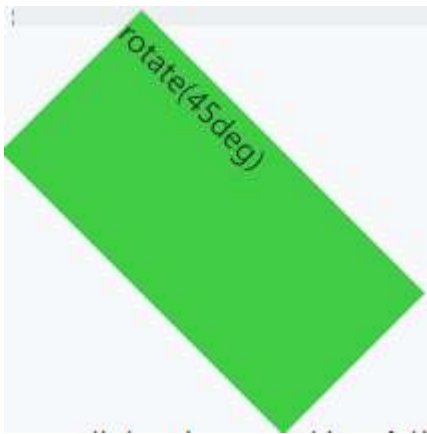
```
transition-timing-function: cubic-bezier(0,0,1,1);
<style>
  div.examp2 {
    width: 50px;
    height: 50px;
    background: #32CD32;
    transition: 3s;
    transition-timing-function: cubic-bezier(0,1.12,.92,-0.21);
  }
  div.examp2:hover {
    width: 100%;
  }
</style>
<div class="examp2"></div>
```



## Xoay phần tử với rotate

Ví dụ sau xoay phần tử đi 10 30 độ

```
<style>
  div.quayphantu {
    width: 200px;
    height: 100px;
    margin-top: 30px;
    background-color: #32CD32;
    transform: rotate(45deg);
  }
</style>
<div class="quayphantu">
  rotate(45deg)
</div>
```



Mặc định gốc quay phần tử là ở giữa phần tử, gốc này có thể thiết lập bằng thuộc tính **transform-origin**

Mặc định thì là **transform-origin: 50% 50%** chính là giữa phần tử

Ví dụ thay gốc quay là ở góc thứ nhất: **transform-origin: 0% 0%** hoặc **transform-origin: top left;**

## Dịch chuyển phần tử translate()

Phương thức **translate(x, y)** để thay đổi vị trí phần tử thêm một đoạn x, y

```
<style>
  div.examp3 {
    padding: 50px;
    position: absolute;
    background-color: #32CD32;
    transform: translate(100px, -10px);
  }
</style>

<div class="examp3"></div>
```

## Nghiêng phần tử skew()

Nghiêng phần tử sử dụng phương thức **skew(x-angle, y-angle)**

```
<style>
  div.examp4 {
    padding: 50px;
    width: 200px;
    background-color: #32CD32;
    transform: skew(-45deg, 0);
    margin-left: 100px;
  }
</style>

<div class="examp4">Nghiêng theo X 45 độ</div>
```



## Thu phóng phần tử scale()

Zoom phần tử sử dụng phương thức `scale(x-scale,y-scale)`

```
<style>
  div.zoomout {
    width: 200px;
    height: 100px;
    background-color: #8BC34A;
    transform: scale(0.7, 0.7);
    color:white;
  }
  div.zoomin {
    margin: 60px;
    width: 200px;
    height: 100px;
    background-color: #8bc34a;
    transform: scale(1.1, 1.7);
    color:white;
  }
</style>

<div class="zoomout">Zoom: x 0.7, y.0.7</div>
<div class="zoomin">Zoom: x 1.1 y 1.7</div>
```

## Kết hợp nhiều hiệu ứng

Ví dụ vừa xoay, vừa dịch chuyển

```
<style>
  div.multi {
    width: 200px;
    height: 100px;
    background-color: #8BC34A;
    transform: rotate(45deg) translate(100px);
    color:white;
    padding: 30px;
  }
</style>

<div class="multi">rotate(45deg) translate(100px);</div>
```