





- Nắm được khái niệm về animation và transform
- Giải thích các thành phần của animation
- Giải thích các thành phần của transform







- Tổng quan về animation và transform
- Giới thiệu các thành phần của animation
- Giới thiệu các thành phần của transform
- Giới thiệu tool online tạo animation





PHÀN 1 Tổng quan về animation và transform





Tại sao lại là css3 animation

- ☐ Trước khi css3 ra đời việc tạo hiệu ứng cho website chủ yếu dựa vào jquery. mặc dù jquery đã đáp ứng rất tốt công việc này nhưng việc dùng jquery lại ít nhiều làm ảnh hưởng tới tốc độ load của website.
- ☐ Tuy nhiên css3 ra đời đã hỗ trợ 2 thuộc tính rất quan trọng là transform và animation, hai thuộc tính này có thể thay thế hoàn toàn jquery trọng việc tạo ra hiệu ứng đẹp mắt cho website và tăng tốc độ lướt web hơn so với việc sử dụng jquery.
- Dể sử dụng CSS animation, chúng ta phải khai báo @keyframe cho từng animation. Keyframe sẽ tạo ra các styles của thành phần tại một thời điểm nhất định.

Ứng dụng của css3 animation

CSS3 animation hỗ trợ việc tạo ra các ảnh động, thay thế các ảnh hoạt hình, ảnh động, flash và javascrip có trong các trang web.

CÚ PHÁP ANIMATION



Cú pháp thuộc tính animation:

animation: name duration timing-function delay iteration-count direction play-state; Trong đó:

- name: là tên của annimation
- duration: là thời gian thực hiện của animation
- timing-function: mô tả hành động, chu kỳ diễn ra của animation
- delay: quy định độ trễ
- iteration-count: số lần thực hiện
- direction: chỉ định việc lặp lại animation sẽ được diễn ra như thế nào(hành động như chu kỳ ban đầu hay sẽ đảo ngược lại hành động). thuộc tính này chỉ có tác dụng khi giá trị iteration-count lớn hơn 1
- * play-state: sẽ chỉ định việc animation có được thực hiện hay không



TRÌNH DUYỆT HỖ TRỢ

Dưới đây là bảng các trình duyệt hỗ trợ animation

thuộc tính	6	е	6		0
@keyframes	43.0	10.0	16.0	9.0	30.0
animation-name	43.0	10.0	16.0	9.0	30.0
animation-duration	43.0	10.0	16.0	9.0	30.0
animation-delay	43.0	10.0	16.0	9.0	30.0
animation-iteration-count	43.0	10.0	16.0	9.0	30.0
animation-direction	43.0	10.0	16.0	9.0	30.0
animation-timing-function	43.0	10.0	16.0	9.0	30.0
animation-fill-mode	43.0	10.0	16.0	9.0	30.0
animation	43.0	10.0	16.0	9.0	30.0





@KEYFRAME TRONG ANIMATION

- Là thuộc tính này có tác dụng thiết lập một chuyển động.
- @keyframe sẽ mô tả chi tiết diễn biến của một animation ra như thế nào trong chu kỳ của nó.
- Cú pháp:

```
@keyframes name{
}
```

- Trong đó:
- ☐ Name là tên của chuyển động.
- code là các đoạn code cho tiến trình chuyển động.
 - ❖ Đoạn code này có thể là các phần trăm từ 0% đến 100%
 - Hoặc cũng có thể là 2 thuộc tính from (tương ứng với 0 %), to (tương ứng với 100 %)



@KEYFRAME TRONG ANIMATION

☐ Ví dụ:





THUỘC TÍNH ANIMATION-NAME

- ☐ Thuộc tính này có tác dụng xác định thành phần sẽ thực thi animation nào.
- Cú pháp: animation-name: name;
- ☐ Trong đó: **name** là tên của thuộc tính **keyframe** ở phía trên.

```
<style>
@keyframes move{
    from{ top: 0px;}
    to{ top: 100px;}
}

p{
    animation-name: move;
    animation-duration: 5s;
    animation-iteration-count: infinite;
    position: absolute;
    width: 100px;
    height: 100px;
    padding: 10px;
    background-color: orange;
}
</style>
Hoc CSS3 Animation
```





THUỘC TÍNH ANIMATION-DURATION

- ☐ Thuộc tính này có tác dụng thiết lập khoảng thời gian thực thị 1 chuyển động animation.
- Cú pháp: animation-duration: time;
- ☐ Trong đó: time là có thể tính bằng s hoặc ms.





THUỘC TÍNH ANIMATION-TIMING-FUNCTION

- -Thuộc tính này có tác dụng xác định tốc độ chuyển động của một animation sẽ như thế nào.
- Cú Pháp: animation-timing-function: value;
- ☐ Trong đó: value là một trong các giá trị sau:
 - linear chuyển động cùng 1 tốc độ từ đầu đến cuối.
 - ease chuyển động lúc đầu chậm sau đó nhanh dần và lại chậm dần về cuối (đây là giá trị mặc định)
 - ❖ ease-in chuyển động chậm ở đầu
 - ❖ ease-out chuyển động chậm ở nửa cuối
 - ❖ ease-in-out chuyển động chậm cả ở đầu và về cuối
 - cubic-bezier(n,n,n,n)



THUỘC TÍNH ANIMATION-TIMING-FUNCTION

Ví dụ với ease: chuyển động lúc đầu chậm sau đó nhanh dần và lại chậm dần về cuối

```
<style>
@keyframes move{
    from{ top: 0px;}
    to{ top: 200px;}
}

p{
    animation-name: move;
    animation-duration: 3s;
    animation-direction: normal;
    animation-iteration-count: infinite;
    animation-timing-function: ease;
    position: relative;
    width: 100px;
    height: 100px;
    height: 100px;
    padding: 10px;
    background-color: orange;
}

/style>
Hoc CSS3 Animation
```





THUỘC TÍNH ANIMATION-DELAY

- ☐ Thuộc tính này có tác dụng xác định độ trễ của mỗi lượt chuyển động.
- Cú pháp: animation-delay: value;
- ☐ Trong đó: **value** là khoảng thời gian trễ, được tính bằng **s** hoặc **ms**.

```
«style>
  @keyframes move{
    from{ top: 0px;}
    to{ top: 200px;}
}

p{
    animation-name: move;
    animation-duration: 3s;
    /*Thiết lập thời gian trễ cho animation 2s*/
    animation-delay: 2s;
    animation-direction: normal;
    animation-iteration-count: infinite;
    position: relative;
    width: 100px;
    height: 100px;
    height: 10px;
    background-color: orange;
}
```





THUỘC TÍNH ANIMATION-ITERATION-COUNT

- ☐ Thuộc tính này có tác dụng, thiết lập số lần thực hiện một animation.
- Cú Pháp: animation-iteration-count: value;
- ☐ Trong đó, **value** có thể là 1 trong các giá trị sau:
 - ❖ là các con số lơn hơn 0 (mặc định value = 1)
 - infinite chạy không giới hạn số lần





THUỘC TÍNH ANIAMTION-DIRECTION

- ☐ Thuộc tính này có tác dụng xác định xem chiều chạy của animation sẽ như thế nào.
- Cú Pháp: aniamtion-direction: value;
- ☐ Trong đó, **value** có thể là 1 trong các giá trị sau:
 - normal animation chạy tự bình thường (đây là giá trị mặc định)
 - reverse animation sẽ chạy ngược lại so với bình thường
 - alternate animation sẽ chạy bình thường ở các lượt lẻ (1,3,5,7,...) và chạy ngược lại ở các lượt chẵn (2,4,6,8,...)
 - alternate-reverse animation se chay ngược lại ở các lượt lẻ (1,3,5,7,...) và chạy bình thường ở các lượt chẵn (2,4,6,8,...)



THUỘC TÍNH ANIAMTION-DIRECTION

Ví dụ: sử dụng aniamtion-direction: reverse

```
<style>
  @keyframes move{
    from{ top: 0px;}
    to{ top: 100px;}
}

p{
    animation-name: move;
    animation-duration: 3s;
    animation-direction: reverse;
    animation-iteration-count: infinite;
    position: relative;
    width: 100px;
    height: 100px;
    padding: 10px;
    background-color: orange;
}

</style>
Hoc CSS3 Animation
```





THUỘC TÍNH ANIMATION-FILL-MODE

- ☐ Thuộc tính này có tác dụng xác định trạng thái của một animation, khi mà animation không được chạy (có thể là animation này đã chạy xong hoặc đang bị delay).
- Cú Pháp: animation-fill-mode: value;
- ☐ Trong đó, **value** có thể là một trong các giá trị sau:
 - none đây là giá trị mặc định. Khi mà animation không hoạt động nữa thì nó sẽ không thêm một style nào vào thành phần.
 - forwards khi mà animation không hoạt động nữa thì nó sẽ apply hết tất cả các thuộc tính cuối cùng của animation
 - backwards khi mà animation không hoạt động nữa nó sẽ apply tất cả các giá trị của thuộc tính trong lần đầu xuất hiện trong keyfames (phụ thuộc vào thuộc tính anmation-direction)



THUỘC TÍNH ANIMATION-FILL-MODE

Ví dụ: sử dụng aniamtion-direction: reverse

```
<style>
    @keyframes move{
        from{ top: 0px;}
        to{ top: 200px;}
}

p{
    animation-name: move;
    animation-duration: 3s;
    animation-direction: normal;
    animation-fill-mode: backwards;
    position: relative;
        width: 100px;
        height: 100px;
        padding: 10px;
        background-color: orange;
}
</style>
Hoc CSS3 Animation
```





THUỘC TÍNH ANIMATION-PLAY-STATE

- ☐ Thuộc tính này có tác dụng xác định trạng thái của animation.
- ☐ Cú pháp: animation-platy-state: value;
- ☐ **Trong đó**, value là 1 trong 2 giá trị sau:
 - running animation đang chạy
 - paused animation đang dừng



THUỘC TÍNH ANIMATION-PLAY-STATE

Ví dụ: animation-play-state: running;

```
<style>
  @keyframes move{
    from{ top: Opx;}
    to{ top: 200px;}
}

p{
    animation-name: move;
    animation-duration: 3s;
    animation-direction: normal;
    animation-fill-mode: both;
    position: relative;
    animation-play-state: running;
    width: 100px;
    height: 100px;
    height: 100px;
    padding: 10px;
    background-color: orange;
}
</style>
Hoc CSS3 Animation
```







GIỚI THIỆU

- ☐ Với transform, bạn có thể xoay, co giãn kích thước hoặc bóp nghiêng hình dạng một phần tử. Ngoài ra nó cũng còn một số tính năng khác cũng liên quan đến việc làm thay đổi hình dạng.
- ☐ Cú pháp: **transform**: function(value);
- ☐ Trong đó: function() là tên hàm làm thay đổi hình dạng và value là giá trị của hàm, mỗi hàm có thể sẽ có cách viết giá trị khác nhau.
 - Skew
 - Scale
 - Rotate
 - translate





☐ Với hàm rotate() bạn có thể thiết lập một đối tượng bị xoay theo độ góc. Ở hàm này bạn có thể thiết lập giá trị kiểu [n]deg (thiết lập giá trị góc, tức là độ) hoặc [n]turn (1 turn = 360 độ).

```
<style>
    #rotate:hover img {
        transform: rotate(90deg);
}

#transition img {
        transition: ls ease-in-out;
}

#transition:hover img {
        transform: rotate(3turn);
}

</style>
<div id="rotate">
        <iny src="https://i.imgur.com/6xM7wdY.png" />
        </div>
```



Hiệu ứng có kèm transition







☐ Với hàm **scale()** bạn có thể thiết lập co giãn kích thước của một phần tử dựa vào trục y (trục thẳng đứng) và trục x (trục ngang), và hàm này bạn sẽ thiết lập là scale(X) hoặc scaleX() và scaleY().

```
    #scale:hover img {
        transform: scaleY(1.2);
    }
    #transition img {
        transition: 1s ease-in-out;
    }
    #transition:hover img {
        transform: scaleX(1.5);
    }
</style>
<div id="scale">
    <img src="https://i.imgur.com/6xM7wdY.png" />
    </div>
<h3>Hiệu ứng có kèm transition</h3>
    <imu src="https://i.imgur.com/6xM7wdY.png" />
    <imu src="https://i.imgur.com/6xM7wdY.png" />
    </div>
</div>
```



Hiệu ứng có kèm transition







□ Bạn có thể kéo một đối tượng nghiêng dựa theo trục Y và trục X với hàm skewX() và skewY(), giá trị bên trong là số [n]deg tương tự rotate().

```
<style>

#skew:hover img {
    transform: skewY(45deg);
}

#transition img {
    transition: 1s ease-in-out;
}

#transition:hover img {
    transform: skew(-45deg);
    transform-origin: 0% 55%;
}

</style>
<div id="skew">
    <img src="https://i.imgur.com/6xM7wdY.png" />
    </div>
<h3>Hiệu ứng có kèm transition</h3>
<div id="transition">
    <img src="https://i.imgur.com/6xM7wdY.png" />
    </div</pre>
```



Hiệu ứng có kèm transition

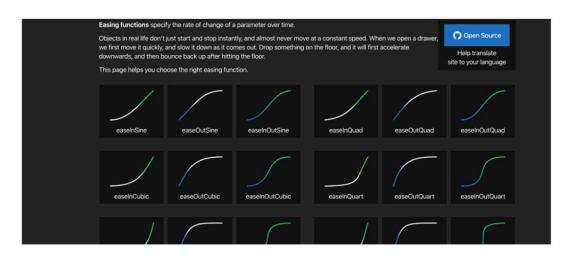


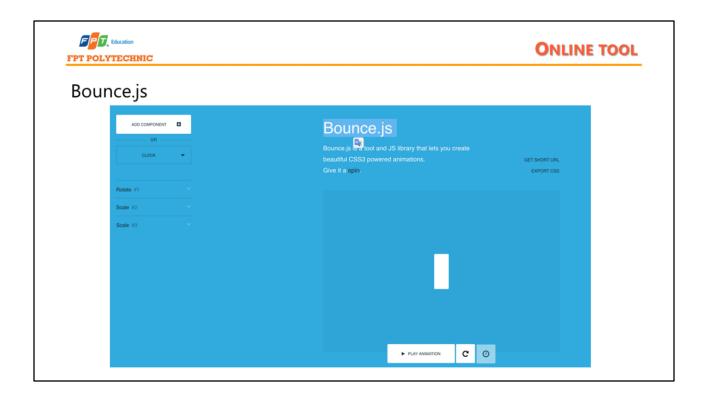


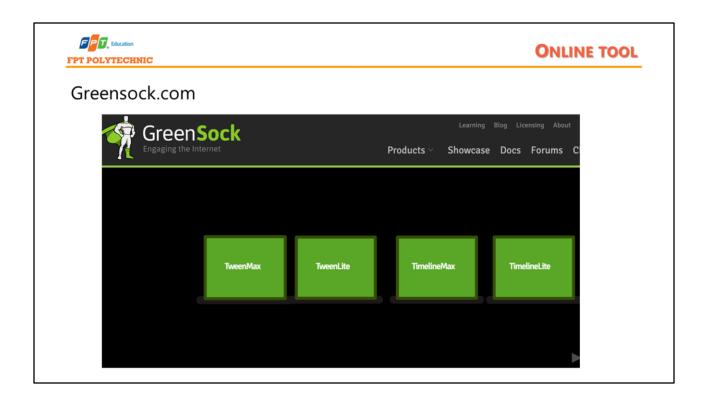




Easings.net









SUMARRY

- Tổng quan về animation và transform
- Giới thiệu các thành phần của animation
- Giới thiệu các thành phần của transform
- Giới thiệu tool online tạo animation



