

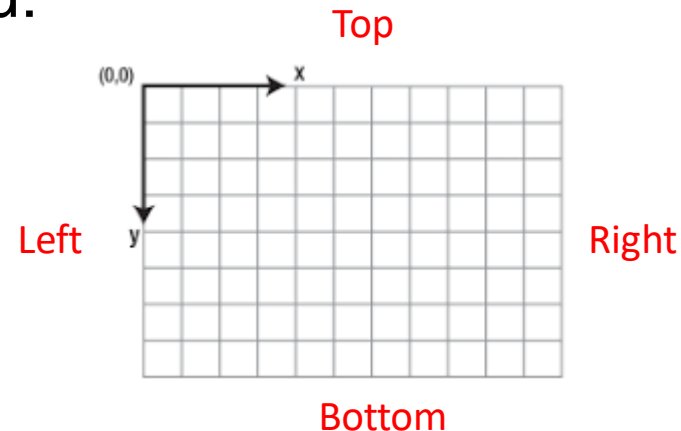
Front-End Essentials

Session 1: CSS Position





- Cho phép chúng ta định vị các phần tử/ vùng chứa các phần tử ở những vị trí khác nhau;
- Cho phép chúng ta thiết lập vị trí của các vùng/phần tử theo mối tương quan vị trí nào đó với nhau;
- CSS Position có những thuộc tính sau:
 - ✓ Position: **static**
 - ✓ Position: **relative**
 - ✓ Position: **absolute**
 - ✓ Position: **fixed** hoặc **sticky**



Position: static

- Đây là giá trị **position mặc định** của tất cả các phần tử HTML;
- Mỗi vùng/phần tử được **cố định nằm trên một hàng** và **được sắp xếp từ trên xuống dưới** (Nếu có nhiều vùng/phần tử liền kề
- Các thuộc tính định vị: **top, left, right, bottom và z-index không sử dụng được** (không có tác dụng);



```
#element_1 {  
  position: static;  
  width: 200px;  
  height: 200px;  
}
```

```
#element_2 {  
  position: static;  
  width: 200px;  
  height: 200px;  
}
```

```
#element_3 {  
  position: static;  
  width: 200px;  
  height: 200px;  
}
```

Position: relative

- Các đối tượng/vùng được định vị tương tự như **position:static**
- Nhưng chúng ta **sử dụng được** các thuộc tính định vị: **top, left, bottom, right và z-index**
 - ✓ **Top:** dịch chuyển vùng xuống dưới (so với vùng bên trên nó)
 - ✓ **Left:** dịch chuyển vùng sang bên trái
 - ✓ **Right:** dịch chuyển vùng cách xa vùng bên phải
 - ✓ **Bottom:** dịch chuyển vùng cách xa vùng bên dưới
- Khi một vùng được xác định là relative thì nó được coi như là vùng cha



```
#element_1 {  
  position: relative;  
  width: 200px;  
  height: 200px;  
}
```



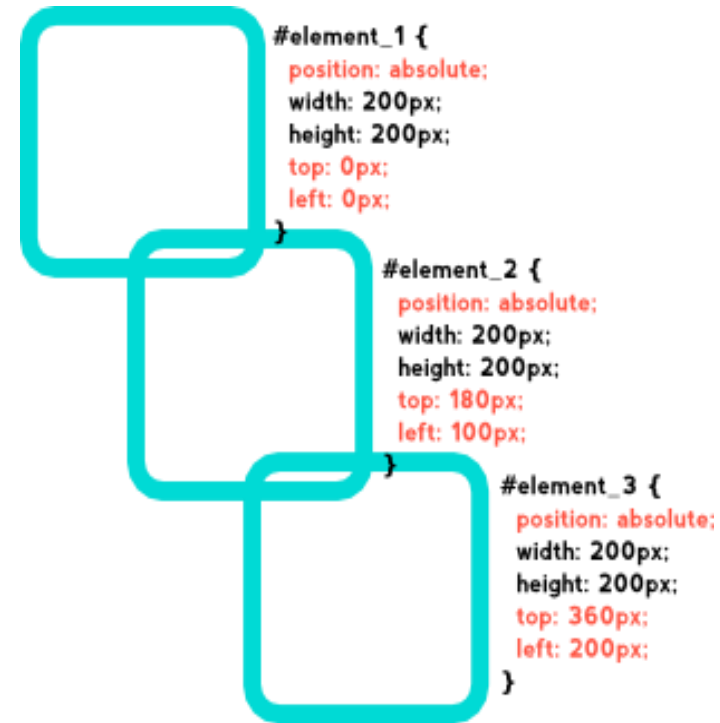
```
#element_2 {  
  position: relative;  
  width: 200px;  
  height: 200px;  
  left: 200px;  
}
```



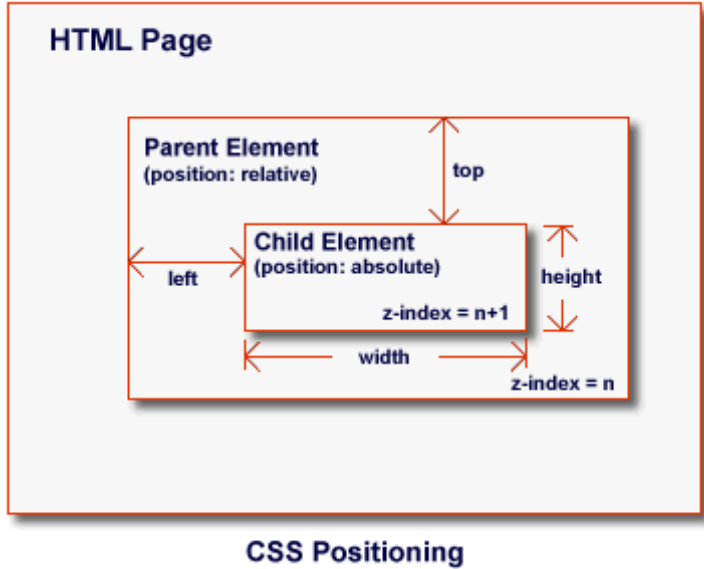
```
#element_3 {  
  position: relative;  
  width: 200px;  
  height: 200px;  
}
```

Position: Absolute

- Khi sử dụng thuộc tính **absolute** thì vùng đó sẽ được **đưa về góc tọa độ trên cùng bên trái** của màn hình hoặc vùng cha (nếu có **vùng cha chứa nó**)
- Vùng sẽ được **xác định là vùng con** của vùng có **position:relative**
- Sử dụng được các thuộc tính định vị để **dịch chuyển vị trí** của vùng
- Các vùng sử dụng absolute có thể chồng lấn lên nhau nếu các giá trị thuộc tính định vị giao nhau;



- **Cố định** vùng chứa các phần tử ở vị trí nào đó trên trang/Layout – Kể cả khi màn hình có thanh cuộn kéo xuống;
- Khi sử dụng thuộc tính này cho vùng thì đồng nghĩa với **việc nó đang có mối quan hệ vị trí với trình duyệt (browser)**
- **Không ảnh hưởng tới vị trí của các phần tử khác (Nhưng có thể bị chồng lấn – Nếu cùng tạo độ)**
- Sử dụng được các thuộc tính định vị: **top, left, right và bottom để dịch chuyển vị trí**
- Các trường hợp thường dùng: Menu, Các Button Media connect, Messenger chat, các vùng quảng cáo, ...



- **Static:** giá trị mặc định của trang html, **không có các thuộc tính top, bottom, left hay right** (Hiển thị tuần tự, từ trái sang phải, từ trên xuống dưới)
- **Fixed:** cố định vị trí của một yếu tố theo màn hình, nó không di chuyển khi chúng ta cuộn chuột
- **Relative:** Tương tự như static, nhưng có thể **sử dụng những thuộc tính left, top, right, bottom để định vị**
- **Absolute:** Có thể sử dụng những thuộc left, top, right, bottom **để định vị trí tương đối của thẻ so với một đối tượng chứa (Cha)**

- Nếu sử dụng cùng lúc các thuộc tính **Left** và **right** thì thuộc tính **left** sẽ có hiệu lực (nếu **direction:ltr** – mặc định), **right** sẽ có hiệu lực nếu **direction:rtl**;
- Nếu sử dụng cùng lúc **top** và **bottom** thì **top** sẽ có hiệu lực

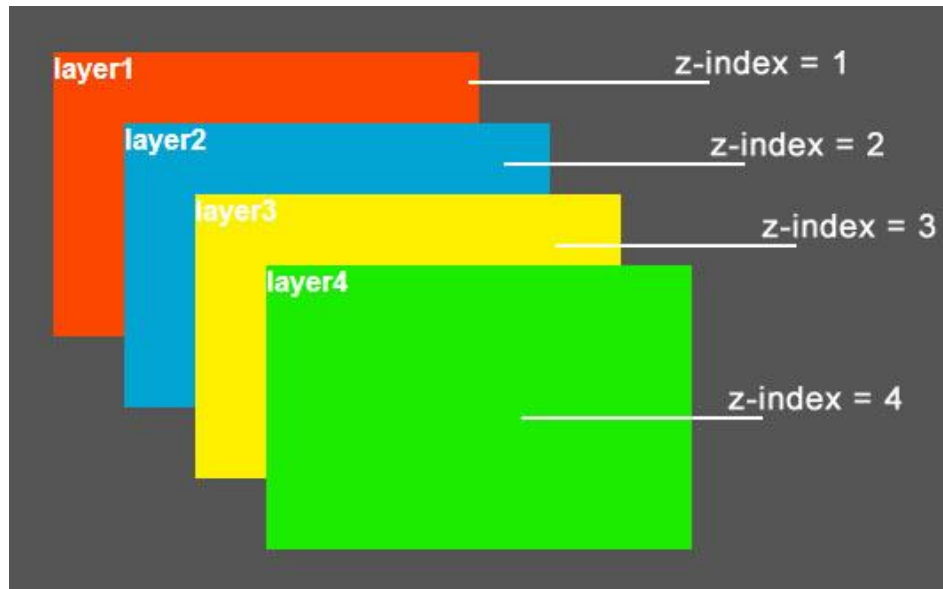


```
<div class="slideshow">
  <!-- vùng chứa Ảnh --->
  <div class="main-pic">
    <a href="">
      
    </a>
  </div>
  <!-- vùng Caption chứa nội dung bên trong Slide -->
  <div class="content">
    <h2>
      <a href=""> The Vietnam national under-22, under-23 football team</a>
    </h2>
    <p>The Vietnam national under-22, under-
    23, and Olympic football team represents Vietnam in international football tournaments at the
    under-22 and under-23 age level</p>
  </div>
  <!-- Các nút tiến - lùi -->
  <ul>
    <li><a href="#" class="next">Next</a></li>
    <li><a href="#" class="prev">Prev</a></li>
  </ul>
</div>
```

```
.slideshow{
  margin: 0 auto; width: 900px;
  height: 480px; position: relative;
}
.slideshow .main-pic, img{
  width: 100%; height: 100%;
}
.slideshow .main-pic a,
.slideshow .content a{
  text-decoration: none; color: black;
}
.slideshow .content{
  position: absolute; width: 80%;
  height: auto; bottom: 1rem;
  right: 0; background-color: rgba(51, 153, 51, 0.3);
}
```

```
.content p, h2{
  padding: 0.5rem; margin: 0;
}
.slideshow ul{
  list-style: none; margin: 0;
  padding: 0;
}
.slideshow ul a{
  position: absolute;
  background-color: rgba(51, 153, 51, 0.3);
  text-indent: -9999px;
}
```

```
.slideshow ul a.next{
  top: 54%; right: 0;
  padding: 1rem 1.6rem;
  background-image: url('../images/next.png');
  background-repeat: no-repeat;
  background-size: 45px 45px;
}
.slideshow ul a.prev{
  top: 65%; right: 0;
  padding: 1rem 1.6rem;
  background-image: url('../images/prev.png');
  background-repeat: no-repeat;
  background-size: 50px 50px;
}
```



- Sử dụng trong để thiết lập thứ tự của các vùng chồng lấn lên nhau tại cùng một vị trí;
- Vùng nào có giá trị z-index cao hơn thì sẽ nằm trên cùng;
- Chỉ có tác dụng khi đi cùng với các vùng sử dụng position: absolute, relative, fixed, hoặc sticky

Exercise 1



Exercise 2



Exercise 3






Front-End Essentials

Session 2: CSS Animation

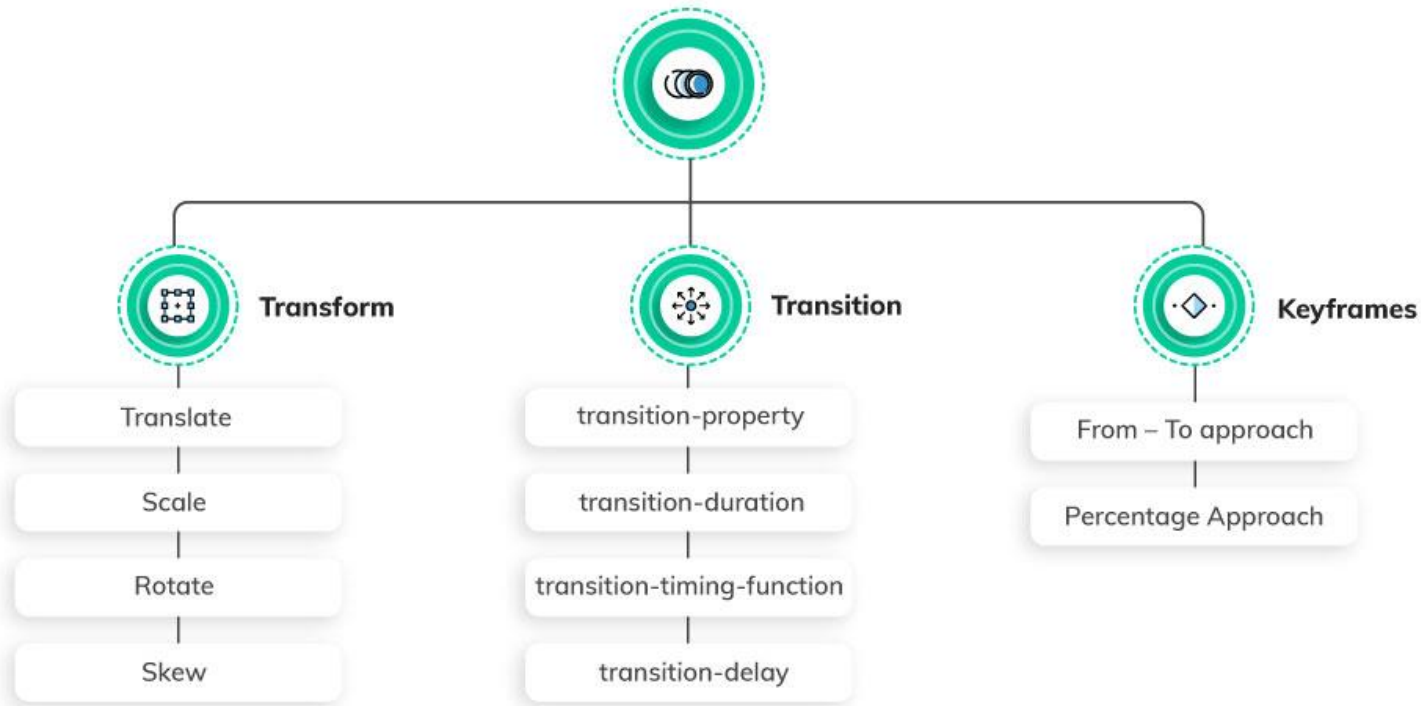


- Tạo hiệu ứng cho các phần tử HTML mà không cần JavaScript hay Flash;
- Hỗ trợ hầu hết **tất cả** các nền tảng **trình duyệt**

Property					
@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

- Thay đổi trạng thái, kiểu dáng của một phần tử (Thẻ, thuộc tính của thẻ) nào đó trong trang
- Chúng ta có thể thay đổi nhiều thuộc tính CSS của một phần tử (Thẻ, thuộc tính của thẻ)
- Để sử dụng CSS Animation hiệu quả, chúng ta cần:
 - ✓ CSS Transforms
 - ✓ CSS Transitions
 - ✓ Keyframes

CSS Animations & Transitions



- CSS Transition thường dùng để thay đổi trạng thái của phần tử mượt hơn, mềm mại hơn;
- Có 04 thuộc tính khi làm việc với CSS Transition:
 - ✓ **Transition-property** (Bắt buộc) → Xác định các thuộc tính CSS của phần tử sẽ được thay đổi
 - ✓ **Transition-duration** (Bắt buộc) → Xác định thời gian hoàn thành sự thay đổi (bằng giây (s) hoặc Mi ly giây (ms))
 - ✓ **Transition-timing-function** → Xác định tốc độ để hoàn thành chuyển đổi theo một dạng hình học nào đó: **ease, linear, ease-in, ease-out, cubic-bezier, ...**
 - ✓ **Transition-delay** → xác định độ trễ (s hoặc ms) trước khi bắt đầu sự thay đổi
- Có thể sử dụng kết hợp 04 thuộc tính trên trong 01 thuộc tính **transitions** →Cú pháp:
transition: (property) (duration) (transition-timing-function) (transition-delay);

CSS Transitions – Ví dụ



Transition

transition-property

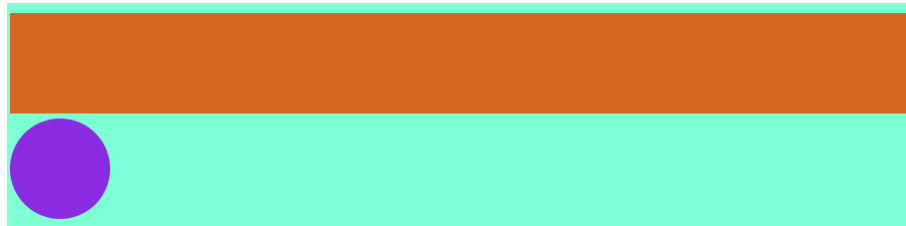
transition-duration

transition-timing-function

transition-delay



Thay đổi trạng thái khi
Hover



```
.wrapper:hover .box2{  
  transition-property: border-radius, background-color;  
  border-radius: 50%;  
  background-color: blueviolet;  
  transition-timing-function: ease-in;  
}
```

```
<div class="wrapper">  
  <div class="box box1"></div>  
  <div class="box box2"></div>  
</div>
```

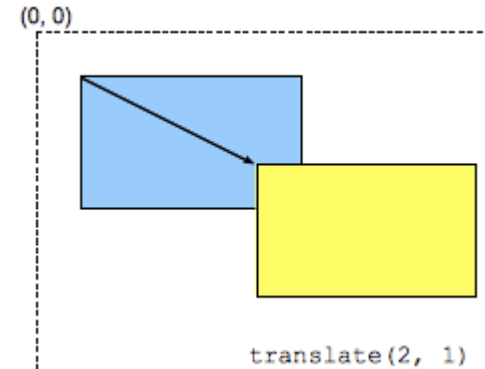
```
.wrapper{  
  margin: 0 auto;  
  width: 900px;  
  height: auto;  
  background-color: aquamarine;  
  padding: 5px 3px;  
}  
.box{  
  width: 100px;  
  height: 100px;  
  margin: 5px 0;  
  background-color: brown;  
  transition-duration: 0.5s;  
}  
.wrapper:hover .box1{  
  transition-  
property: width, background-color;  
  width: 100%;  
  background-color: chocolate;  
  transition-timing-  
function: ease-in-out;  
}
```



- CSS transforms thường được dùng để:
 - ✓ Dịch chuyển phần tử → **translate**
 - ✓ Phóng to, thu nhỏ → **scale**
 - ✓ Xoay → **rotate**
 - ✓ Làm nghiêng phần tử → **skew**
 - Trong không gian 2D hoặc 3D

- CSS Transform – **translate**: dịch chuyển phần tử trong không gian 2D hoặc 3D

translate(x,y)	Dịch chuyển phần tử trong không gian 2D theo giá trị tọa độ X và Y
translate3d(x,y,z)	Dịch chuyển phần tử trong không gian 3D theo giá trị tọa độ X, Y và Z
translateX(x)	Dịch chuyển phần tử trong không gian 2D theo chiều trục X
translateY(y)	Dịch chuyển phần tử trong không gian 2D theo chiều trục Y
translateZ(z)	Dịch chuyển phần tử trong không gian 3D theo chiều trục Z



CSS Transform – translate



```
<div class="wrapper">  
  <div class="circle first"></div>  
  <div class="circle moving"></div>  
  <div class="circle second"></div>  
</div>
```

```
.wrapper{  
  width: 900px; height: 100px;  
  background-color: antiquewhite;  
  margin: 0 auto; position: relative;  
}  
.circle{  
  position: absolute;  
  width: 50px; height: 50px;  
  margin-top: 25px; border-radius: 50%;  
}  
.first{  
  background-color: red;  
  left: 5px; z-index: 20;  
}  
.second{  
  background-color: green;  
  right: 5px; z-index: 20;  
}  
.moving{  
  left: 10px; top: 5px; z-index: 10;  
  width: 40px; height: 40px;  
  background-color: blueviolet;  
  transition: transform 1s linear;  
}  
.wrapper:hover .moving{  
  transform: translateX(840px);  
}
```

- CSS Transform – **scale** dùng để phóng to/thu nhỏ phần tử

scale(x,y)	Thay đổi chiều cao và chiều rộng của phần tử trong không gian 2D
scaleX(n)	Thay đổi chiều rộng của phần tử trong không gian 2D
scaleY(n)	Thay đổi chiều của phần tử trong không gian 2D

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

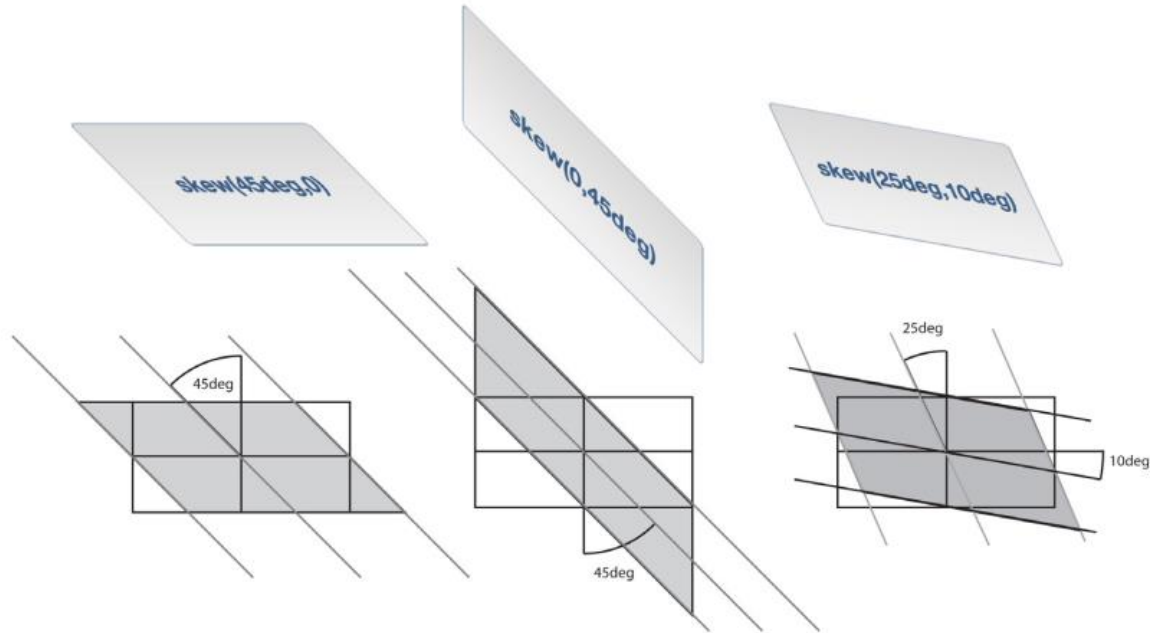
```
.square {  
  background-color:aquamarine;  
  border-radius: 5px;  
  height: 100px;  
  margin: 100px;  
  transition: transform 1s;  
  width: 100px;  
}  
.square:hover{  
  transform: scale(2);  
}
```



Hover



CSS Transform - skew



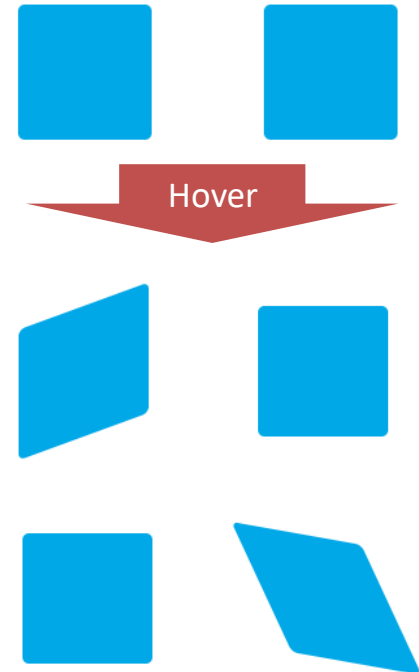
■ CSS Transform – skew làm nghiêng phần tử

<code>skew(<i>x-angle</i>,<i>y-angle</i>)</code>	Nghiên phần tử trong không gian 2D theo trục X, Y (độ)
<code>skewX(<i>angle</i>)</code>	Nghiên phần tử trong không gian 2D theo trục X (độ)
<code>skewY(<i>angle</i>)</code>	Nghiên phần tử trong không gian 2D theo trục Y (độ)

```
<div class="container">
  <div class="square square1"></div>
  <div class="square square2"></div>
</div>
```

```
.square1:hover {
  transform: skewY(-20deg);
}
.square2:hover {
  transform: skew(25deg, 10deg);
}
```

```
.container {
  margin-top: 180px;
  text-align: center;
}
.square {
  display: inline-block;
  background-color: #00A8E8;
  border-radius: 5px;
  height: 100px; width: 100px;
  margin: 20px 40px;
  transition: transform 0.3s ease;
}
```



- CSS Transforms – **rotate**: dùng để xoay đối tượng

rotate(*angle*)

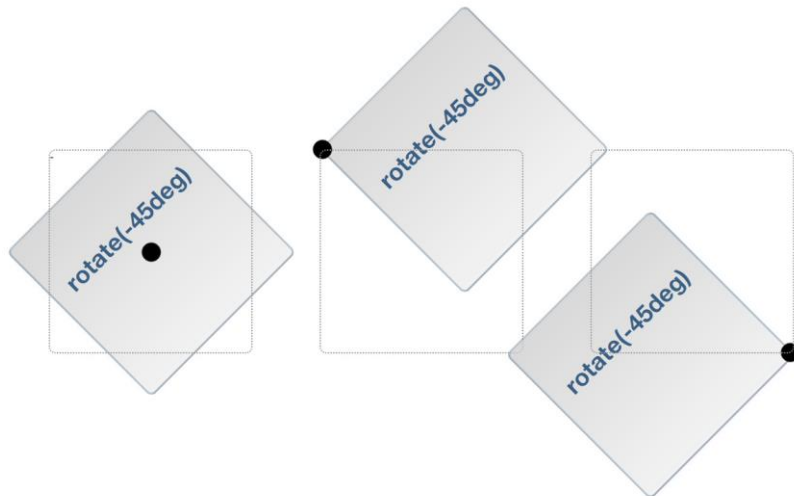
Xoay đối tượng trong không gian 2D, đơn vị là deg (độ)

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



```
.droplet{  
  background: mediumaquamarine;  
  border-radius: 2% 50%;  
  height: 100px;  
  margin: 100px;  
  transition: all 3s;  
  transition-timing: ease-in-out;  
  width: 100px;  
}  
.droplet:hover{  
  transform: rotate(360deg);  
}
```

- Dùng để xoay vùng chứa đối tượng tại một điểm tọa độ nào đó.
- **Mặc định là ở giữa** vùng chứa đối tượng

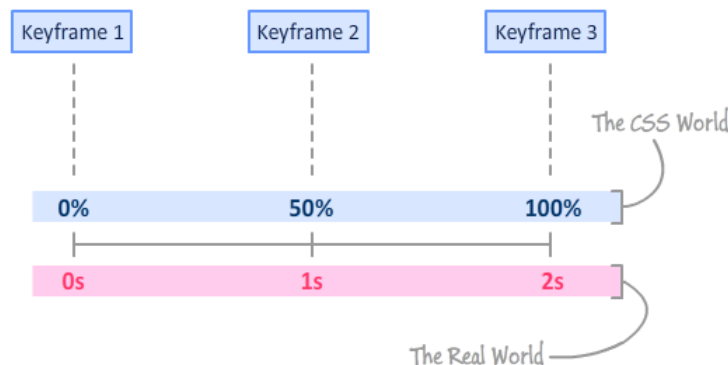


transform-origin: *x-axis y-axis z-axis*|initial|inherit;

- Các giá trị của **x-axis** và **y-axis**: left, center, right, length, %
- Giá trị của z-axis: length

- Thay đổi trạng thái của phần tử theo từng khoảng thời gian mong muốn;
- Có 02 cấu trúc tiếp cận:
 - ✓ Cấu trúc **from – to**
 - ✓ Cấu trúc **tỉ lệ phần trăm**

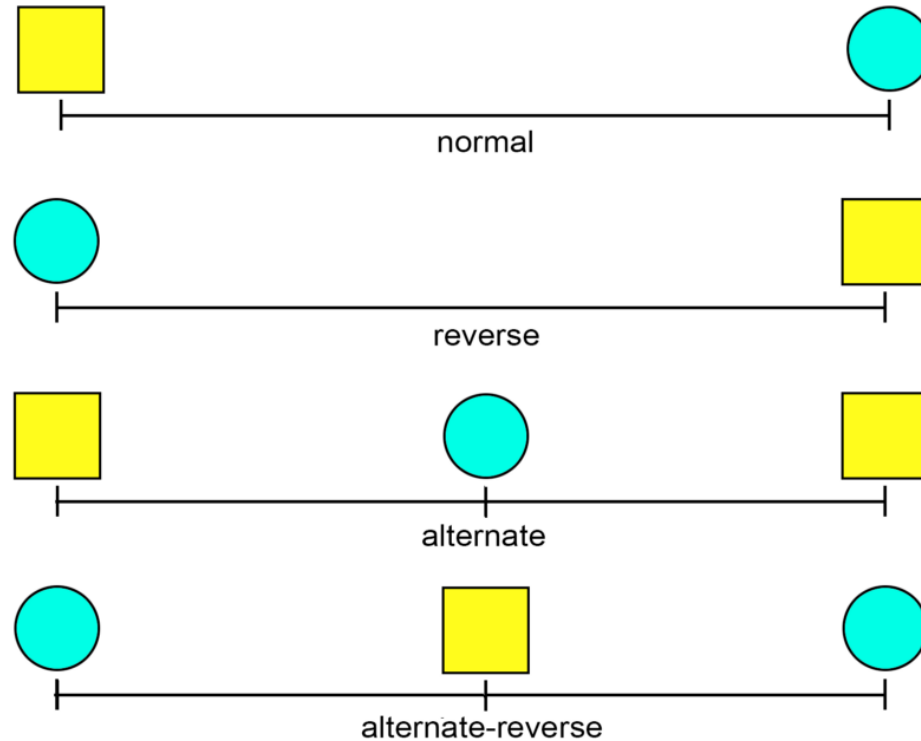
```
@keyframes animation_name {  
  from { } /* tương đương 0%*/  
  to { }   /*tương đương với 100%*/  
}
```



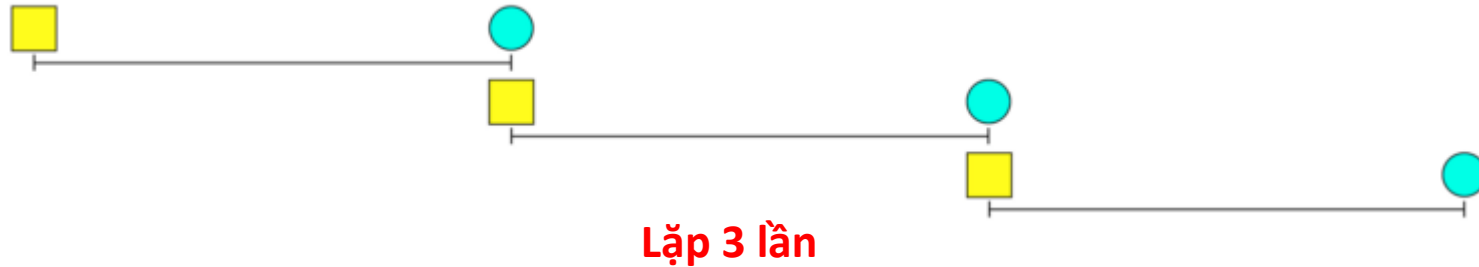
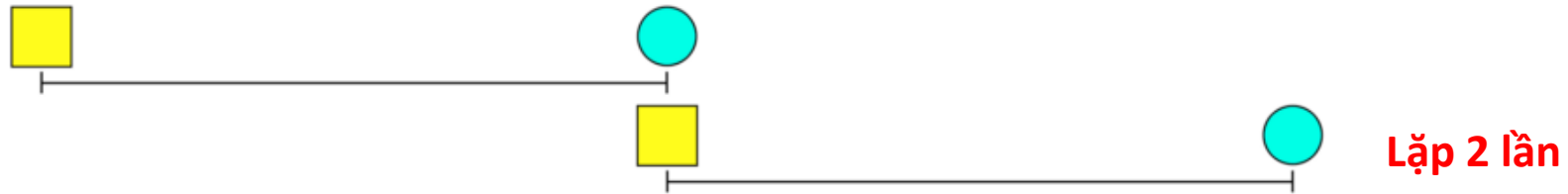
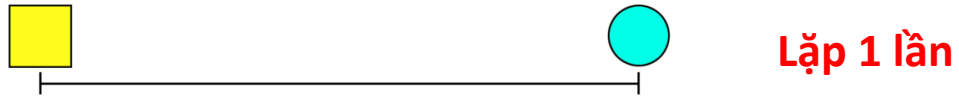
Các thuộc tính CSS3 Animation

animation	Thẻ vắt tắt để thiết giá trị cho tất cả các thuộc tính animation dưới đây . <i>animation: [name] [duration] [timing-function] [delay] [iteration-count] [direction] [fill-mode] [play-state];</i>
animation-name	Tên của animation đã được đặt trong @keyframes
animation-duration	Xác định khoảng thời gian (giây hoặc mili giây) để hiệu ứng hoàn thành
animation-timing-function	Xác định khoảng thời gian hoàn thành hiệu ứng theo một dạng hình học nào đó: ease, linear, ease-in, ease-out, ease-in-out, initial, inherit
animation-delay	Xác định khoảng thời gian chờ trước khi hiệu ứng bắt đầu
animation-iteration-count	Xác định số lần mà chu trình hiệu ứng sẽ thực hiện (Số lần lặp) trước khi dừng
animation-direction	Xác định xem hiệu ứng sau khi kết thúc có lặp lại hay không? Và theo hướng nào? Normal (Không lặp lại 0 -> 100), reverse (có lặp lại 0 → 100 → 0), alternate(0-100-100-0)
animation-fill-mode	Xác định chế độ áp dụng cho phần tử trước và sau khi thực hiện
animation-play-state	Xác định việc hiệu ứng được chạy hay tạm dừng
@keyframes	Đặt tên và xác định các thuộc tính hiệu ứng cho phần tử

animation-direction



animation-iteration-count



Cách tiếp cận from - to

Ví dụ: Tạo hiệu ứng loading với 05 vùng trạng thái



```
<div class="loadingdiv">
```

```
>
```

```
<span></span>
```

```
<span></span>
```

```
<span></span>
```

```
<span></span>
```

```
<span></span>
```

```
</div>
```

```
/* Thay đổi màu sắc của item 2 */
.loadingdiv span:nth-of-type(2) {
  background: #207178;
  animation-delay: 0.2s;
  animation-delay: 0.2s;
}
/* Thay đổi màu sắc của item 3 */
.loadingdiv span:nth-of-type(3) {
  background: #FF9666;
  animation-delay: 0.4s;
  animation-delay: 0.4s;
}
/* Thay đổi màu sắc của item 4 */
.loadingdiv span:nth-of-type(4) {
  background: #FFE184;
  animation-delay: 0.6s;
  animation-delay: 0.6s;
}
/* Thay đổi màu sắc của item 5 */
.loadingdiv span:nth-of-type(5) {
  background: #F5E9BE;
  animation-delay: 0.8s;
  animation-delay: 0.8s;
}
}
```

```
.loadingdiv{
  margin: 0 auto;
  width: 200px;
  height: 100px;
  margin-top: 50px;
}
.loadingdiv span{
  display: inline-block;
  vertical-align: middle;
  width: 15px;
  height: 15px;
  background: #174C4F;
  border-radius: 50%;
  box-shadow: 1px 1px 1px #444444;
  /* CSS 3 Animation - từng thành phần */
  animation-name: loading;
  animation-duration: 0.9s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
  /* Rút gọn trong thuộc tính animation */
  /* animation: loading 0.9s infinite alternate; */
}
```

```
/* Thiết lập các giá trị hiệu ứng cho @keyframe có tên là loading */  
@keyframes loading {  
  from {  
    width: 15px;  
    height: 15px;  
    transform: translateZ(0);  
  }  
  to {  
    width: 25px;  
    height: 25px;  
    transform: translateZ(-25px);  
  }  
}
```

Đổi `translateZ()` sang `stranslateY()`
để cảm nhận sự khác biệt



Chú ý: để hiệu ứng animation thực hiện được trên tất cả các trình duyệt khác thì cần thêm Các tiền tố sau đây vào các thẻ thuộc tính animation:

`/* Chrome, Safari 4+ */`
`-webkit-`

`/* Firefox 5+ */`
`-moz-`

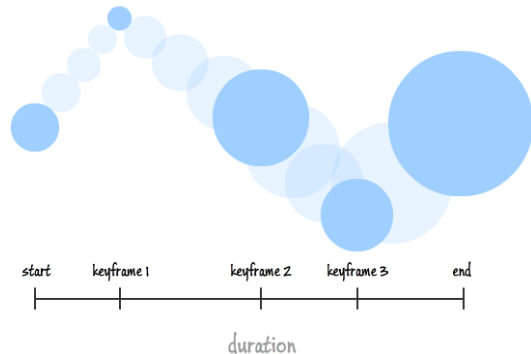
`/* Opera 12+ */`
`-o-`

Keyframes – tiếp cận %

```
<div class="wrapper">  
  <div class="element"></div>  
</div>
```

```
.wrapper{  
  margin: 50px auto;  
  margin-top: 150px;  
  width: 900px;  
}  
.element{  
  width: 50px;  
  height: 50px;  
  background-color:aqua;  
  border-radius: 50%;  
  position: relative;  
  top: 0;  
  left: 0;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}  
.wrapper:hover .element{  
  animation: jump 1s infinite alternate  
;  
}
```

```
@keyframes jump{  
  0%{  
    left: 0; top: 0;  
  }  
  25%{  
    left: 80px; top: -80px;  
    transform: scale(0.8);  
  }  
  50%{  
    left: 200px;  
    top: 0;  
    transform: scale(1.5);  
  }  
  75%{  
    left: 280px;  
    top: -80px;  
    transform: scale(1.3);  
  }  
  100%{  
    left: 450px;  
    top: 0px;  
    transform: scale(2);  
  }  
}
```



Case study Animation với hình ảnh

Làm mờ ảnh khi hover → sử dụng **opacity** để thiết lập độ trong suốt của ảnh **[0-1]**



Opacity:0.5



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

```
.ima{  
  width: 800px;  
  height: 450px;  
  margin: 0 auto;  
}  
.ima img{  
  width: 100%;  
  height: 100%;  
  transition: opacity 1s ease-out;  
}  
.ima img:hover{  
  opacity: 0.5;  
}
```

Thay đổi ảnh khi hover vào Ảnh

Ảnh ban đầu



Thay đổi ảnh khác khi hover

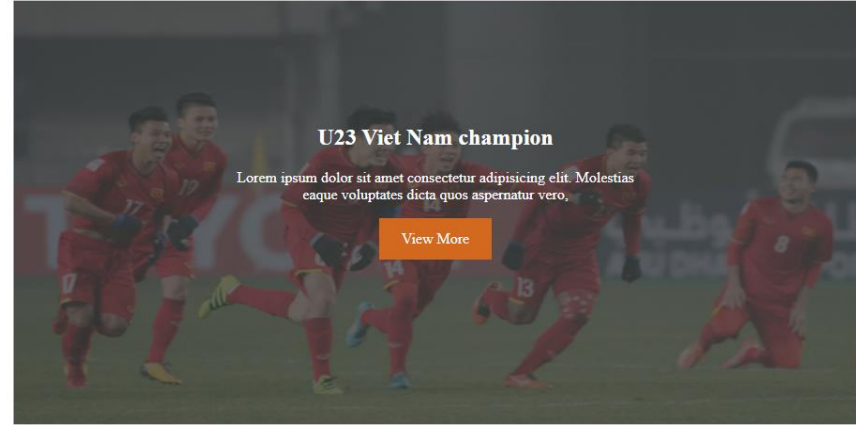


```
.image-area{
  width: 800px;
  height: 450px;
  margin: 0 auto;
  position: relative;
}
.image-area img.origin: hover{
  opacity: 0;
}
```

```
<div class="image-area">
  <!-- Image sẽ thay thế -->
  
  <!-- Image ban đầu -->
  
</div>
```

```
.image-area img{
  width: 100%;
  height: 100%;
  position: absolute;
  left: 0; top: 0;
  transition: opacity 1s ease-in;
```

Một đối tượng khác **overlay** ảnh

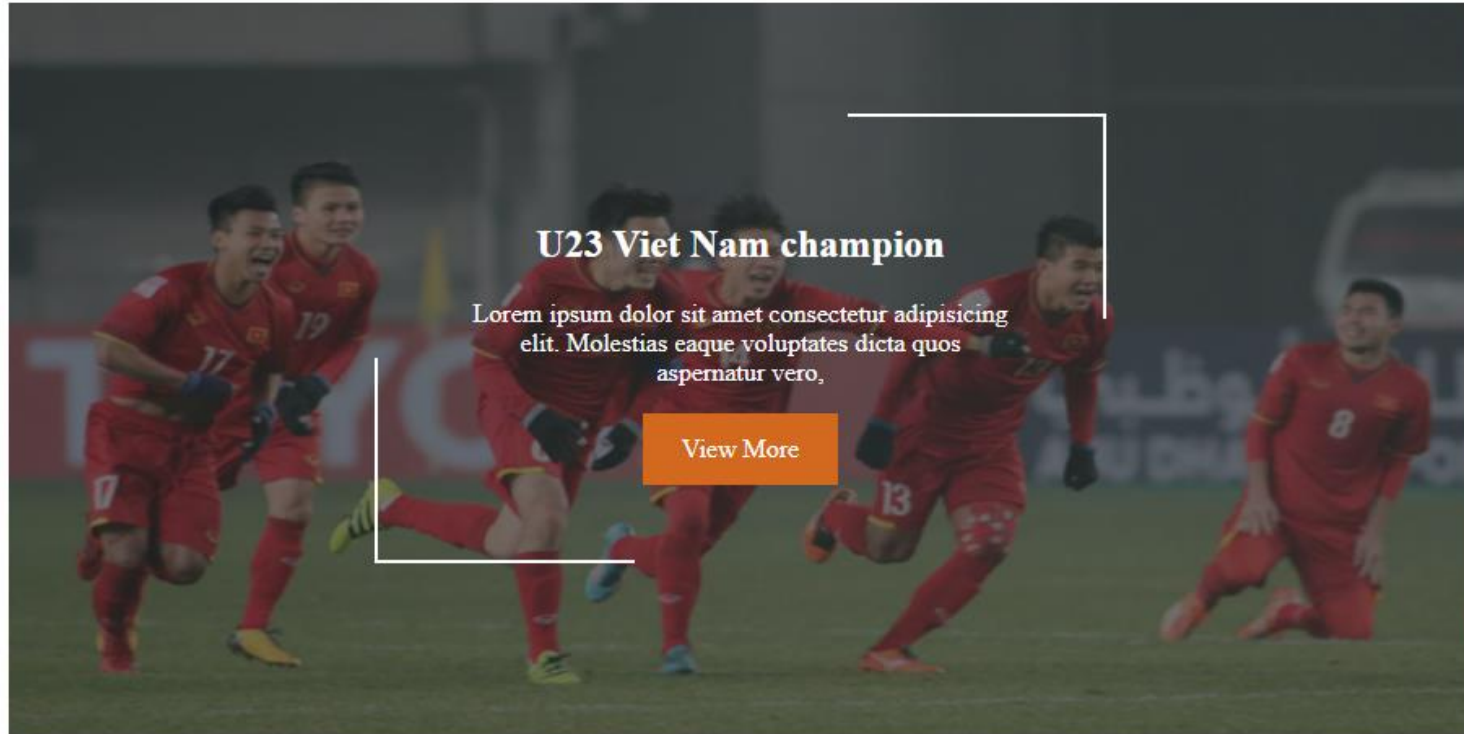


```
<div class="image-area">
  
  <div class="overlay-content">
    <div class="midle">
      <h2><a href="">U23 Viet Nam champion</a></h2>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias eaque voluptates dicta quos aspernatur v
ero</p>
      <a href="#" class="more">View More</a>
    </div>
  </div>
</div>
```

```
.image-area{
  width: 900px;
  height: 450px;
  margin: 0 auto;
  position: relative;
}
.image-area img{
  width: 100%;
  height: 100%;
  opacity: 1;
  transition: 0.5s ease-in;
}
.image-area .overlay-content{
  opacity: 0;
  width: 100%;
  height: 100%;
  position: absolute;
  top: 0;
  left: 0;
  color: white;
  transition: 0.5s ease-in;
}
.midle{
  position: relative;
  width: 50%;
  left: 25%;
  top: 25%;
}
```

```
.midle a{
  display: block;
  width: 100%;
  text-decoration: none;
  text-align: center;
}
.midle a:visited, a:link{
  color: white;
}
.midle h2 a:hover{
  color: chocolate;
}
.midle p{
  display: block;
  width: 100%;
  text-align: center;
  box-sizing: border-box;
  padding: 0rem 0.5rem;
}
```

```
.midle a.more{
  width: 120px;
  margin: 0 auto;
  box-sizing: border-box;
  padding: 0.8rem 1rem;
  background-color: chocolate;
  color: white;
}
.midle a.more:hover{
  background-color:white;
  color: chocolate;
}
.image-area:hover img{
  opacity: 0.5;
}
.image-area:hover .overlay-content{
  opacity: 1;
  background-color: rgba(18, 16, 16, 0.6);
}
```

Các trường hợp tương tự



Good Job

Thank you

