

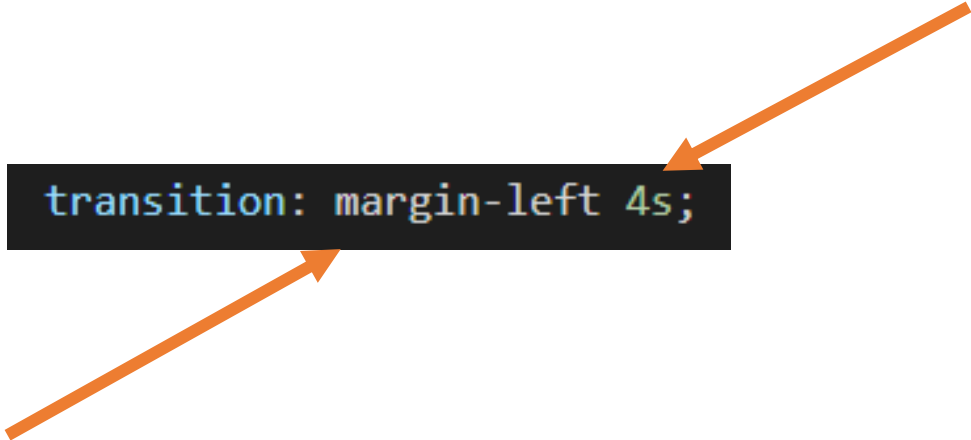
Transición

Css

Transición

Permite definir la transición entre dos estados de un elemento de forma interpolada.

Tiempo que se demorará en transicionar a un nuevo valor.



```
transition: margin-left 4s;
```

The diagram features a central black box containing the CSS code 'transition: margin-left 4s;'. Two orange arrows originate from the text blocks. One arrow points from the text 'Propiedad que no podrá actualizar su valor instantáneamente, sino deberá ir transicionando en el tiempo.' to the word 'transition' in the code. The other arrow points from the text 'Tiempo que se demorará en transicionar a un nuevo valor.' to the value '4s' in the code.

Propiedad que no podrá actualizar su valor instantáneamente, sino deberá ir transicionando en el tiempo.

Estructura

```
/* Aplicar a 1 propiedad */  
/* nombre de la propiedad | duración */  
transition: margin-left 4s;  
  
/* nombre de la propiedad | duración | retardo */  
transition: margin-left 4s 1s;  
  
/* nombre de la propiedad | duración | función | retardo */  
transition: margin-left 4s ease-in-out 1s;  
  
/* Aplicar a 2 propiedades */  
transition: margin-left 4s, color 1s;  
  
/* Aplicar a todas las propiedades que cambien */  
transition: all 0.5s ease-out;
```

Observación: No olvidar que el tiempo lleva una “s” después del número.

Curvas de tiempo

```
/* Aplicar a todas las propiedades que cambien */  
transition: all 0.5s ease-out;
```

En el tercer parámetro se puede indicar una curva de velocidad referente a la transición de valores en un tiempo específico.



Valor posibles:

1. ease - Especifica un efecto de transición con un inicio lento, luego rápido, luego finaliza lentamente (esto es predeterminado)
2. linear - Especifica un efecto de transición con la misma velocidad de principio a fin.
3. ease-in - Especifica un efecto de transición con un inicio lento.
4. ease-out - Especifica un efecto de transición con un final lento.
5. ease-in-out - especifica un efecto de transición con inicio y fin lentos.
6. cubic-bezier(n,n,n,n) - le permite definir sus propios valores en una función de cubic-bezier.

Ejercicio

Escriba el siguiente código.

Observe el comportamiento e indique el por qué

```
div{  
  width: 100px;  
  height: 100px;  
  background-color:  teal;  
  transition: background-color 1s;  
}  
  
div:hover{  
  background-color:  coral;  
}
```

```
<div></div>
```

Ejercicio

Diseñe un div de 100x100 de color coral y letras blancas.

Cuando se posicione el Mouse sobre el, su tamaño debe cambiar a 200x200 y su color a Azul, por último el font-size de la letra debe crecer a 20px.

Utilice transition.

Recordar que se pueden agregar más de una propiedad separándolas por una coma o utilice **all** seguido del tiempo en segundos.

```
transition: propiedad tiempo, propiedad tiempo, propiedad tiempo;
```

Solución

```
div{
  width: 100px;
  height: 100px;
  background-color: ■ coral;
  color: ■ white;

  transition: background-color 1s, width 1s, height 0.5s, font-size 0.5s;
  /*
  transition: all 1s;
  */
}

div:hover{
  background-color: ■ rgb(15, 173, 113);
  font-size: 40px;
  width: 200px;
  height: 150px;
}
```

```
<div >
|   Hi Hi!
</div>
```

Ejercicio

Diseñe un input que se expanda a 300px cuando el mouse se posiciona sobre él.

Ancho inicial: 150px.

Incluya una imagen.

Observación: Para los inputs no funciona la pseudoclase `::before`, por lo tanto se necesita de un contenedor que simule el borde del input.

```
<div class="button">
|   <input type="text" placeholder="Esperando..." />
</div>
```



Esperando...



Esperando...

Solución

```
.button {  
    height: 30px;  
    width: 150px;  
    border: 1px solid gray;  
    padding: 7px;  
    border-radius: 7px;  
    transition: width 0.5s;  
}  
  
.button::before {  
    content: "";  
    display: inline-block;  
    width: 24px;  
    height: 24px;  
    background: url("spaceship.png");  
    vertical-align: middle;  
}
```

```
.button > input {  
    height: 100%;  
    /*Calcula el 100% de su contenedor  
    menos 30px*/  
    width: calc(100% - 30px);  
    outline: none;  
    border-width: 0px;  
}  
  
.button:hover {  
    width: 300px;  
}
```