

Bootcamp Full Stack

CSS avanzado



Transiciones avanzadas

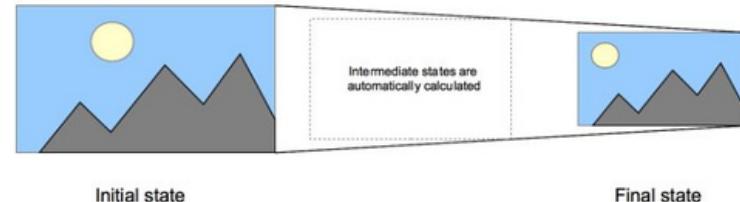
Transiciones

Transition

Son un **pasaje entre un valor de una propiedad a otro valor, de manera gradual, a través de una línea de tiempo.**

Las transiciones se activan generalmente con el **:hover** y el **:focus**. Sin embargo, encontrarán ejemplos en el módulo también con la **:checked**, pseudo clase asociada una casilla o radio chequeado o verificado.

Por ejemplo: al pasar el mouse sobre un elemento o hacer foco en un campo de texto, se activa la transición.



Esta es una imagen tomada de la [página oficial de Mozilla](#)

transition-property

Transition-property es una **propiedad obligatoria** porque si no indico la propiedad a cambiar en la transición no hay transición.

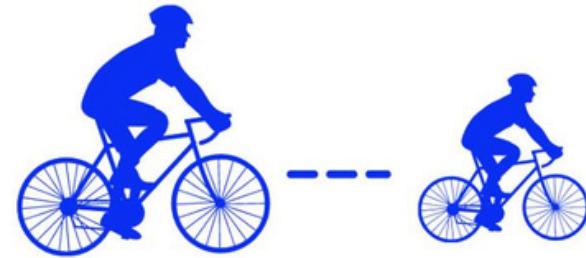
Hace referencia **a la propiedad que se va a modificar.**



transition-duration

Es una propiedad **también obligatoria y su valor predeterminado es 0s**, es decir que si no coloco **transition-duration** tampoco hay transición.

Hace referencia al tiempo que **toma la transición en efectuarse**.



Sintaxis

El bloque de código de la derecha muestra cómo trabajar con **transition-property** y **transition-duration** en una sola línea:

```
<style>
div {
    width: 200px;
    height: 200px;
    border-radius: 20px;
    padding: 20px;
    background: blue;
    color: white;
    border: 2px dashed black;
    transition: background-color 2s;
    cursor:pointer;    }

div:hover { background-color: red; }

</style>
```



Complementos

Algunos complementos

transition-timing-function

Esta propiedad permite indicar el ritmo de la transición. Los valores son:

- **Ease:** valor predeterminado. Inicia despacio, sucede rápido y termina despacio.
- **Linear:** mantiene la transición con el mismo estilo de velocidad en toda su duración.
- **Ease-in:** inicia despacio, luego sucede normal y termina normal.

- **Ease-out:** inicia normal, sucede normal y termina despacio.
- **Ease-in-out:** inicia despacio, sucede normal y termina despacio.

Bezier

Es un poco complicado indicar de manera personalizada en una curva cuál va a ser la velocidad.

Por eso, tenemos aplicaciones para hacerlo, por ejemplo: cubic-bezier.com

```
p { transition: background-color 2s 3s cubic-bezier(.17,.67,.83,.67); }
```

transition-delay

Hace referencia a **cuánto tarda la transición en comenzar**, cuanto mayor sea el tiempo más va a demorar el efecto.

Se puede trabajar con **segundos o ms.**

```
<style>
div {
    width: 200px;
    height: 200px;
    border-radius: 20px;
    padding: 20px;
    background: blue;
    color: white;
    border: 2px dashed black;
    transition: background-color 2s 4s ease-in ;
    cursor:pointer;    }

div:hover { background-color: red; }
```

```
</style>
```



Modificar más de una propiedad

Implementación

Ejemplo: **transition-property** y **transition-duration**

```
<style>
div {
    width: 200px;
    height: 200px;
    border-radius: 20px;
    padding: 20px;
    background: blue;
    color: white;
    border: 2px dashed black;
    transition-property: background, width; transition-duration:5s, 1s; width: 200px; height: 200px;
    background: coral; cursor:pointer; }

div:hover { background-color: red; width: 300px; }

</style>

<div> transiciones </div>
```

Ahora veamos un ejemplo combinado de **transition** y **transform**:

```
<style>
div {
    width: 200px;
    height: 200px;
    border-radius: 20px;
    padding: 20px;
    background: blue;
    color: white;
    border: 2px dashed black;
    transition: all 3s linear ;
    cursor:pointer; }

div:hover { background-color: red; width: 300px; transform: rotate(34deg); }

</style>

<div> transiciones </div>
```

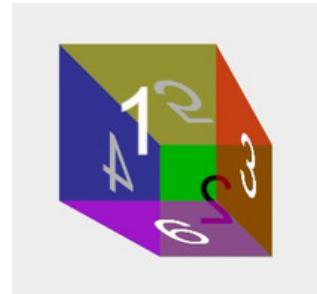


Backface-visibility

Con sus valores **hidden** o **visible** permite trabajar **escondiendo o no la cara de atrás de un elemento** cuando lo rotamos **en 3D**.

En las imágenes de la derecha vemos cómo trabajar con el valor **hidden** y el valor **visible** nos genera un resultado distinto.

```
backface-visibility: visible;
```



```
backface-visibility: hidden;
```



Focus + Transition

Se puede trabajar con la transición en un focus. Por ejemplo: un campo de texto que al hacerle foco se agrande o se resalte con una sombra, como el que encontramos en Google o en tantas otras páginas donde hay un buscador.

```
<!DOCTYPE html>
<head>

<style>
input { width: 200px; height: 30px; transition: all linear 2s; border:none; box-shadow: 2px 1px gray; }

input:focus { width: 400px; }

</style>

<label> Busqueda
<input type="search"></label>

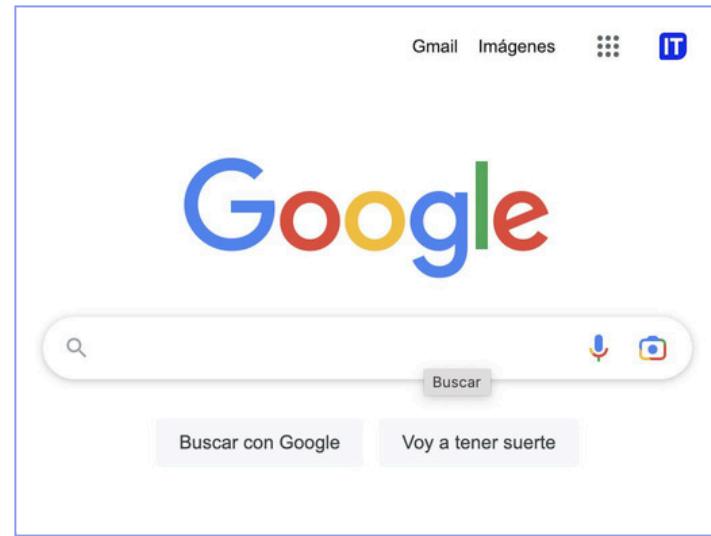
</body>
```



Ejemplo de **transition + focus**



Estado original del campo de búsqueda.



Al pasar el mouse sobre el campo de búsqueda.



Trabajo con *all*

También vemos **all**, que nos permite **afectar a todas las propiedades de la transición**, sin especificarlas **una por una**:

```
<!DOCTYPE html>
<head>

<style>
input { width: 200px; height: 30px; transition: all linear 2s; border:none; box-shadow: 2px 1px gray; }

input:focus { width: 400px; }

</style>

<label> Busqueda
<input type="search"></label>

</body>
```

