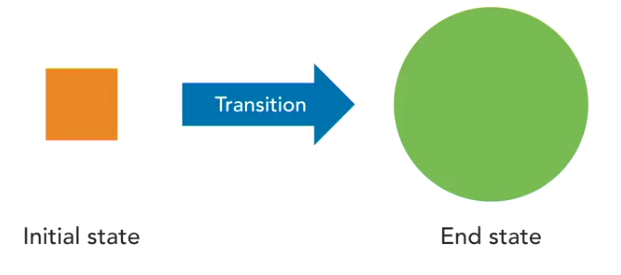
**Transforms and Transitions**

Contenido

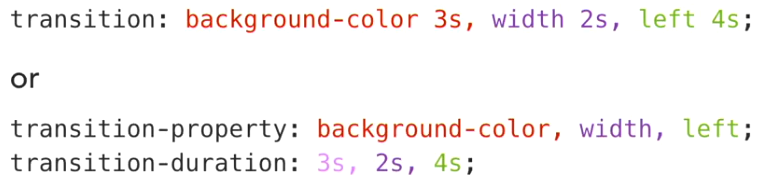
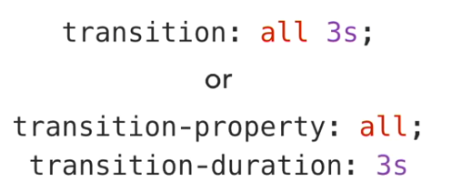
[Capítulo 1. Explorando Transiciones 2](#_Toc11069734)

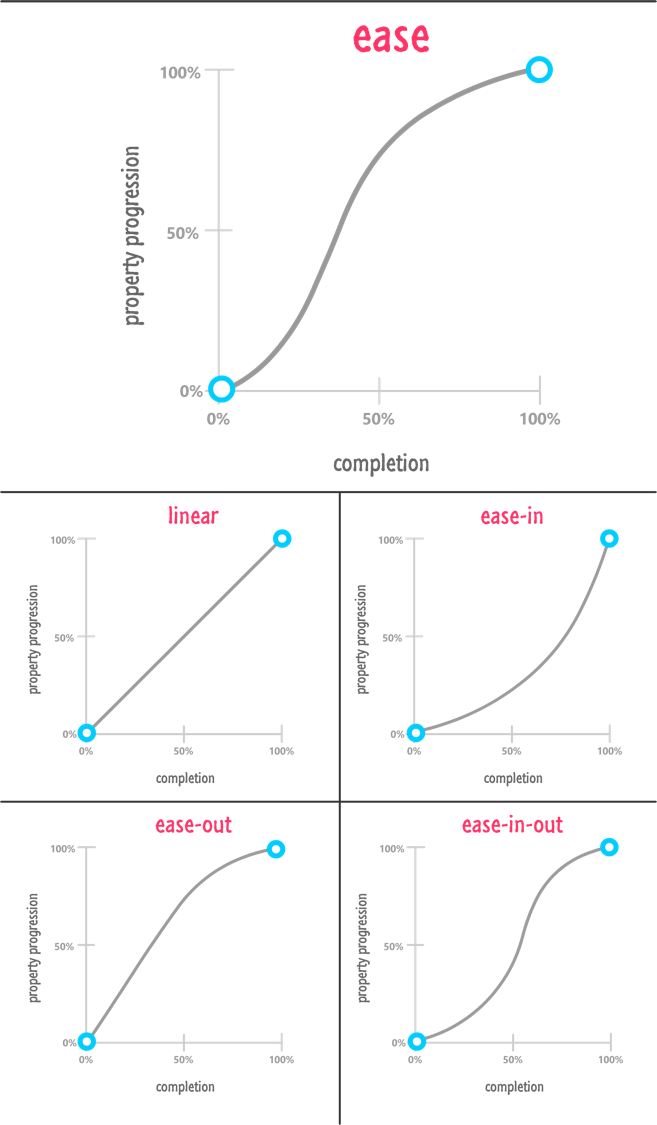
# Capítulo 1. Explorando Transiciones

Transiciones son un cambio de estado a partir de un estado inicial a un estado final, el cual es la apariencia final de un objeto.



La forma de aplicar *transition* en CSS es indicando la propiedad y el tiempo de duración. En caso de elegir varias propiedades, se pueden separar por coma como se muestra en la segunda imagen.





Timing en la transición

El *timing* de la transición está controlado dando un tercer valor a la propiedad de *transition:*

Usa 4 *keyword* que se representan en la imagen de la derecha.

Para usar una curva propia se usa la opción de *cubic-bezier* que toma cuatro valores (x^1 – y^1, x^2 – y^2)

Delays en transiciones

Es el último parámetro de la propiedad de *transition* en CSS.



La razón por la que se retrasa una transición es que cuando se tienen distintos estilos que están cambiando y queremos que se apliquen a distintos tiempos o en un orden secuencial. Por ejemplo, cuando se hace un *hover* a veces indicamos que cambie el tamaño, el color y varias propiedades y por defecto cambiarian todas al mismo tiempo. Sin embargo, el *delay*, podemos retrasar unas propiedades de las otras.

Transiciones asimétricas 01-04

La transición es simétrica cuando va y viene del estado inicial al final usando la misma transición. Sin embargo, a veces se desea que haya una transición partiendo del estado inicial y otra distinta para devolverse, a esto se le conoce como una transición asimétrica.

A continuación, se muestran las diferencias a tener en cuenta cuando se define una transición simétrica y asimétrica:

* **Simétrica:**

Cuando se aplica la transición en el *estado inicial*, no hay sorpresa alguna. Se aplica los mismos efectos tanto para el estado inicial como al final.

figure {

    width: 100px;

    height: 100px;

    transition: left 2s ease-out, top 2s ease-out, width 3s ease, height 3s ease;

}

figure:hover {

    width: 400px;

    height: 400px;

    left: 350px;

    top: 0px;

    z-index: 99;

}

* **Asimétrica (Colocándola solo en el estado final):**

Si se aplica la transición *solo* al *estado final*, lo que ocurre es que la transición ocurrirá únicamente cuando va del estado inicial al final, pero no hay transición alguna devolviéndose.

figure {

    width: 100px;

    height: 100px;

}

figure:hover {

    width: 400px;

    height: 400px;

    left: 350px;

    top: 0px;

    z-index: 99;

    transition: left 2s ease-out, top 2s ease-out, width 3s ease, height 3s ease;

}

* **Asimétrica (Definida en ambos estados)**

Cuando se definen transiciones en ambos estados, lo que ocurre es que la transición definida en el estado final *(marcada en vinotinto)* ocurrirá cuando la transición va del estado inicial al estado final. Asimismo, la transición definida en el estado inicial (marcada en negro) ocurrirá cuando la transición va desde el estado final hacia el inicial:

figure {

    width: 100px;

    height: 100px;

    transition: all 0.2s;

}

figure:hover {

    width: 400px;

    height: 400px;

    left: 350px;

    top: 0px;

    z-index: 99;

    transition: left 2s ease-out, top 2s ease-out, width 3s ease, height 3s ease;

}

Las transiciones se aplican en la dirección del cambio de estado

# Capítulo 2. Trabajando con transformaciones de 2D

Las transformaciones son una propiedad que altera la apariencia de un elemento cambiando su tamaño, moviéndolo a otra localización en la página, rotándolo, etc. Cada transformación es aplicada usando la propiedad de *transform:*

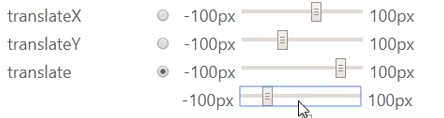


El efecto es el tipo de función de transformación y los parámetros son los valores asociados a esa función.

<https://developer.mozilla.org/es/docs/Web/CSS/transform>

* **Translate**:

Mueven el objeto, tanto como por el eje X o el eje Y o ambos:

* **Scale:**

Aumenta o disminuye el tamaño del objeto donde scaleX cambia el ancho del objeto y scaleY el alto

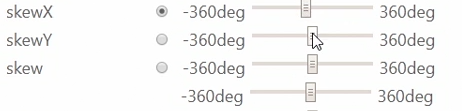




Con valores negativos se muestra la misma imagen pero al revés, es decir modo espejo.

* **Skew:**

Inclina la imagen





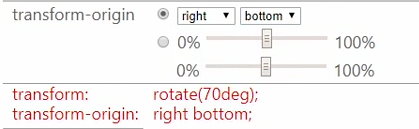
* **Rotate:**

Rota la imagen de acuerdo a los ángulos definidos:

Todas estas transformadas ocurren relativamente con el centro del objeto por default, es por eso que en rojo se resalta: ***transform-origin: center center;***

Sin embargo, es fácil cambiar el origen de la transformada. Por ejemplo, se puede mover el origen en la esquina inferior derecha para que de ahí empiece la rotación:



*Translations no son afectadas por el origen*

El orden en que se especifica efectos de la transformación afecta los resultados, es decir:

*translateX(10px) rotate(10deg)* ***diferente a*** *rotate(10deg) translateX(10px)*

Transformación de matrices

En CSS se puede usar la matriz de transformación de la siguiente manera:

