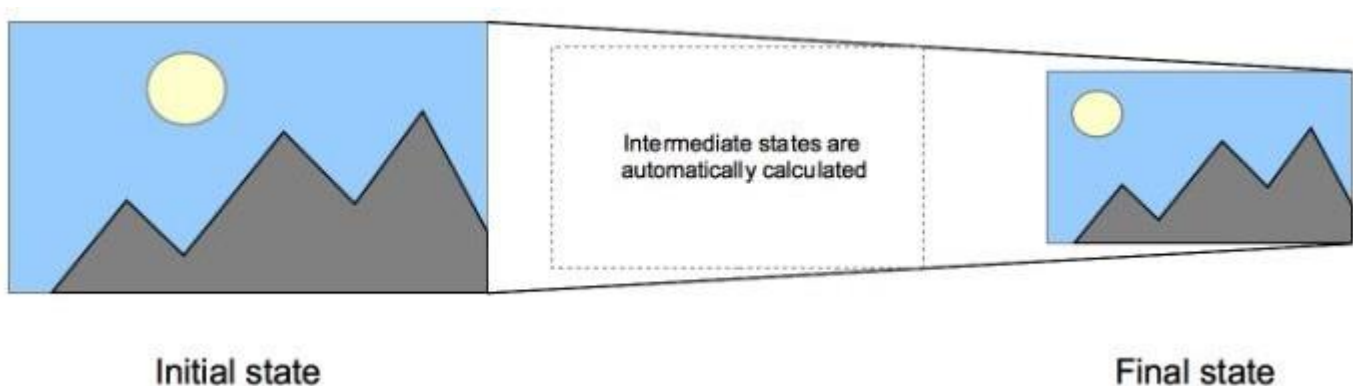


# Transiciones

Es un pasaje entre una valor de una propiedad a otro valor de manera gradual a través de una línea de tiempo. Las transiciones se activan con el **:hover** y el **:focus**.

Es decir que por ejemplo al pasarle el mouse sobre un elemento o hacer foco en un campo de texto es que activó **la transición**, veamos una imagen tomada de la página oficial de **Mozilla**



# transition-property & transition-duration

**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** también es 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.

```
<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; }

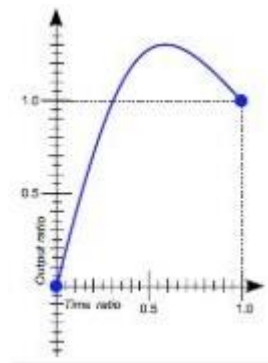
</style>
```

# transition-timing-function

Es cómo sucede la transición, los valores posibles son,

- **ease** ,valor predeterminado, inicia despacio, sucede rápido y termina despacio
- **linear**, mantiene la transición con estilo de velocidad igual en toda la duración de la misma
- **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

Con el valor **bezier**, podemos indicar de manera **personalizada en una curva** cuál va a ser la velocidad, como es medio complicado hacerlo , para eso tenemos aplicaciones por ejemplo la siguiente:<http://cubic-bezier.com/>



# 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>
```

# Transition Shorthand

Si quiero modificar más de una **propiedad** lo haré de la siguiente manera,

```
<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>
```

# Transition y Transform

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>
```

# Transition y Transform

Cuando rotamos **180deg** tanto de **x** , **y** o **z** un elemento hay que tener cuidado con la **superficie de apoyo** porque en cuanto el mouse deja de estar sobre el elemento este volverá hacia atrás en transición , las rotaciones tan abruptas es mejor **hacerla es la animaciones donde no tenemos ese problema.**

```
<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;
  backface-visibility: hidden; }

div:hover { background-color: red; width: 300px; transform: rotateX(185deg); }

</style>

<div> transiciones </div>
```

# Transition y Transform

En el ejemplo anterior, vemos que se usa la propiedad **backface-visibility** esta nos permite trabajar escondiendo o no la cara de atrás de un elemento cuando lo rotamos en **3D**, con sus valores **hidden** o **visible**.

También el ejemplo anterior vemos **all**, esto nos permite sin especificar todas las propiedades de la transición una por una , afectar a todas.

```
1 | backface-visibility: visible;  
2 | backface-visibility: hidden;
```



# Focus

Se puede trabajar con la transición en un **focus**, es decir por ejemplo un campo de texto que la hacerle foco se agrande, por ejemplo el que encontramos en dropbox 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>
```

# Focus

Focus es interesante para trabajar con varios elementos, pero especialmente con el visto anteriormente en el **código**, veamos como quedaría en el **navegador**,

El ejemplo anterior se verá así en tu navegador, hacele foco (que el cursor titila dentro) para probar que pasa:

Busqueda