

# Unidad 8: Animaciones

## Introducción

Vamos a explorar algunos conceptos clave en animaciones y estilos en CSS. Primero, hablaremos sobre los ejes en animaciones 3D y cómo se utilizan para mover objetos en el espacio. Luego, veremos cómo crear gradientes de fondo con CSS para mejorar el diseño visual de nuestras páginas web.

<https://vimeo.com/975639011>

Como has visto, comprender los ejes en animaciones 3D y utilizar gradientes en CSS puede llevar tus proyectos web a un nivel superior. Estas técnicas no solo mejoran la estética de tus diseños, sino que también aportan dinamismo y profundidad.

## Animaciones 3D en CSS

### Ejes en Animaciones 3D

En CSS, las animaciones 3D utilizan un sistema de coordenadas tridimensionales basado en tres ejes:

- **Eje X:** Controla el movimiento horizontal (de izquierda a derecha).
- **Eje Y:** Controla el movimiento vertical (de arriba hacia abajo).
- **Eje Z:** Controla la profundidad (hacia adentro y hacia afuera de la pantalla).

El eje Z es lo que diferencia las transformaciones 3D de las 2D, ya que permite dar la sensación de profundidad. Por ejemplo, al rotar un objeto en el eje Z, parecerá que está girando hacia o desde el usuario.

### Ejemplo práctico: Cubo en 3D

El siguiente ejemplo muestra cómo crear un cubo que rota continuamente en los ejes X e Y:

```
.cube {
```

```
width: 100px;

height: 100px;

position: relative;

transform-style: preserve-3d;

animation: rotateCube 5s infinite linear;

}

@keyframes rotateCube {

  from {

    transform: rotateX(0deg) rotateY(0deg);

  }

  to {

    transform: rotateX(360deg) rotateY(360deg);

  }

}
```

Este código crea un cubo animado que rota continuamente. El uso de `transform-style: preserve-3d` asegura que las transformaciones en 3D se rendericen correctamente.

## Gradientes en CSS

### Tipos de Gradientes

CSS permite crear gradientes que generan transiciones suaves entre dos o más colores. Existen tres tipos principales:

1. **Gradiente Lineal:** Los colores cambian a lo largo de una línea definida.

```
background: linear-gradient(90deg, red, blue);
```

2. **Gradiente Radial:** Los colores se expanden en círculos desde un punto central.

```
background: radial-gradient(circle, red, blue);
```

3. **Gradiente Cónico:** Los colores giran alrededor de un punto central como un reloj.

```
background: conic-gradient(from 0deg, red, blue);
```

## Ejemplo práctico: Gradiente Animado

Un fondo dinámico con gradientes animados puede crearse utilizando **@keyframes**:

```
body {  
  
    background: linear-gradient(90deg, red, blue);  
  
    background-size: 200% 200%;  
  
    animation: gradientAnimation 5s infinite alternate;  
  
}  
  
@keyframes gradientAnimation {  
  
    0% {  
  
        background-position: 0% 50%;  
  
    }  
  
    100% {  
  
        background-position: 100% 50%;  
  
    }  
  
}
```

Este gradiente se mueve continuamente, creando un efecto visual atractivo y dinámico.

## Transformaciones en CSS

Las transformaciones son una poderosa herramienta que permite modificar la forma en que se muestra un elemento sin cambiar su posición en el documento. En este video, aprenderemos cómo mover, escalar, rotar y sesgar elementos utilizando CSS.

<https://vimeo.com/975638847>

Hemos cubierto cómo usar `translate()`, `rotate()`, `scale()` y `skew()` para modificar la apariencia de los elementos en tu página web.

## Transformaciones en CSS

### Transformaciones Combinadas

CSS permite combinar transformaciones como `translate()`, `rotate()`, `scale()` y `skew()` para efectos avanzados. Por ejemplo:

- `translate()`: Mueve un elemento en el eje X o Y.
- `rotate()`: Gira un elemento alrededor de un punto fijo.
- `scale()`: Cambia el tamaño del elemento.
- `skew()`: Inclina el elemento en un ángulo.

### Ejemplo práctico: Tarjeta Animada

El siguiente ejemplo crea un efecto de tarjeta que se voltea al pasar el mouse:

```
.card {  
  
    width: 200px;  
  
    height: 300px;  
  
    perspective: 1000px;  
  
}  
  
.card-inner {  
  
    width: 100%;  
  
    height: 100%;  
  
    transform-style: preserve-3d;  
  
    transition: transform 0.6s;  
  
}  
  
.card:hover .card-inner {  
  
    transform: rotateY(180deg);  
  
}  
  
.card-front, .card-back {  
  
    width: 100%;
```

```
height: 100%;

position: absolute;

backface-visibility: hidden;
}

.card-back {

transform: rotateY(180deg);

}
```

Este ejemplo combina transformaciones en 3D con animaciones suaves para lograr un efecto interactivo y atractivo.

## Resumen

Estas explicaciones y ejemplos prácticos ofrecen una comprensión más profunda de las animaciones y estilos avanzados en CSS. Los estudiantes podrán implementar estas técnicas para mejorar tanto la funcionalidad como la estética de sus proyectos web.

# Transiciones en CSS

Las transiciones en CSS permiten cambiar gradualmente de un estilo a otro. Estas son útiles para crear efectos visuales suaves cuando los elementos cambian de estado, mejorando la experiencia del usuario en la web.

## Propiedades de las Transiciones

Para definir una transición en CSS, utilizamos varias propiedades clave:

1. **transition-property:** Especifica el nombre de la propiedad CSS que queremos animar.
2. **transition-duration:** Define el tiempo que tarda en completarse la transición.
3. **transition-timing-function:** Describe cómo progresará la transición a lo largo de su duración.
4. **transition-delay:** Indica el tiempo que se espera antes de que comience la transición.

## Ejemplo Básico

cssCopiar código

```
.button {
```

```
background-color: blue;

color: white;

padding: 10px 20px;

transition: background-color 0.5s ease;
}

.button:hover {

background-color: green;

}
```

En este ejemplo, cuando el usuario pasa el ratón sobre el botón, el color de fondo cambia de azul a verde en 0.5 segundos.

## Transición de Múltiples Propiedades

Podemos aplicar transiciones a múltiples propiedades especificándolas en una lista separada por comas.

### Ejemplo

cssCopiar código

```
.box {

width: 100px;

height: 100px;

background-color: red;

transition: width 0.5s ease, background-color 1s ease;

}

.box:hover {

width: 200px;

background-color: blue;

}
```

Aquí, tanto el ancho como el color de fondo de la caja cambiarán cuando se pase el ratón por encima, pero con diferentes duraciones.

## Funciones de Temporización

Las funciones de temporización (**transition-timing-function**) controlan la velocidad del efecto de transición. Las más comunes son:

- **ease**: La transición comienza lentamente, acelera en el medio y luego disminuye al final.
- **linear**: La transición tiene una velocidad constante de principio a fin.
- **ease-in**: La transición comienza lentamente y acelera hacia el final.
- **ease-out**: La transición comienza rápido y desacelera hacia el final.
- **ease-in-out**: Comienza y termina lentamente, con una aceleración en el medio.

## Ejemplo con Diferentes Funciones de Temporización

cssCopiar código

```
.ease-box {  
  
    transition: transform 1s ease;  
  
}  
  
.linear-box {  
  
    transition: transform 1s linear;  
  
}  
  
.ease-in-box {  
  
    transition: transform 1s ease-in;  
  
}  
  
.ease-out-box {  
  
    transition: transform 1s ease-out;  
  
}  
  
.ease-in-out-box {  
  
    transition: transform 1s ease-in-out;
```

```
}  
  
.box:hover {  
  
  transform: translateX(100px);  
  
}
```

En este ejemplo, cada caja tendrá un comportamiento diferente de acuerdo con la función de temporización utilizada.

## Delays en Transiciones

Podemos retrasar el inicio de una transición utilizando la propiedad **transition-delay**.

### Ejemplo con Delay

cssCopiar código

```
.delay-box {  
  
  transition: opacity 2s ease 1s;  
  
  opacity: 1;  
  
}  
  
.delay-box:hover {  
  
  opacity: 0;  
  
}
```

En este caso, la transición de opacidad comenzará 1 segundo después de pasar el ratón sobre la caja y durará 2 segundos.

## Conclusión

Las transiciones en CSS son una herramienta poderosa para mejorar la interactividad y la experiencia de usuario en un sitio web. Al dominar las propiedades de transición y las funciones de temporización, puedes crear efectos visuales atractivos y profesionales.

## Referencias

[Documentación oficial de CSS Transitions](#)



# Creación de Animaciones en CSS

Vamos a aprender cómo crear animaciones en CSS utilizando la regla `@keyframes`. Veremos cómo definir animaciones y aplicarlas a diferentes elementos para hacer que nuestro sitio web sea más dinámico e interactivo.

<https://vimeo.com/975638705>

Hemos visto cómo crear animaciones básicas en CSS usando `@keyframes` y cómo aplicarlas a diferentes elementos. Las animaciones pueden hacer que tu sitio web sea más interactivo y atractivo.