

# ANIMACIONES CON CSS

---

# ¿Por qué usar animaciones en una página web?

Las animaciones ayudan a mejorar la experiencia del usuario.

Sirven para:

- Hacer la interfaz más atractiva visualmente.
- Guiar la atención del usuario hacia elementos importantes.
- Retroalimentación, por ejemplo, cuando pasas el mouse sobre un elemento.
- Dar sensación de fluidez.

Un sitio sin animaciones puede parecer estático y poco moderno, incluso las animaciones mas sutiles hacen mas inmersiva la experiencia de navegación de una pagina

# ¿Qué es :hover y cómo se usa?

:hover es un pseudoselector que aplica o sobrescribe propiedades cuando el usuario pasa el mouse sobre el elemento seleccionado

Ejemplo práctico:

```
button:hover {  
  background-color: ☐ black; /* Cambia el color de fondo */  
  color: ☐ white; /* Cambia el color del texto */  
  transform: scale(1.1); /* Agranda el botón ligeramente */  
  transition: all 0.3s ease; /* Hace que los cambios ocurran suavemente */  
}
```

Este tipo de animación es ideal para botones y enlaces.


# Entendiendo transition

La propiedad transition nos permite controlar cómo cambian las propiedades CSS.

Sintaxis básica:

transition: propiedad // duración // tipo-aceleración;

Ejemplo:

```
div {  
  background-color:  blue;  
  transition: background-color 0.5s ease-in-out;  
}  
div:hover {  
  background-color:  red;  
}
```

Explicación:

- El color cambiará en 0.5 segundos.
- El cambio no será instantáneo, sino gradual.

# Propiedades que controlan una animación

Para personalizar una animación se usan estas propiedades:

- `animation-name`: nombre de la animación.
- `animation-duration`: duración en segundos.
- `animation-timing-function`: tipo de movimiento (`ease`, `linear`, `ease-in-out`).
- `animation-delay`: cuánto tiempo esperar antes de iniciar.
- `animation-iteration-count`: cuántas veces se repite.
- `animation-direction`: dirección de la animación (`normal`, `reverse`, `alternate`).

Ejemplo:

```
animation: mover 3s ease-in-out 1s infinite alternate;
```

# Animaciones más completas con @keyframes

@keyframes nos permite definir una animación personalizada paso a paso.

```
@keyframes mover {  
  0% { transform: translateX(0); }  
  100% { transform: translateX(100px); }  
}  
  
div {  
  animation: mover 2s infinite alternate;  
}
```

Explicación:

- translateX(100px): mueve el elemento hacia la derecha.
- 2s: la animación dura 2 segundos.
- infinite: se repite para siempre.
- alternate: primero hace una dirección normal y luego una reverse.

# Actividad de repaso

Objetivo: Crear una animación de un cuadrado con los siguientes efectos:

1. Que cambie de color de fondo cuando pasas el mouse sobre el elemento.
2. Que se desplace en diagonal infinitamente.
3. Que se agrande y luego vuelva a su tamaño original cuando pasas el cursor sobre el elemento.

# Ejemplo avanzado: botón que vibra al pasar el mouse

```
@keyframes vibrar {  
  0% { transform: rotate(0); }  
  25% { transform: rotate(2deg); }  
  50% { transform: rotate(-2deg); }  
  100% { transform: rotate(0); }  
}  
  
button:hover {  
  animation: vibrar 0.3s linear;  
}
```

Este efecto hace que el botón vibre rápidamente cuando pasas el cursor encima. Ideal para botones de "Enviar", "Eliminar", o "Confirmar".



# Buenas prácticas al usar animaciones

- Usa animaciones solo cuando tengan un propósito claro.
- No satures la interfaz con efectos innecesarios.
- Asegúrate de que no afecten el rendimiento en móviles.
- Prueba tu animación en varios navegadores.
- Usa transiciones rápidas (0.2s a 0.5s) para mantener la fluidez.