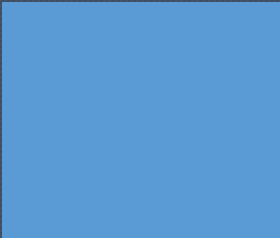


CSS | Transiciones y animaciones

Objetivos de la clase

- ✓ Crear transiciones usando el módulo de transiciones CSS3
- ✓ Explicar las diferencias entre transiciones y animaciones
- ✓ Comprender cómo crear animaciones usando bloques de animación
- ✓ Aplicar propiedades de animación



Agregar animación un su sitio web es una forma poderosa de atraer la atención de los usuarios.

Las animaciones pueden agregar interacción y agregar personalidad a su interfaz.

Transiciones CSS

Las transiciones CSS proporcionan una forma de controlar la velocidad de la animación al cambiar las propiedades CSS.

Nos permiten decidir qué propiedades animar, cuándo comenzará la animación (estableciendo un retraso), cuánto durará la transición (estableciendo una duración) y cómo se ejecutará la transición (definiendo una función de temporización, por ejemplo, lineal o rápida al principio, lenta al final).



Transiciones CSS

El módulo de Transiciones CSS3 introduce una serie de propiedades:

1. **transition-property:** Especifica la propiedad (o propiedades) de CSS para realizar la transición.
2. **transition-duration:** Especifica la duración de la transición.
3. **transition-timing-function:** Especifica la función de temporización de la transición.
4. **transition-delay:** Especifica un retraso opcional.

1. transition-property

Establece las propiedades CSS a las que se debe aplicar un efecto de transición.

Sintaxis:

```
/* Keyword values */  
transition-property: none;  
transition-property: all; → DEFAULT VALUE
```

```
/* <custom-ident> values */  
transition-property: margin-right;
```

```
/* Multiple values */  
transition-property: margin-right, color;
```


2. transition-duration

Establece el período de tiempo que una animación de transición debería tardar en completarse.

0s → default value (transición instantánea).

El único valor requerido para crear una transición es **transition-duration**.

Se pueden especificar múltiples duraciones; cada duración se aplicará a la propiedad correspondiente según lo especificado.

3. transition-timing-function

Permite establecer una curva de aceleración para que la velocidad de la transición pueda variar durante su duración.

La forma más fácil de implementarlo es especificando las keywords:

```
transition-timing-function: ease;  
transition-timing-function: ease-in;  
transition-timing-function: ease-out;  
transition-timing-function: ease-in-out;  
transition-timing-function: linear;
```


4. transition-delay

Especifica la duración a esperar (delay) antes de comenzar el efecto de transición.

El delay puede ser cero, positivo o negativo:

- Valor 0s: comenzará el efecto de transición inmediatamente. (DEFAULT VALUE)
- Valor positivo: retrasará el inicio de la transición durante un período de tiempo determinado.
- Valor negativo: comenzará el efecto de transición de inmediato, y a la mitad del efecto (comenzará a la mitad de su ciclo).

The transition shorthand property

En vez de escribir cuatro líneas de código para crear una transición, podemos simplificarlo mediante el uso de "transition shorthand property". La sintaxis es:

```
<transition> = <transition-property> <transition-duration> <transition-timing-function> <transition-delay>;
```




A diagram illustrating the components of a CSS transition rule. The text 'transition: margin-top 1s ease-in-out 0.5s;' is shown in a dark purple box. Handwritten labels in white, uppercase letters are connected to parts of the code by thin white lines. 'PROPERTY' points to 'margin-top', 'TIMING FUNCTION' points to 'ease-in-out', 'DELAY' points to '0.5s', and 'DURATION' points to '1s'. The word 'transition:' is in red, and the semicolon at the end is also in red.

transition: margin-top 1s ease-in-out 0.5s;

PROPERTY

TIMING FUNCTION

DELAY

DURATION

PRÁCTICA

Cambie el siguiente código para que coincida con la sintaxis "transition shorthand property":

```
div{  
  transition-duration: 2s;  
  transition-timing-function: linear;  
  transition-delay: 1s;  
  transition-property: width;  
}
```

<transition> = <transition-property> <transition-duration> <transition-timing-function> <transition-delay>;

RESULTADO

```
div{  
  transition: width 2s linear 1s;  
}
```



<transition> = <transition-property> <transition-duration> <transition-timing-function> <transition-delay>;

Resumiendo: Propiedades de transición

Propiedad	Descripción
transition	propiedad abreviada para establecer las cuatro propiedades de transición en una sola
transition-delay	Especifica un delay (en segundos) para el efecto de transición.
transition-duration	Especifica cuántos segundos o milisegundos tarda un efecto de transición en completarse.
transition-property	Especifica el nombre de la propiedad CSS para la que es el efecto de transición
transition-timing-function	Especifica la curva de velocidad del efecto de transición.

Transiciones múltiples

“The transition shorthand property” se puede usar para definir múltiples transiciones, usando una lista separada por comas. Por ejemplo:

```
div{  
  transition: width 2s ease-in 1s, height 4s, background-color 6s;  
}
```

<transition> = <transition-property> <transition-duration> <transition-timing-function> <transition-delay>;

Transform property

Permite modificar el espacio de coordenadas del modelo de formato visual CSS. Puedes trasladar, rotar, escalar los elementos. Por ejemplo:

`transform: none;` → DEFAULT VALUE

```
/* Some function values */  
transform: translate(12px, 50%);  
transform: translateX(2em);  
transform: translateY(3in);  
transform: scale(2, 0.5);  
transform: scaleX(2);  
transform: scaleY(0.5);  
transform: rotate(180deg);  
transform: skew(30deg, 20deg);
```


CSS Animations

Permiten animar la transición entre un estilo CSS y otro.

Constan de dos componentes: un estilo que describe la animación CSS y un conjunto de *keyframes* que indican su estado inicial y final, así como posibles puntos intermedios en la misma.

Ventajas de las animaciones CSS

- ✓ Son muy fáciles de usar.
- ✓ No se necesita JavaScript
- ✓ La animación se muestra siempre correctamente, a diferencia de JavaScript que no siempre se ve bien.
- ✓ El navegador contrala la secuencia de la animación, por lo que puede optimizar el rendimiento y eficiencia de la misma.

1. Animaciones: La regla @keyframes

Para usar la animación CSS, primero debes especificar algunos *keyframes*. Luego debes vincular la animación a un elemento.

```
/* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

Las palabras clave "from" y "to" (que representa 0% (inicio) y 100% (completo)).

Animaciones: La regla @keyframes

Además de "from" (0%) y "to" (100%), se pueden utilizar porcentajes. Por ejemplo:

```
/* The animation code */
@keyframes example {
  0%    {background-color: red;}
  25%   {background-color: yellow;}
  50%   {background-color: blue;}
  100%  {background-color: green;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

En este ejemplo cambiará el color de fondo y la posición del <div>, cuando la animación esté 25% completa, 50% completa, y nuevamente 100% completa

1. Animaciones: *animation-delay*

Especifica un delay para el inicio de una animación. Los valores negativos también están permitidos.

```
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
  animation-delay: -2s;  
}
```

2. Animaciones: *animation-iteration-count*

La propiedad especifica el número de veces que se debe ejecutar una animación. Su valor puede ser:

- Un número específico de iteraciones (1 → DEFAULT VALUE).
- **infinite**: repetir para siempre.
- **initial**: establece el valor al default value.
- **inherit**: hereda valor del padre.

```
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
  animation-iteration-count: infinite;  
}
```


PREGUNTA

¿Cuántas veces se va a ejecutar una animación si no se especificó la propiedad *animation-iteration-count*?

3. Animaciones: *animation-direction*

Especifica si una animación debe reproducirse hacia adelante, hacia atrás o en ciclos alternos. Su valor puede ser:

- **normal** - animación se reproduce normalmente → DEFAULT VALUE
- **reverse** - dirección inversa (hacia atrás)
- **alternate** - hacia adelante primero, luego hacia atrás
- **alternate-reverse** - primero hacia atrás y luego hacia adelante.

4. Animaciones: *animation-timing-function*

Especifica la curva de velocidad de la animación. Su valor puede ser:

- ease - inicio lento, luego rápido y finaliza lento. → DEFAULT VALUE
- linear – misma velocidad durante toda la animación
- ease-in - inicio lento
- ease-out – final lento
- ease-in-out – inicio y final lentos

```
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 4s;  
  animation-timing-function: linear;  
}
```

5. Animaciones: *animation-fill-mode*

Especifica un estilo para el elemento cuando la animación no se está reproduciendo (antes o después de que termine). Su valor puede ser:

- none → Default value.
- forwards
- backwards
- both

Animation Shorthand Property

Propiedad abreviada para configurar todas las propiedades de animación

```
div {  
  animation-name: example;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
  animation-delay: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```



```
div {  
  animation: example 5s linear 2s infinite alternate;  
}
```

Animation Shorthand Property

Para que la animación funcione correctamente, necesitamos seguir el **orden** adecuado y especificar al menos los **dos primeros valores**.

animation: [animation-name] [animation-duration] [animation-timing-function]
[animation-delay] [animation-iteration-count] [animation-direction]
[animation-fill-mode] [animation-play-state];

Animaciones múltiples

Para agregar múltiples animaciones a un selector, simplemente separe los valores con una coma.

```
.div {  
  animation: myslidein 2s, myrotate 1.75s;  
}
```

PREGUNTA

¿ Cuales son los valores obligatorios para que una animación funcione correctamente?

RESPUESTA

¿ Cuales son los valores obligatorios para que una animación funcione correctamente?

animation-name
animation-duration

Resumiendo: propiedades de animación CSS

Propiedad	Descripción
@keyframes	Especifica el código de animación.
animation	Propiedad abreviada para configurar todas las propiedades de animación.
animation-delay	Especifica un retraso para el inicio de una animación.
animation-direction	Especifica la dirección de la animación (hacia adelante, hacia atrás o en ciclos alternativos)
animation-duration	Especifica el tiempo de una animación completa.
animation-fill-mode	Especifica un estilo para el elemento cuando la animación no se está reproduciendo (antes o después de que termine)
animation-iteration-count	Especifica el número de veces que se debe reproducir una animación.
animation-name	Especifica el nombre de la animación @keyframes
animation-play-state	Especifica si la animación se está ejecutando o está en pausa
animation-timing-function	Especifica la curva de velocidad de la animación.

Transiciones vs. Animaciones

	Transiciones	Animaciones
Comienzan	Requiere un disparador para correr.	No necesita un disparador.
Puntos intermedios	Limitado a un estado inicial y final	Pueden incluir estados intermedios (keyframes) como inicial y final.
Iteración CSS	No pueden cambiar propiedades de CSS.	Pueden cambiar propiedades de CSS dentro de keyframes.
Looping	No están diseñados para looping.	Puede haber looping.

EJERCICIO

En el CodePen encontrará tres figuras geométricas. Tu misión es crear una animación diferente para cada una de ellas:

- El círculo: debes hacer que el círculo rebote para siempre.
- El cuadrado: con un delay de 5 segundos, haz que vuele hacia la derecha.
- El triángulo: haz que se mueva en zigzag hasta el final de la página y luego vuelva a su posición real.

Resumen

- ✓ Puedes crear increíbles transiciones y animaciones.
- ✓ Tus sitios web serán más interactivos y atractivos para los usuarios.
- ✓ Tus sitios web tendrán movimiento automático mediante el uso de complejas propiedades de animación y *keyframes*.