

UD8. ELEMENTOS MULTIMEDIA Y CONTENIDOS INTERACTIVOS EN LA WEB CON CSS

Prof: Alicia Fernández Catalán
CURSO 24-25 2º DAW - DIW



CONTENIDOS

- Las animaciones en CSS

ANIMACIONES EN CSS

- Las animaciones en CSS permiten **agregar efectos visuales dinámicos a los elementos de una página web.**
- A través de las animaciones, se puede hacer que los elementos cambien de manera fluida en propiedades como colores, tamaño, posición, opacidad, entre otros.
- CSS proporciona varias propiedades que nos permiten crear animaciones.
- A continuación, veremos las propiedades más comunes con ejemplos.

@KEYFRAMES

- Con la regla **@keyframes** se definen las etapas de la animación y sus correspondientes estilos a lo largo del tiempo.
- Es donde se especifica cómo deben cambiar las propiedades durante la animación.

CSS

```
@keyframes nombre-de-la-animacion {  
  0% { propiedad: valor; }  
  50% { propiedad: valor; }  
  100% { propiedad: valor; }  
}
```

- 0%: El inicio de la animación.
- 100%: El final de la animación.
- Puedes usar cualquier porcentaje entre 0% y 100% para definir pasos intermedios.

@KEYFRAMES - EJEMPLO

CSS

```
@keyframes ejemploAnimacion {  
  0% {  
    transform: scale(1);  
    opacity: 1;  
  }  
  50% {  
    transform: scale(1.5);  
    opacity: 0.5;  
  }  
  100% {  
    transform: scale(1);  
    opacity: 1;  
  }  
}
```

ANIMATION

- La propiedad **animation** se usa para aplicar las animaciones a un elemento.
 - Previamente hemos de definir un @KEYFRAME
- Esta propiedad es un atajo que agrupa varias propiedades de animación.

```
elemento {  
    animation: nombre-de-la-animacion duracion tipo-de-animacion retraso-de-repeticion de-timing-function;  
}
```

ANIMATION - EXPLICACIÓN PROPIEDADES

- **nombre-de-la-animacion:** El nombre de la animación, que hace referencia a un conjunto de reglas definidas con la propiedad @keyframes.
- **duracion:** El tiempo total que debe durar la animación. Se puede expresar en segundos (s) o milisegundos (ms).
- **tipo-de-animacion:** Define cuántas veces se debe repetir la animación. Los valores comunes son:
 - infinite: Repite la animación indefinidamente.
 - 1: La animación se ejecuta una sola vez.
- **retraso:** Tiempo de espera antes de que la animación comience a ejecutarse.
- **de-timing-function:** Controla el ritmo de la animación. Algunos valores comunes son:
 - linear: La animación tiene una velocidad constante.
 - ease: La animación comienza lentamente, luego acelera y luego se desacelera.
 - ease-in: La animación comienza lentamente y luego acelera.
 - ease-out: La animación comienza rápidamente y luego desacelera.

ANIMATION - EJEMPLO

```
CSS

div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation: ejemploAnimacion 2s ease-in-out 1s infinite;
}
```

Explicación del Ejemplo:

- **2s**: Duración de la animación (2 segundos).
- **ease-in-out**: El tipo de animación (comienza lentamente, luego acelera y vuelve a desacelerar al final).
- **1s**: El retardo antes de que comience la animación (1 segundo).
- **infinite**: La animación se repite infinitamente.

ANIMATION - OTRO EJEMPLO

CSS

```
@keyframes mover {  
  from {  
    transform: translateX(0);  
  }  
  to {  
    transform: translateX(100px);  
  }  
}
```

```
.elemento {  
  animation: mover 2s ease-in-out infinite;  
}
```

En este ejemplo:

- **@keyframes mover:** Define una animación llamada mover que mueve un elemento desde su posición original (0) hasta 100 píxeles a la derecha.
- **animation:** mover 2s ease-in-out infinite: Aplica la animación al elemento con una duración de 2 segundos, un ritmo de entrada y salida (ease-in-out) y la hace repetirse infinitamente.

ANIMATION - OTRO EJEMPLO

```
@keyframes color-fondo {  
  0% {  
    background-color: red;  
  }  
  50% {  
    background-color: blue;  
  }  
  100% {  
    background-color: green;  
  }  
}
```

```
.elemento {  
  animation: color-fondo 4s linear infinite;  
}
```

Este ejemplo hace que el fondo de un elemento cambie de color:

- En el 0% del tiempo (al inicio), el fondo es rojo.
- En el 50%, el fondo es azul.
- En el 100% (al final), el fondo es verde.

La animación dura 4 segundos y se repite infinitamente.

ANIMATION - PROPIEDADES ADICIONALES

- **animation-delay:** Define un retraso antes de que la animación comience.
 - Se puede usar en lugar de incluirlo directamente en la propiedad animation.
- **animation-direction:** Define si la animación debe reproducirse en reversa o alternar la dirección entre cada ciclo.
- Los valores comunes son:
 - normal: Reproduce la animación en la misma dirección.
 - reverse: Reproduce la animación en la dirección opuesta.
 - alternate: Alterna entre la dirección normal y reversa.

ANIMATION - EJEMPLO PROPIEDADES ADICIONALES

```
@keyframes ejemplo {  
  0% {  
    transform: rotate(0deg);  
  }  
  100% {  
    transform: rotate(360deg);  
  }  
}
```

```
.elemento {  
  animation: ejemplo 5s linear alternate;  
}
```

ANIMATION-NAME

- La propiedad **animation-name** se utiliza para definir el nombre de la animación que quieres aplicar a un elemento HTML.
- El nombre de la animación se refiere a un conjunto de reglas de animación definidas mediante @keyframes.
- Esto permite que se apliquen animaciones personalizadas a un elemento y se asocien a través de su nombre.

```
css

elemento {
  animation-name: nombre-de-la-animacion;
}
```

- La propiedad animation-name se usa para indicar qué animación se debe aplicar a un elemento. Esta animación debe estar previamente definida usando la regla @keyframes.
- animation-name no define cómo se realiza la animación (esto lo hace @keyframes), sino que simplemente vincula un conjunto de pasos de animación definidos con un nombre.

ANIMATION-NAME EJEMPLO

```
css
@keyframes mover {
  0% {
    transform: translateX(0);
  }
  100% {
    transform: translateX(100px);
  }
}

.elemento {
  animation-name: mover;
  animation-duration: 2s;
  animation-timing-function: ease-in-out;
}
```

En este ejemplo:

- **@keyframes mover:** Define una animación llamada mover que mueve un elemento de su posición original a 100 píxeles hacia la derecha.
- **animation-name: mover;** Aplica la animación llamada mover al elemento con la clase .elemento.
- **animation-duration: 2s;** Establece la duración de la animación a 2 segundos.
- **animation-timing-function: ease-in-out;** Define que la animación comenzará y terminará lentamente, acelerando en el medio.

EJEMPLO CON VARIAS ANIMACIONES

- Puedes aplicar varias animaciones a un elemento, utilizando animation-name para cada una.

```
CSS

@keyframes mover {
  0% { transform: translateX(0); }
  100% { transform: translateX(100px); }
}

@keyframes cambiar-color {
  0% { background-color: red; }
  100% { background-color: blue; }
}

.elemento {
  animation-name: mover, cambiar-color;
  animation-duration: 2s, 4s;
  animation-timing-function: ease-in-out, linear;
}
```

En este ejemplo:

- La animación mover desplaza el elemento a la derecha en 2 segundos.
- La animación cambiar-color cambia el color de fondo de rojo a azul en 4 segundos.
- Ambas animaciones se aplican simultáneamente al elemento.

USANDO ANIMATION-NAME CON OTRAS PROPIEDADES DE ANIMACIÓN

- La propiedad animation-name se suele usar junto con otras propiedades de animación, como animation-duration, animation-timing-function, animation-delay, entre otras.
- A continuación, un ejemplo más detallado que utiliza varias propiedades relacionadas con la animación.

```
css

@keyframes rotar {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}

.elemento {
  animation-name: rotar;
  animation-duration: 3s;
  animation-timing-function: linear;
  animation-iteration-count: infinite;
  animation-delay: 1s;
}
```

En este caso:

- @keyframes rotar: Define una animación llamada rotar que rota el elemento desde 0 grados hasta 360 grados.
- animation-name: rotar;; Aplica la animación llamada rotar.
- animation-duration: 3s;; Establece que la animación dure 3 segundos.
- animation-timing-function: linear;; La animación tiene una velocidad constante.
- animation-iteration-count: infinite;; La animación se repetirá infinitamente.
- animation-delay: 1s;; Hay un retraso de 1 segundo antes de que comience la animación.

ANIMATION-DURATION

- Especifica cuánto tiempo durará la animación.
 - Define cuánto tiempo debe tardar un elemento en completar un ciclo de animación.
- Esta propiedad es fundamental para controlar la velocidad de las animaciones, ya que establece el intervalo de tiempo que se tarda en ejecutar el conjunto de pasos definidos en una regla @keyframes.

CSS

```
elemento {  
  animation-duration: tiempo;  
}
```

- tiempo: El valor de duración de la animación. Puede expresarse en segundos (s) o milisegundos (ms).

ANIMATION-DURATION

- **Explicación**

- animation-duration especifica el tiempo total que durará un ciclo completo de la animación.
- Este valor puede ser un número seguido de una unidad de tiempo, como segundos (s) o milisegundos (ms).
 - 1s (1 segundo)
 - 500ms (500 milisegundos)
- Cuando se asigna un valor a animation-duration, el navegador aplica la animación a un elemento durante ese período de tiempo. Si la animación tiene varios pasos definidos en @keyframes, el tiempo total de duración se divide entre estos pasos.

EJEMPLO BÁSICO DE ANIMATION-DURATION

CSS

```
@keyframes mover {  
  0% {  
    transform: translateX(0);  
  }  
  100% {  
    transform: translateX(100px);  
  }  
}  
  
.elemento {  
  animation-name: mover;  
  animation-duration: 3s;  
}
```

En este ejemplo:

- @keyframes mover: Define una animación llamada mover que mueve el elemento desde su posición original (0px) a 100px a la derecha.
- animation-duration: 3s;: La animación durará 3 segundos desde el inicio hasta el final.

EJEMPLO BÁSICO DE ANIMATION-DURATION

CSS

```
@keyframes cambiar-color {  
  0% {  
    background-color: red;  
  }  
  100% {  
    background-color: blue;  
  }  
}  
  
.elemento {  
  animation-name: cambiar-color;  
  animation-duration: 1500ms;  
}
```

En este ejemplo:

Aquí, animation-duration: 1500ms; significa que la animación durará 1.5 segundos (1500 milisegundos) para cambiar el color de fondo de rojo a azul.

EJEMPLO CON VARIAS ANIMACIONES Y DURACIÓN

CSS

```
@keyframes mover {  
  0% { transform: translateX(0); }  
  100% { transform: translateX(200px); }  
}  
  
@keyframes rotar {  
  0% { transform: rotate(0deg); }  
  100% { transform: rotate(360deg); }  
}  
  
.elemento {  
  animation-name: mover, rotar;  
  animation-duration: 2s, 4s;  
}
```

En este ejemplo:

- La animación mover tiene una duración de 2 segundos.
- La animación rotar tiene una duración de 4 segundos.
- Ambas animaciones se ejecutan simultáneamente en el mismo elemento, pero cada una tiene su propio tiempo de duración.

PROPIEDADES RELACIONADAS CON ANIMATION-DURATION

- **animation-delay:** Permite establecer un retraso antes de que comience la animación.

```
CSS

.elemento {
  animation-duration: 2s;
  animation-delay: 1s; /* Retrasa la animación 1 segundo */
}
```

- **animation-timing-function:** Controla el ritmo de la animación, es decir, cómo cambia la velocidad de la animación a lo largo del tiempo.

```
CSS

.elemento {
  animation-duration: 3s;
  animation-timing-function: ease-in-out;
}
```


PROPIEDADES RELACIONADAS CON ANIMATION-DURATION

- **animation-iteration-count:** Define cuántas veces debe repetirse la animación. Puede ser un número entero o infinite para repetir indefinidamente.

```
CSS
```

```
.elemento {  
  animation-duration: 3s;  
  animation-iteration-count: infinite;  
}
```

ANIMATION-DIRECTION

- Controla la dirección de la animación en cada ciclo. Las opciones son:
 - normal: La animación sigue el flujo normal.
 - reverse: La animación se ejecuta en reversa.
 - alternate: La animación se alterna entre el ciclo normal y el inverso.
 - alternate-reverse: Comienza con la animación en reversa, y luego alterna entre los ciclos.

ANIMATION-FILL-MODE

- Especifica cómo se deben aplicar los estilos definidos por la animación antes de que comience o después de que termine.
 - none: No se aplican estilos antes o después de la animación.
 - forwards: Mantiene los estilos del último fotograma después de la animación.
 - backwards: Aplica los estilos del primer fotograma antes de la animación.
 - both: Aplica los estilos del primer y último fotograma.

```
css
div {
    animation-fill-mode: forwards; /* Los estilos
    finales de la animación se mantienen después de que
    termine */
}
```

EJEMPLO COMPLETO DE ANIMACIÓN

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de Animación CSS</title>
  <link type="text/css" rel="stylesheet" href="CSS/estilos.css">
</head>
<body>

<div></div>

</body>
</html>

```

```

1  /* Definir la animación */
2  @keyframes animacionCuadrado {
3      0% {
4          transform: scale(1) translateX(0);
5          background-color: red;
6      }
7      50% {
8          transform: scale(1.5) translateX(200px);
9          background-color: yellow;
10     }
11     100% {
12         transform: scale(1) translateX(0);
13         background-color: green;
14     }
15 }
16
17 /* Estilo del cuadrado */
18 div {
19     width: 100px;
20     height: 100px;
21     margin: 50px;
22     animation: animacionCuadrado 3s ease-in-out infinite;
23 }

```

CONCLUSIÓN

- Las animaciones en CSS son una herramienta poderosa para mejorar la interacción y la experiencia visual de una página web.
- **Usando @keyframes y las propiedades de animación**, podemos controlar cómo y cuándo los elementos cambian en la página, dando un toque dinámico a la interfaz.