Introducción a las animaciones con CSS 

Objetivos

Comprender los conceptos básicos de las animaciones con CSS.

Aprender a utilizar la propiedad animation y sus diferentes atributos.

¿Qué son las animaciones?

Las animaciones en CSS son una poderosa característica que permite crear efectos visuales y transiciones en elementos HTML sin necesidad de utilizar JavaScript o librerías externas. Con las animaciones CSS, puedes controlar y modificar gradualmente las propiedades de un elemento a lo largo del tiempo, lo que produce cambios suaves y atractivos en la interfaz del usuario.

Con CSS podemos dar vida y dinamismo a nuestros elementos HTML utilizando diferentes técnicas, entre ellas, se encuentran las transiciones, transformaciones y animaciones.

Transiciones

Las transiciones en CSS son una característica que permite cambiar suavemente los valores de una o varias propiedades de estilo de un elemento a medida que se produce un cambio de estado. En otras palabras, con las transiciones, puedes controlar cómo una propiedad de un elemento cambia de un valor a otro en respuesta a una acción del usuario o algún evento, como un "hover" (cuando el cursor pasa por encima del elemento).

La sintaxis básica para una transición:

```
elemento {
  propiedad: valor-inicial;
  transition: propiedad duración [función-de-tiempo] [retardo];
}
```

Donde:

elemento:
El elemento HTML al que se le aplicará la transición.

propiedad:
La propiedad CSS que deseas animar durante la transición, como width, height, background-color, etc.

valor-inicial:
El valor inicial de la propiedad antes de la transición.

duración:
La cantidad de tiempo que tomará completar la transición (por ejemplo, 1s para un segundo o 500ms para medio segundo).

función-de-tiempo:
Especifica la aceleración y desaceleración de la animación. Puedes utilizar valores como ease (predeterminado), linear, ease-in, ease-out, ease-in-out, etc.

retardo:
Un retardo antes de que comience la transición.

Ejemplo simple de una transición en CSS que cambia el color de fondo de un botón cuando se pasa el cursor sobre él:

```
Estilos del botón
.boton {
  padding: 10px 20px;
  background-color: blue;
  color: white;
  transition: background-color 0.3s ease;
}

Estilos del botón al pasar el cursor sobre él
.boton:hover {
  background-color: red;
}
```

Propiedad transition

Las propiedades de la transición en CSS son las que se utilizan para definir cómo se animarán los cambios de estilo en un elemento durante una transición. Estas propiedades te permiten controlar la duración, la función de tiempo (timing function), el retardo y otras características relacionadas con la animación.

Las principales propiedades de la transición son las siguientes:

transition-property:
Especifica qué propiedades CSS deben animarse.

Ej:

```
Transición para el color de fondo y el tamaño de fuente
transition-property: background-color, font-size;
```

transition-duration:
Define la duración de la transición en segundos o milisegundos.

Ej:

```
Transición que dura 1 segundo
transition-duration: 1s;

Transición que dura 500 milisegundos
transition-duration: 500ms;
```

transition-delay:
añade un retraso antes de que comience la transición. Puedes especificar el tiempo en segundos o milisegundos.

Ej:

```
Transición que comienza después de 500 milisegundos
transition-delay: 500ms;
```

transition-timing-function:
Es el tipo de aceleración o desaceleración de la transición. es decir establece la función de temporización que controla cómo se produce la transición.

Hay diferentes funciones de temporización predefinidas como por ejemplo linear, ease, ease-in, ease-out, ease-in-out.

linear:
La transición se produce de manera constante a lo largo de toda la duración. No hay aceleración ni desaceleración.

ease:
proporciona una aceleración suave al principio y una desaceleración suave al final de la transición.

ease-in:
La transición comienza lentamente y luego acelera hacia el final.

ease-out:
La transición comienza rápidamente y luego desacelera hacia el final.

ease-in-out:
La transición tiene una aceleración suave al principio, se mantiene constante en el medio y luego desacelera suavemente hacia el final.

Ej:

```
 Transición con una aceleración suave
transition-timing-function: ease;

Transición con una aceleración rápida al principio
transition-timing-function: ease-in;

todas las propiedades combinadas
Aplica una transición a la propiedad 'color' durante 0.5 segundos,
utilizando una aceleración suave al principio y un retraso
  de 200 milisegundos
transition: color 0.5s ease-in 200ms;
```

Propiedad transform

La propiedad CSS "transform" es una característica poderosa que te permite modificar la posición, rotación, escala y perspectiva de un elemento HTML. Con esta propiedad, puedes transformar la apariencia visual de un elemento sin afectar el flujo normal del documento ni cambiar el espacio ocupado por el elemento en el diseño.

La sintaxis básica de la propiedad "transform" es la siguiente:

```
selector {
  transform: valor;
}
```

Donde:

selector:
Es el selector CSS que representa el elemento al que se aplicará la transformación.

valor:
Especifica el tipo de transformación que deseas aplicar al elemento. Puede ser una función como translate(), rotate(), scale(), skew(), o una combinación de varias funciones.

Tipos de transformación

translate():
Mueve un elemento en la dirección y distancia especificada.

translateX():
Mueve un elemento horizontalmente en la distancia especificada.

translateY():
Mueve un elemento verticalmente en la distancia especificada.

Ejemplo de translate:

```
Desplazar el elemento 20px hacia la derecha y 10px hacia abajo
transform: translate(20px, 10px);
```

rotate():
Rota un elemento en el ángulo especificado.

Ejemplo de rotate:

```
Rotar el elemento 45 grados en sentido horario
transform: rotate(45deg);
```

scale():
Escala un elemento en la cantidad especificada.

scaleX():
Escala horizontalmente un elemento en la cantidad especificada.

scaleY():
Escala verticalmente un elemento en la cantidad especificada.

Ejemplo de scale:

```
Aumentar el tamaño del elemento al doble en ambos ejes
transform: scale(2);
```

skew():
Sesga un elemento en los ángulos especificados.

skewX():
Sesga horizontalmente un elemento en el ángulo especificado.

skewY():
Sesga verticalmente un elemento en el ángulo especificado.

Ejemplo de skew:

```
Inclinar el elemento 20 grados en sentido horizontal
transform: skewX(20deg);

Inclinar el elemento 30 grados en sentido vertical utilizando radianes
transform: skewY(0.5236rad);
```

Unidades de medida para skew

deg:
La unidad "deg" en CSS se usa para medir ángulos en grados. Es una abreviatura de "degrees" (grados en inglés) y es una de las unidades angulares más comúnmente utilizadas en CSS. La unidad "deg" se emplea en propiedades como transformaciones (rotate(), skew()).

rad:
La unidad "rad" en CSS se usa para medir ángulos en términos de radianes. Radian (rad) es una unidad angular que se utiliza comúnmente en matemáticas y trigonometría. En CSS, puedes usar la unidad "rad" en propiedades como transformaciones (rotate(), skew()) y en ciertas propiedades de gradiente (por ejemplo, linear-gradient).

Matrix

En CSS, la función matrix() es una de las funciones de transformación que se utiliza en la propiedad transform para aplicar transformaciones 2D personalizadas a un elemento.

La matriz de transformación 2D es una forma matemática de combinar varias transformaciones simples, como rotaciones, escalados, sesgados y traslaciones, en una sola matriz para aplicarlas todas a la vez al elemento. Esta matriz se especifica utilizando seis valores que representan los coeficientes de la matriz en el siguiente orden:

Ejemplo:

```
matrix(a, b, c, d, e, f);
```

Donde:

a y d son los factores de escala en los ejes X e Y, respectivamente.

b y c son los factores de sesgo (skew) en los ejes X e Y, respectivamente.

e y f son los valores de traslación (movimiento) en los ejes X e Y, respectivamente.

¿Que es animation en css?

En CSS, las animaciones (animation en inglés) son una característica que permite crear efectos de animación más complejos y sofisticados en elementos HTML. Con las animaciones CSS, puedes controlar y modificar gradualmente las propiedades de un elemento a lo largo del tiempo, lo que produce cambios suaves y atractivos en la interfaz del usuario.

La animación en CSS se basa en la combinación de dos componentes principales:

@keyframes:
Esta regla CSS define los pasos o estados intermedios de la animación. Permite especificar cómo se deben comportar las propiedades del elemento en diferentes momentos durante la animación.

animation:
Es una propiedad CSS que se aplica al elemento que se va a animar. Permite definir la duración, el retardo, la velocidad, la dirección y el nombre de los @keyframes que se aplicarán a la animación.

La sintaxis básica de una animación en CSS es la siguiente:

Ejemplo:

```
Definición de los @keyframes llamados 'nombre-de-la-animacion'
@keyframes nombre-de-la-animacion {
  Pasos intermedios de la animación
}
Pasos intermedios de la animación
selector {
  animation-name: nombre-de-la-animacion;
  animation-duration: duracion;
  Otras propiedades de animación (opcional)
}
```

Donde:

nombre-de-la-animacion:
Es un nombre que le das a la animación y que se utilizará para referenciar los @keyframes y la propiedad animation-name.

selector:
Es el selector CSS que representa el elemento al que se aplicará la animación.

duracion:
Especifica la cantidad de tiempo que tomará completar la animación.

Además de estas propiedades tenemos las siguientes:

animation-delay:
Establece un retraso antes de que la animación comience a ejecutarse. Puede ser especificado en segundos o milisegundos.

animation-iteration-count:
Define el número de veces que la animación se repetirá antes de detenerse. Se puede utilizar un número entero o el valor especial infinite para indicar que la animación se repetirá indefinidamente..

animation-timing-function:
Especifica la función de tiempo utilizada para controlar la aceleración o desaceleración de la animación a medida que avanza. Puedes utilizar funciones predefinidas como ease, linear, ease-in, ease-out, ease-in-out

animation-direction:
Determina la dirección de la animación. Puede ser normal (hacia adelante), reverse (hacia atrás), alternate (ida y vuelta), o alternate-reverse (ida y vuelta inversa).

animation-fill-mode:
se utiliza para definir cómo se aplican los estilos al elemento antes y después de que se ejecute una animación.

animation-play-state:
Controla el estado de reproducción de la animación. Puede ser running (en reproducción) o paused (pausada).

animation-fill-mode recibe los siguientes valores:

none:
Es el valor predeterminado. Indica que no se aplicarán estilos al elemento antes o después de la animación.

forwards:
Indica que el elemento mantendrá los estilos finales de la animación una vez que esta finalice. Después de la animación, el elemento se quedará con los estilos definidos en el último paso.

backwards:
Indica que el elemento aplicará los estilos iniciales de la animación antes de que esta comience, incluso si hay un retraso (animation-delay) antes de que la animación se ejecute.

both:
Combina los comportamientos de forwards y backwards. El elemento aplicará los estilos iniciales antes de que la animación comience y mantendrá los estilos finales después de que finalice.

@keyframes

Dentro de los @keyframes, se definen los diferentes estados o pasos intermedios de la animación, especificando cómo deberían comportarse las propiedades CSS del elemento en cada paso.

Ejemplo:

```
@keyframes slide-in {
  0% {
    transform: translateX(-100%);
  }
  50%{
    background-color: seagreen;
  }
  75%{
    transform: translate(300px, 300px) rotate(90deg) scale(2.2) skew( 20c
  }
  100% {
    transform: translateX(0);
  }

}
```