Animaciones CSS: Uso de keyframes

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Ene - Feb
- iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice

OBJETIVO

- Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.
- Utilizar marcos y tablas para presentar la información de manera ordenada.
- Identificar nuevos elementos y etiquetas en HTML5.
- Reconocer las posibilidades de modificar etiquetas HTML.
- Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.

GLOSARIO

Formularios. Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

Fuentes seguras. Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

Guías de estilo. Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

HTML. Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

GLOSARIO

HTML5. Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

Legibilidad. Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

Marcos. Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

Tipografía. Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa ”modo o estilo en que está impreso un texto” o ”clase de tipos de imprenta”.

INTRODUCCIÓN

Las **animaciones CSS** se crean mediante la propiedad `animation` y definiendo secuencias de fotogramas utilizando la regla `@keyframes`. Con la propiedad animation se pueden realizar animaciones simples, como un parpadeo o un desplazamiento suave, hasta animaciones más elaboradas, como la rotación de objetos o el cambio de color gradual. Estas animaciones no requieren interacción del usuario, ya que pueden ejecutarse automáticamente al cargar la página o repetirse indefinidamente.

DIFERENCIAS ENTRE LAS PROPIEDADES *animation, transition y transform*

En lecciones anteriores estudiamos cómo hacer transiciones y transformaciones en CSS3 sobre los elementos de nuestra página web. Veamos las diferencias entre esas dos propiedades y animation.

- **La propiedad** `transition` **en CSS** permite realizar cambios suaves entre dos estados de un elemento cuando ocurre un evento, como pasar el cursor sobre él (`:hover`). Es ideal para efectos básicos como cambios en el color, tamaño o posición, pero solo anima entre un estado inicial y final, sin pasos intermedios.
- **La propiedad** `transform` modifica instantáneamente la apariencia de un elemento, permitiendo rotarlo, escalarlo o moverlo en el espacio. Aunque `transform` no genera animaciones por sí mismo, puede combinarse con `transition` o `animation` para lograr efectos más elaborados.
- **La propiedad** `animation` permite definir múltiples estados intermedios en una animación y controlar aspectos como la duración, la repetición y la dirección. A diferencia de `transition`, las animaciones pueden ejecutarse automáticamente al cargar la página o repetirse indefinidamente sin requerir la interacción del usuario.

ANIMACIONES CSS CON PROPIEDAD *animation* (propiedad abreviada)

En este apartado veremos **cómo encadenar diferentes animaciones utilizando la propiedad animation y sus subpropiedades.** Para ello, **definiremos la propiedad animation sobre nuestros selectores y después definiremos nuestra secuencia de animación mediante @keyframes.**

Cuando trabajamos con animaciones en CSS, podemos optar por usar la propiedad abreviada `animation`, que incluye todas las configuraciones en una sola línea, o podemos utilizar las subpropiedades de `animation` de forma individual para tener un mayor control sobre cada aspecto de la animación.

ANIMACIONES CSS CON PROPIEDAD *animation*

Sintaxis propiedad *animation*

La propiedad `animation` engloba todas las subpropiedades relacionadas con la animación en CSS, por lo que sería la propiedad abreviada. Las subpropiedades permiten controlar aspectos específicos de la animación, como su nombre, duración, dirección, relleno, etc.

```
selector {
  animation: name duration timing-function delay iteration-count direction
fill-mode play-state;
}
```

- `name`: Especifica el nombre de la regla `@keyframes` que describe los fotogramas de la animación.
- `duration`: Indica la cantidad de tiempo que la animación consume en completar su ciclo. Se expresa en segundos (s) o milisegundos (ms).

ANIMACIONES CSS CON PROPIEDAD *animation*

Sintaxis propiedad *animation*

```
selector {
  animation: name duration timing-function delay iteration-count direction
fill-mode play-state;
}
```

- `timing-function`: Define el ritmo de la animación, es decir, cómo se muestran los fotogramas a lo largo del tiempo. Puede ser lineal, ease, ease-in, ease-out, ease-in-out, cubic-bezier, step-start, step-end, steps, etc.
- `delay`: Especifica un tiempo de espera antes de que la animación comience a ejecutarse. Se expresa en segundos (s) o milisegundos (ms).
- `iteration-count`: Indica el número de veces que se repetirá la animación. Puede ser un número entero, infinite para que se repita infinitamente, o un valor específico como 2, 3, etc.

ANIMACIONES CSS CON PROPIEDAD *animation*

Sintaxis propiedad *animation*

```
selector {
  animation: name duration timing-function delay iteration-count direction
fill-mode play-state;
}
```

- `direction`: Define si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia. Puede ser normal, reverse, alternate, o alternate-reverse.
- `fill-mode`: Especifica qué valores tendrán las propiedades después de finalizar la animación. Puede ser none, forwards, backwards, o both.
- `play-state`: Permite pausar y reanudar la secuencia de la animación. Puede ser running o paused.

ANIMACIONES CSS CON PROPIEDAD *animation*

Subpropiedades de *animation*

Si optamos por usar las subpropiedades, comenzamos añadiendo la propiedad `animation-name` al selector, a la cual le asignamos un nombre identificativo, denominado `identifier`. Luego, podemos utilizar las demás subpropiedades para ajustar diferentes aspectos de la animación, como la duración, dirección, entre otros.

Sintaxis:

```
selector{
        animation-name: identifier;
        subproperty: value;
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

Subpropiedades de *animation*

Las subpropiedades de `animation` son:

- `animation-name`: Especifica el nombre de la regla `@keyframes` que describe los fotogramas de la animación.
- `animation-delay`: Tiempo de retardo entre el momento en que el elemento se carga y el comienzo de la secuencia de la animación.
- `animation-direction`: Indica si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia.
- `animation-duration`: Indica la cantidad de tiempo que la animación consume en completar su ciclo (duración).
- `animation-iteration-count`: El número de veces que se repite. Podemos indicar `infinite` para repetir la animación indefinidamente.

ANIMACIONES CSS CON PROPIEDAD *animation*

Subpropiedades de *animation*

Las subpropiedades de `animation` son:

- `animation-play-state`: Permite pausar y reanudar la secuencia de la animación.
- `animation-timing-function`: Indica el ritmo de la animación, es decir, como se muestran los fotogramas de la animación, estableciendo curvas de aceleración.
- `animation-fill-mode`: Especifica qué valores tendrán las propiedades después de finalizar la animación (los de antes de ejecutarla, los del último fotograma de la animación o ambos).

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-name

Esta subpropiedad vincula el selector con una regla `@keyframes`. Los identificadores o nombres que utilizamos deben ser únicos.

```
.mi-elemento {
  animation-name: mi-animacion;
}

@keyframes mi-animacion {
  from { opacity: 0; }
  to { opacity: 1; }
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-delay

Determina un tiempo de espera antes de que la animación comience.

```
.mi-elemento {
  animation-delay: 500ms;
/* Espera medio segundo antes de comenzar */
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-direction

Indica si la animación debe alternar la dirección en cada ciclo.

Valores: `normal`, `reverse`, `alternate`, `alternate-reverse`.

```css
.mi-elemento {
  animation-direction: alternate;
  /* Va y viene */
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-duration

Establece cuánto tiempo tarda en completar un ciclo de la animación, se expresa en segundos (s) o milisegundos (ms).

```
.mi-elemento {
  animation-duration: 2s;
  /* Completa la animación en 2 segundos */
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-iteration-count

Define cuántas veces se repetirá la animación. `infinite` para una animación sin fin.

```
.mi-elemento {
  animation-iteration-count: infinite;
  /* Repite la animación indefinidamente */
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-play-state

Permite pausar y reanudar la reproducción de la animación.

Valores: `running`, `paused`.

```css
.mi-elemento:hover {
  animation-play-state: paused;
  /* Pausa la animación al pasar el mouse */
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-timing-function

Controla el ritmo de la animación, definiendo cómo se acelera y desacelera durante su ejecución.

Valores: `linear`, `ease`, `ease-in`, `ease-out`, `ease-in-out`, `cubic-bezier(n,n,n,n)`.

```
.mi-elemento {
  animation-timing-function: ease-in-out;
  /* Comienza lentamente, se acelera en el medio,
  y termina lentamente */
}
```

ANIMACIONES CSS CON PROPIEDAD *animation*

animation-fill-mode

Decide los estilos que se aplican a un elemento antes y después de su animación.

Valores: `none`, `forwards`, `backwards`, `both`.

```
.mi-elemento {
  animation-fill-mode: forwards;
  /* Mantiene los estilos del último keyframe después de finalizar */
}
```

ANIMACIONES CSS: USO DE *@keyframes*

Una vez añadida la propiedad animation a nuestro selector, haremos uso de los `@keyframes` para crear animaciones completas.

Los **@keyframes** son un conjunto de fotogramas clave que describen cómo se muestra cada elemento animado durante la secuencia de la animación. La sintaxis es la siguiente:

```
@keyframes identifier {
  from { ... }
  percentage { ... }
  to { ... }
}
```

ANIMACIONES CSS: USO DE *@keyframes*

```
@keyframes identifier {
  from { ... }
  percentage { ... }
  to { ... }
}
```

- `identifier`: Nombre que identifica la lista de keyframe. Debe coincidir con animation-name.
- `fromDesde`: (por ejemplo: desde 0%).
- `to`: Hasta (por ejemplo hasta 100%).
- `percentage`: Porcentaje intermedio de las veces que va a ocurrir una animación.

Ver descripción completa de propiedades

EJEMPLOS ANIMACIONES CSS CON *animation*

Ejercicio: rotación de forma indefinida

Crea una animación mediante `@keyframes` del logo de la *Junta de Andalucía* o del "Kiosko de Fernando", de forma que se encuentre rotando de forma indefinida. Para ello:

- Utiliza la propiedad `transform` dando varios valores a su rotación.
- Crea una animación con tres `keyframes`:
    - Uno inicial que tendrá rotación de 0 grados.
    - Otro en el 50% que tendrá una rotación sobre el eje *y* de 180 grados.
    - Uno final que volverá a tener una rotación de 0 grados.

EJEMPLOS ANIMACIONES CSS CON *animation*

Rotación de forma indefinida - Solución

Código HTML:

```
 <header>
   <nav>
      <a href="#"><img class="animacionLogo"
src="https://www.juntadeandalucia.es/sites/default/files/2023-02/norm
al_1.png" alt="logo Junta Andalucía" height="50"></a>
   </nav>
</header>
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Rotación de forma indefinida - Solución

Código CSS:

```css
.animacionLogo {
    animation-duration: 5s;
    animation-name: animacionLogo;
    animation-iteration-count: infinite;
  }
```

```css
@keyframes animacionLogo{
    from {
        -webkit-transform: rotate(0deg);
          transform: rotateY(0deg);
    }
   50%{
        -webkit-transform: rotate(180deg);
          transform: rotateY(180deg);
    }
    to {
        -webkit-transform: rotate(360deg);
          transform: rotateY(360deg);
        }
}
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Rotación de forma indefinida - Solución

Código CSS:

```css
header{
    position: fixed;
    width: 100%;
    height: 70px;
    background-color: #F7F9FA;
}
nav{
    max-width: 1024px;
    margin: 0 auto;
}
header img{
    padding: 18px 10px;
}
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/RNbqpPr

Junta de Andalucía

EJEMPLOS ANIMACIONES CSS CON *animation*

Ejercicio: deslizar un texto por el navegador

Crea una animación de un párrafo que se deslice por el navegador desde el borde derecho de la ventana.

- Genera el párrafo con Lorem Ipsum

EJEMPLOS ANIMACIONES CSS CON *animation*

Deslizar un texto por el navegador - Solución

Código HTML:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.</p>
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Deslizar un texto por el navegador - Solución

Código CSS:

```css
p {
  animation-duration: 4s;
  animation-name: animTexto;
  margin: 20px;
}
```

```css
@keyframes animTexto {
  from {
    margin-left: 100%;
    width: 100%;
  }
  to {
    margin-left: 0%;
    width: 100%;
  }
}
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Deslizar un texto por el navegador - Solución

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/azowJvx

EJEMPLOS ANIMACIONES CSS CON *animation*

Ejercicio: animación de una pelota botando

Crea una animación que simula una pelota botando.

- La `.bouncing-ball` comienza en su posición original.
- Se traslada 100 píxeles hacia arriba y luego vuelve a caer.
- La animación tiene una duración de 2 segundos y se repite infinitamente.

Nota: *Los `animation-timing-function` de `ease-in` y `ease-out` se aplican en los puntos clave para simular la aceleración y desaceleración de la pelota al rebotar.*

Puedes ajustar la altura del rebote (`translateY(-100px)`) y la duración de la animación para cambiar la apariencia del rebote. La forma redondeada de la pelota se logra con `border-radius: 50%`.

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de una pelota botando - Solución

Código HTML:

```
<div class="bouncing-ball"></div>
<hr>
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de una pelota botando - Solución

Código CSS:

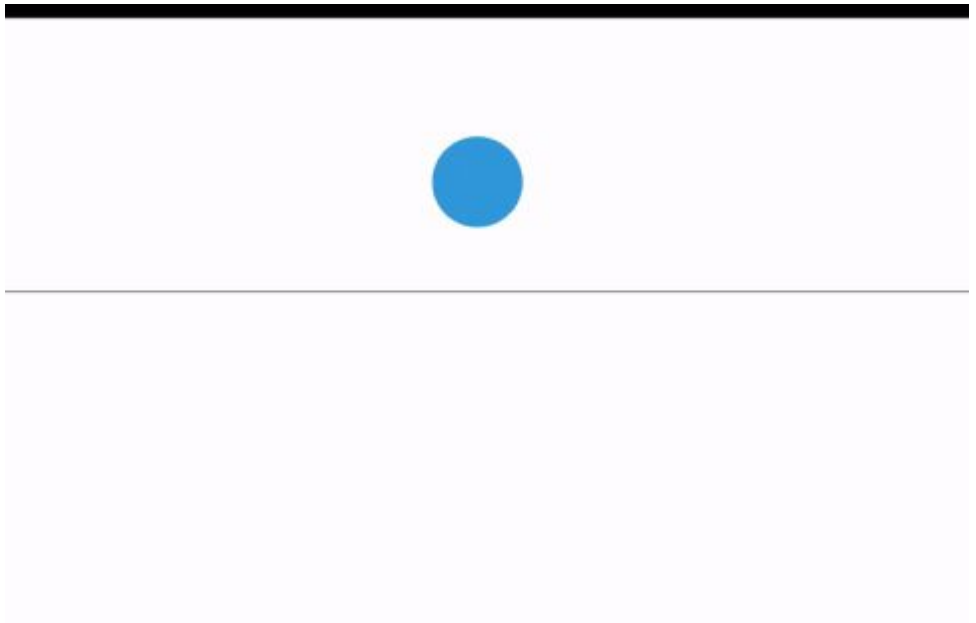
```css
.bouncing-ball {
  width: 50px;
  height: 50px;
  background-color: #3498db;
  border-radius: 50%;
  animation-name: bouncing;
  animation-duration: 2s;
  animation-timing-function:
ease-in-out;
  animation-iteration-count: infinite;
  margin: 100px auto;
}

hr{margin: -100px;}

@keyframes bouncing {
  0% {
    transform: translateY(0);
    animation-timing-function: ease-in;
  }
  50% {
    transform: translateY(-100px);
    animation-timing-function: ease-out;
  }
}
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de una pelota botando - Solución

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/JoPyorq



EJEMPLOS ANIMACIONES CSS CON *animation*

Ejercicio: animación de una rueda girando

Crea una animación que simule una rueda girando.

- La `.spinning-wheel` gira continuamente.
- La animación `spin` rota la rueda 360 grados indefinidamente.
- El tiempo de duración de la rotación es de 2 segundos y `linear` asegura que la velocidad de la animación sea constante.

Nota: *La rueda es representada por un `div` circular con un borde sólido y un `border-top-color` transparente para dar la apariencia de una rueda.*

Puedes ajustar el tamaño de la rueda, el grosor del borde y la duración de la animación para personalizar la apariencia y el comportamiento de la rueda giratoria.

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de una rueda girando - Solución

Código HTML:

```
<div class="spinning-wheel"></div>
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de una rueda girando - Solución

Código CSS:

```css
.spinning-wheel {
  width: 100px;
  height: 100px;
  border-radius: 50%;
  border: 10px solid #3498db;
  border-top-color: transparent;
  /*border-bottom-color: transparent;*/
  animation-name: spin;
  animation-duration: 2s;
  animation-timing-function: linear;
  animation-iteration-count: infinite;
}
```

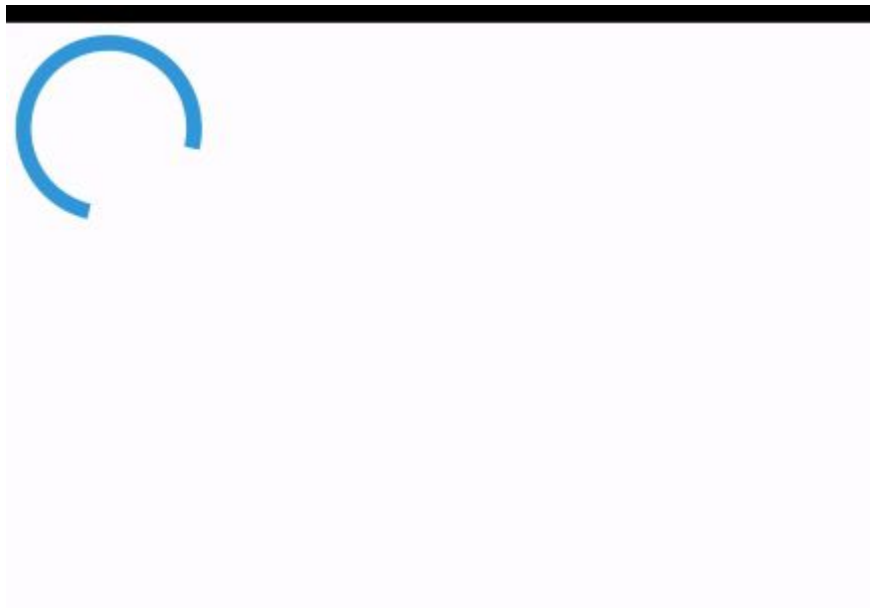
```css
@keyframes spin {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de una rueda girando - Solución

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/GgKvqGr



EJEMPLOS ANIMACIONES CSS CON *animation*

Ejercicio: animación de un indicador de carga

Crea una animación que simule un indicador de carga que aumenta de tamaño y cambia de color durante la animación.

- El tiempo de duración de la rotación es de 2 segundos, tiene un retraso de 0.5 segundos antes de iniciar y se repite infinitamente.
- Alterna la dirección en cada iteración y utiliza la función de tiempo `ease-in-out` para una transición suave.
- La propiedad `animation-fill-mode` con el valor `both` asegura que el círculo mantenga el estado final después de completar la animación.

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de un indicador de carga - Solución

Código HTML:

```
<div class="loading-indicator"></div>
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de un indicador de carga - Solución

Código CSS:

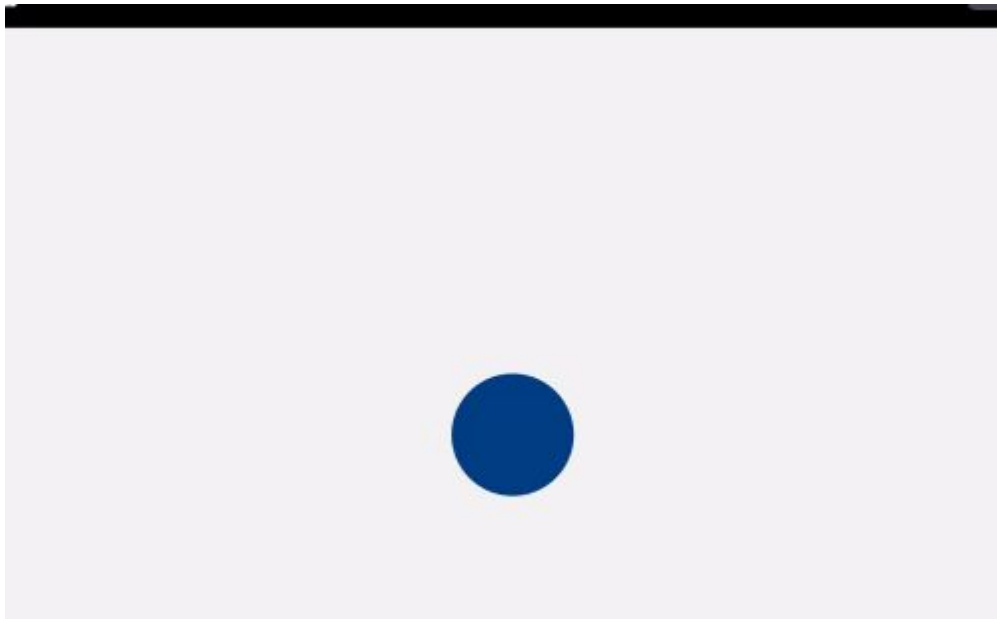
```css
body {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  margin: 0;
  background-color: #f5f5f5;
}
.loading-indicator {
  width: 50px;
  height: 50px;
  background-color: blue;
  border-radius: 50%;
  animation: scaleAndColorChange 2s
ease-in-out 0.5s infinite alternate both;}

@keyframes scaleAndColorChange {
  0% {
    transform: scale(1);
    background-color: blue;
  }
  100% {
    transform: scale(2);
    background-color: green;
  }
}
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de un indicador de carga - Solución

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/dPbzPQM



EJEMPLOS ANIMACIONES CSS CON *animation*

Ejercicio: animación de un latido de corazón

Crea una animación que simule *un latido del corazón* que puede ser útil para resaltar elementos interactivos como botones de donación, alertas importantes o cualquier elemento que desees destacar en tu página web.

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de un latido de corazón - Solución

Código HTML:

```
<div class="heart-beat">❤</div>
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de un latido de corazón - Solución

Código CSS:

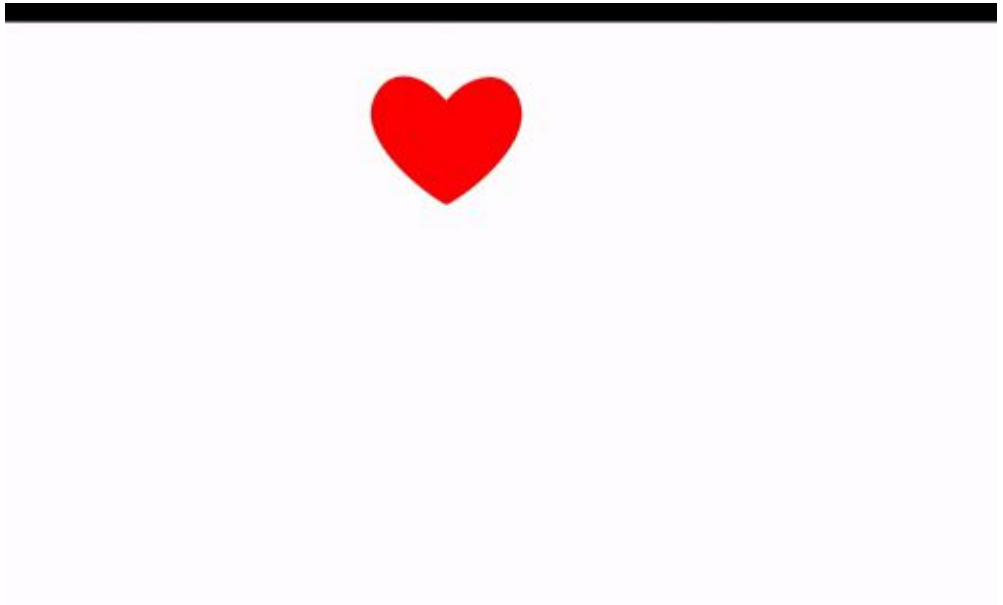
```css
.heart-beat {
  font-size: 80px;
  color: red;
  text-align: center;
  animation: heartbeat 3s infinite;
}

@keyframes heartbeat {
  0% {
    transform: scale(1);
  }
  25% {
    transform: scale(1.2);
  }
  50% {
    transform: scale(1);
  }
  75% {
    transform: scale(1.2);
  }
  100% {
    transform: scale(1);
  }
}
```

EJEMPLOS ANIMACIONES CSS CON *animation*

Animación de un latido de corazón - Solución

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/bNbrNZv



EJEMPLOS ANIMACIONES CSS

VER MÁS EJEMPLOS DE ANIMACIONES