

TEMA 7: ANIMACIONES

1. Introducción.

Uno de los aspectos más vistosos y atractivos de un sitio web es la inclusión de animaciones. Una animación (según Wikipedia) es un proceso utilizado para dar la sensación de movimiento a imágenes o dibujos. Para el caso concreto del diseño web, se puede completar esta definición añadiendo, además de imágenes, textos y audios que pueden visualizarse directamente desde un navegador, dando todo ello sensación de movimiento (animando la web). Las animaciones web se usan para muchos fines:

- Para mostrar un contenido (por ejemplo, un Sistema solar).
- Para publicidad, como banners.
- Detalles de diseño (por ejemplo, marcos o cenefas).
- Efectos.
- Objetos animados (por ejemplo, botones y opciones y menú).

Actualmente la web ofrece varias alternativas para la creación de estas animaciones.

2. Formatos de animaciones

Los formatos de vídeo vistos en la unidad anterior: mp4, ogv, webm, avi, etc... también se pueden emplear para introducir animaciones. Además, hay otros formatos más específicos que siguen la misma filosofía que el video, pero con otra finalidad.

GIF Animado.

Consiste, en una serie de imágenes, en formato gif que están colocados consecutivamente y se muestran en pantalla durante un intervalo de tiempo determinado. Esta secuencia se suele hacer en modo bucle (loop) para se repita indefinidamente.

- Ventajas:
 - Rápida descarga.
 - Nitidez.
 - Permite transparencia.
 - Se puede insertar en HTML como cualquier otra imagen.
- Desventaja:
 - Reducido número de colores (256)

Existen muchos sitios web que ofrecen gifs animados ya creados. Y también hay herramientas para crearlos y convertirlos a otros formatos (.avi, .mpeg, .swf, etc.). Algunas son las siguientes:

- GIF Construction Set Professional.
- GIMP.
- Picasion (Online).

Flash.

Archivos (.swf) (.fla) creados con Adobe Flash, ahora Adobe Animate (existen otras herramientas).

- Ventajas:
 - o Vectorial, ocupa poco espacio.
 - o Integra: imagen, texto, vídeo (Flash video) y audio.
 - o Incluye un lenguaje de programación (ActionScript).

En desuso. Dificultad para indexar estos archivos los motores de búsqueda. Apple no soporta esta tecnología en sus dispositivos. Solución: HTML5. Conversores de swf a HTML5.

Shockwave.

Era un formato muy popular para presentar contenidos animados más complejos que con Flash (juegos, entre otros). Ambos fueron creados por la misma compañía (Macromedia) luego bajo el nombre de Adobe. En 2019 Adobe deja de ofrecer shockware.

Canvas.

<canvas> es un elemento HTML el cual puede ser usado para dibujar gráficos usando scripts (normalmente JavaScript). Este puede, por ejemplo, ser usado para dibujar gráficos, realizar composición de fotos o simples (y no tan simples) animaciones.

<canvas> fue introducido primero por Apple para el Mac OS X Dashboard y después implementado en Safari y Google Chrome. Navegadores basados en Gecko 1.8, tal como Firefox 1.5, que también soportan este elemento.

3. Animaciones, transformaciones y transiciones CSS3

Con las nuevas propiedades CSS3 para animaciones, transformaciones y transiciones es posible crear muchos interesantes efectos sin necesidad de JavaScript, uso de frameworks o de la librería jQuery. La combinación de HTML5 con CSS3 nos permitirá crear sitios de la potencia de Flash, pero optimizados para SEO y con soporte nativo en la mayoría de los navegadores modernos.

Transformaciones

El efecto de una transformación en CSS3 consiste en aplicar un cambio de apariencia geométrica a cualquier elemento. Esto proporciona una notable capacidad de control visual que anteriormente no era posible en lo que respecta a HTML o a las hojas de estilo. Las dos propiedades que nos sirven para definir las transformaciones son `transform` y `transform-origin`.

- **transform-origin** La posición de origen que se utilizará para la transformación es por defecto el lado superior izquierdo del elemento.
- **transform** La posición de origen para realizar la transformación es el eje central del elemento.

Las transformaciones que podemos aplicar son las siguientes:

- **Scale**: modifica el tamaño de los elementos.
- **Translate**: cambia la posición del elemento. Izquierda, derecha, arriba o abajo.
- **Rotate**: gira o rota los elementos en grados.
- **Skew**: distorsiona los elementos.
- **Matrix**: mueve o transforma los elementos con precisión de píxel.

Esta propiedad requiere el uso de prefijos propietarios para garantizar la compatibilidad entre navegadores.

Prefijos CSS de los navegadores.

Actualmente los navegadores tienen implementadas muchas de las nuevas características de CSS3 utilizando sus propias versiones de cada propiedad mediante prefijos. Puede parecer que esto limita los beneficios de CSS3, pero los desarrolladores de navegadores lo hacen por lo siguiente: las primeras implementaciones de nuevas especificaciones CSS tienden a tener errores, así que, se proporcionan valores utilizando los prefijos propios de cada navegador, y también se proporciona una versión permanente de cada propiedad a través de una declaración sin prefijo. Después de un tiempo, cuando las especificaciones son estables, se eliminarán las propiedades con prefijo.

Los prefijos para los navegadores más comunes son los siguientes:

PREFIJO	NAVEGADOR
-moz-	Firefox
-webkit-	Safari y Chrome
-o-	Opera
-khtml-	Konqueror
-ms-	Internet Explorer
-chrome-	Google Chrome

Función	Descripción
rotate(angulo)	Especifica una rotación 2D en el ángulo especificado en el parámetro sobre el origen del elemento.
scale(sx,sy)	Especifica una operación de escala 2D con el vector de escala [sx,sy] descrita por los dos parámetros.
scaleX(sx)	Especifica una operación de escala con el vector de escala [sx,1], donde sx se indica como parámetro.
scaleY(sy)	Especifica una operación de escala con el vector de escala [1,sy], donde sy se indica como parámetro.
skew(angX,angY)	Especifica una transformación de deformación a lo largo de los ejes X e Y. El primer parámetro de ángulo especifica la deformación en el eje X. El segundo parámetro de ángulo especifica la deformación en el eje Y.
skewX(angulo)	Especifica una operación de deformación a lo largo del eje X según el ángulo indicado.
skewY(angulo)	Especifica una operación de deformación a lo largo del eje Y según el ángulo indicado.
translate(tx,ty)	Especifica una traslación 2D según el vector [tx,ty] descrito por los dos parámetros
translateX(tx)	Especifica una traslación de cantidad determinada en la dirección X.
translateY(ty)	Especifica una traslación de cantidad determinada en la dirección Y.

Se pueden realizar varias transformaciones sobre un elemento al mismo tiempo.

```
div:nth-child(3) {  
  transform: scale(1.25) rotate(30deg) translate(40px,0px);  
  -moz-transform: scale(1.25) rotate(30deg) translate(40px,0px);  
  -webkit-transform: scale(1.25) rotate(30deg) translate(40px,0px);  
  -o-transform: scale(1.25) rotate(30deg) translate(40px,0px);  
}
```

Transiciones en CSS3

En CSS también tenemos la posibilidad de animar los mismos efectos. Mediante la propiedad: transition. Las transiciones en CSS3 permiten modificar el valor de una propiedad de un elemento HTML en forma gradual durante un tiempo determinado de un estado inicial a un estado final. Sintaxis:

```
transition:[propiedad a modificar] [Duración] [Tipo de animación] [Retardo];
```

Se transiciona el color de fondo para que se realice de forma sutil y gradualmente:

```
.caja1{  
  background-color: #C0392B;  
  transition: background-color 1s linear;  
}  
.caja1:hover{  
  background-color: #3F51B5;  
}
```

Se pueden hacer transiciones de múltiples propiedades. También todas las propiedades de un elemento (“all”).

- **Propiedades para modificar:**

Algunas de las propiedades que podemos modificar utilizando transiciones son las que se muestran en la siguiente lista:

all	margin
background-color	opacity
border	word-spacing
border-radius	letter-spacing
color	fill
top	padding
bottom	stroke
left	text-shadow
right	vertical-align
box-shadow	visibility
width	z-index
height	
line-height	

- **Duración en segundos.**

Se debe especificar el número de segundos que va a durar la animación. Por ejemplo: 3s (3 segundos).

- **Tipo de animación:**

Esta propiedad es opcional y sirve para indicar la velocidad de la animación. Algunas de las posibilidades son las siguientes:

- linear: la velocidad de la animación es uniforme en todo el recorrido.
- ease: la velocidad se acelera al inicio, luego se retarda un poco y se acelera al final de nuevo.
- ease-in: la animación empieza lentamente y va aumentando progresivamente.
- ease-out: la animación empieza muy rápida y va descendiendo progresivamente.
- ease-in-out: la animación empieza y acaba lentamente, y es en la parte central del recorrido donde la velocidad es más rápida.

- **Retardo.**

Tiempo (en segundos) que el navegador esperará antes de poner en marcha la animación. Se especifica el número de segundos a esperar seguido de la “s” (ejemplo: 1s).



El cuarto parámetro opcional de la propiedad `transition` es indicar un tiempo de espera hasta que se inicie la transición:

```
transition:[propiedad a modificar] [Duración] [Tipo de animación] [Retardo];
```

```
Elemento {  
    transition: [nombre_propiedad] [duración_transición]  
               [función de transición] [tiempo de inicio];  
}
```

Sintaxis alternativa.

CSS3 dispone de otras cuatro funciones para indicar cada uno de los valores de las transiciones en forma independiente:

```
Elemento {  
    transition-property: [nombre de propiedad];  
    transition-duration: [duración de la transición];  
    transition-timing-function: [función de transición];  
    transition-delay: [tiempo de inicio];  
}
```

Animaciones en CSS3.

Las animaciones en CSS3 nos permiten hacer cosas que con las transiciones no alcanzamos y nos quedamos cortos. La sintaxis básica para una animación es la siguiente:

```
Elemento {  
    animation-name: [nombre de la animación];  
    animation-duration: [tiempo de duración];  
}  
@ keyframes [nombre de la animación] {  
    from {  
        [propiedades y valores del estado inicial de la animación]  
    }  
    to {  
        [propiedades y valores del estado final de la animación]  
    }  
}
```

Unidad 7. Contenido multimedia en la web.

Propiedades:

- **animation-iteration-count**

La propiedad animation-iteration-count permite controlar el número de repeticiones que tendrá la animación. (valor numérico o infinite)

```
Elemento {  
    animation-iteration-count: [cantidad de veces a repetir la animación o  
    "infinite"];  
}
```


- **animation-direction**

Indica como debe repetirse la animación. Esta propiedad puede asumir dos valores:

- normal: con lo que la animación cada vez que finaliza comienza desde el principio.
- alternate: cuando finaliza la animación comienza desde el final hasta el principio.

```
Elemento {  
    animation-direction: ["normal" o "alternate"];  
}
```

- **animation-timing-function y animation-delay**

Similar a las transiciones disponemos de dos propiedades para definir la función de transición y el tiempo que debe esperar para comenzar la animación:

```
Elemento {  
    animation-timing-function: [función de transición];  
    animation-delay: [tiempo de demora para iniciar la animación];  
}
```

Los valores posibles para la propiedad animation-timing-function son exactamente las mismas funciones disponibles para las transiciones, es decir: ease, linear, ease-in, ease-out y ease-in-out.

- Definición de más de 2 keyframes

Hasta ahora hemos indicado en la animación solo 2 keyframes (el inicial y el final), pero para animaciones más complejas es posible que necesitemos más de 2 keyframes, para esto tenemos la siguiente sintaxis:

```
@keyframes [nombre de la animación] {  
  0%{  
    [propiedades y valores]  
  }  
  [valor en porcentaje]%{  
    [propiedades y valores]  
  }  
  [valor en porcentaje]%{  
    [propiedades y valores]  
  }  
  100%{  
    [propiedades y valores]  
  }  
}
```

Animaciones con steps() y sprites

En vez de lograr una animación continua de inicio a fin con cada keyframe de la imagen, con Sprites podemos crearlo por pasos, es decir, steps().

```
.div {  
  animation: miAnimacion .8s steps(10) infinite;  
}  
  
@keyframes miAnimacion { 100% { background-position: -1900px; } }
```

Animaciones con steps() y sprites:

- 1- Imagen en Sprite.
- 2- Marco o visor.
- 3- Crear animación por pasos.

4. Animaciones mediante programación

Además de los formatos vistos en la sección anterior, existen otras maneras de incluir animaciones en un sitio web usando código de programación. Algunas de ellas:

- HTML y JavaScript
- HTML5, CSS3 y JavaScript

HTML y JavaScript.

Esta combinación permite crear animaciones fácilmente soportables por los navegadores de manera nativa. jQuery: es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

- Es software libre y de código abierto.
- Ahorra tiempo y espacio.

HTML5, CSS3 y JavaScript

Desde la aparición de los dispositivos móviles (concretamente iPad e iPhone) y su rápida extensión, Flash ha perdido mercado en las animaciones. HTML5 combinado con CSS3 y JavaScript permite crear animaciones tan potentes como Flash.

- Ventaja: soportado por los navegadores actuales de forma nativa.

El uso de HTML5, CSS3 y JavaScript tiene una estructura muy parecida a lo mostrado en el punto anterior. Algunas aplicaciones para estos desarrollos:

- Google Web Designer.
- Sencha Animator.
- Tumult Hype (para Mac).
- HTML5 Maker (Web. necesita Flash Player en el navegador).
-

5. Webs de consulta

6 webs para crear animaciones y vídeos de forma sencilla:

<http://hipertextual.com/2015/05/crear-animaciones-y-videos>

Tendencias en el diseño web para 2021:

<https://www.maxcf.es/tendencias-diseno-web-2021/>

Crear GIF animados:

<http://aulacm.com/programas-crear-hacer-gifs-animados/>

Transformaciones:

<https://lenguajecss.com/css/animaciones/transformaciones/>

Transiciones:

<https://lenguajecss.com/css/animaciones/transiciones/>

Animaciones:

<https://lenguajecss.com/css/animaciones/animaciones/>

Usando animaciones CSS:

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Animations/Using_CSS_animations

Herramienta con ejemplos elegantes y muy visuales de animaciones con código incluido:

<https://animista.net/play>

Animate.CSS:

<https://animate.style/>

Juegos con canvas:

<https://www.dacostabalboa.com/es/10-juegos-en-html5-y-canvas/8941>

Ejemplos animaciones HTML+CSS+JavaScript:

<https://www.ajaxshake.com/es/JS/12601/animacion.html>

jQuery:

<https://jquery.com/>

Para probar web con diferentes navegadores:

<https://www.browserling.com/>