

## Unidad 7

# Contenidos multimedia en la web: Animaciones

Módulo: Diseño de interfaces Web  
2º Curso Desarrollo de Aplicaciones Web  
Curso 2018-2019



XUNTA DE GALICIA

CONSELLERÍA DE EDUCACIÓN, UNIVERSIDADE  
E FORMACIÓN PROFESIONAL

CIFP A Carballreira-Marcos Valcárcel

Rúa A Carballreira s/n Ourense  
CP 32002 Ourense  
Tfno. 988788470



Centros  
Integrados  
Formación Profesional

# CONTENIDO DE LA UNIDAD

7.1 Introducción

7.2 Formatos de animaciones

7.3 Animaciones, transformaciones y transiciones en CSS3

7.4 Animaciones mediante programación

## 7.1 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).

# INTRODUCCIÓN

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 e menú)
- etc....

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

## 7.2 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 animados
- Flash / Shockwave
- HTML5 + CSS3 (+ JS)



# FORMATOS DE ANIMACIONES



## 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)

# FORMATOS DE ANIMACIONES



## GIF Animado

- 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**

<http://www.mindworkshop.com/gifcon.html>

- **GIMP**

<http://www.gimp.org.es/>

- **Picasion (Online)**

<http://picasion.com/es/>

# FORMATOS DE ANIMACIONES



## Flash

- Archivos (.swf) (.fla)
- Creados con Adobe Flash, ahora Adobe Animate (existen otras herramientas)
- Ventajas:
  - Vectorial, ocupa poco espacio.
  - Integra: imagen, texto, vídeo (Flash video) y audio.
  - 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.



## 7.3 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 navegadores modernos.

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transformaciones en CSS3

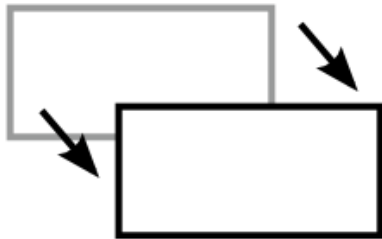
- El efecto de una transformación en CSS3 consiste en aplicar un cambio de apariencia geométrica a cualquier elemento.
- Esto proporciona una notable capacidad de control visual que anteriormente no era posible en lo que respecta a HTML o a las hojas de estilo.
- La propiedad CSS3 para realizar es: **transform**
- Esta propiedad requiere el uso de prefijos propietarios para garantizar la compatibilidad entre navegadores.

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

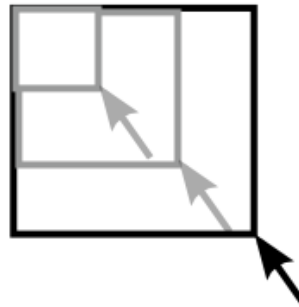
## Transformaciones en CSS3

- Permite hacer los siguientes tipos de transformaciones:
  - **Traslación**
  - **Escalado**
  - **Rotación**
  - **Distorsión o deformación**

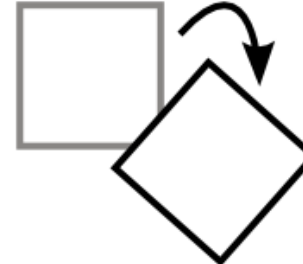
`translate()`



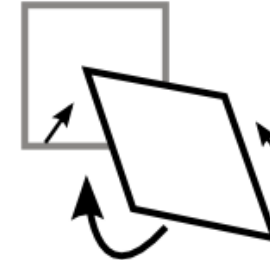
`scale()`



`rotate()`



`skew()`



# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transformaciones en CSS3. Traslación

- Las funciones de translación son aquellas que realizan una transformación en la que **mueven** un elemento de un lugar a otro.
- Si especificamos un valor positivo en el eje X (*horizontal*), lo moveremos hacia la derecha, y si especificamos un valor negativo, lo moveremos hacia la izquierda.
- Lo mismo con el eje Y (*vertical*)

Por ejemplo, `transform: translate(20px, -30px)` traslada el elemento 20 píxeles a la derecha y 30 píxeles hacia arriba, que es equivalente a utilizar `transform: translateX(20px) translateY(-30px)`.

Ejemplos: <https://css-tricks.com/almanac/properties/t/transform/>

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transformaciones en CSS3. Escalado

- Las funciones de escalado realizan una transformación en la que aumentan o reducen el tamaño de un elemento, basándose en el parámetro indicado, que no es más que un factor de escala.

En este ejemplo, `transform: scale(2, 2)` realiza una transformación de escalado del elemento, ampliándolo al doble de su tamaño original. Si utilizamos `scale()` con dos parámetros iguales, estamos manteniendo la proporción del elemento, pero si utilizamos diferentes valores, acabaría deformándose.

Ejemplos: <https://css-tricks.com/almanac/properties/t/transform/>

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transformaciones en CSS3. Rotación

- Las funciones de rotación simplemente giran el elemento el número de grados indicado.

Con `transform: rotate(5deg)` realizamos una rotación de 5 grados del elemento. Utilizando `rotateX()` y `rotateY()` podemos hacer lo mismo respecto al eje X o el eje Y respectivamente.

Ejemplos: <https://css-tricks.com/almanac/properties/t/transform/>

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transformaciones en CSS3. Distorsión o deformación

- Las funciones de deformación establecen un ángulo para torcer o inclinar un elemento en 2D.

Mediante `transform: skew(10deg, 5deg)` podemos hacer una deformación de varios grados, así como realizarla sólo respecto al eje X o eje Y con `skewX()` y `skewY()`.

Ejemplos: <https://css-tricks.com/almanac/properties/t/transform/>

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

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



# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

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

Ejemplo:

```
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);  
}
```

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## 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:

```
Elemento {  
    transition: [nombre_propiedad] [duración_transición] ;  
}
```

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transiciones en CSS3. Ejemplo

- Se transiciona la rotación de un div para que esta rotación se realice de forma sutil y gradualmente:

```
div {  
    display: inline-block;  
    padding: 1em;  
    margin: 1em;  
    background-color: #666;  
    color: #fff;  
    transition: transform 2s;  
}  
  
div:nth-child(1):hover {  
    background-color: #3F51B5;  
    transform: rotate(30.5deg);  
}
```

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

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transiciones en CSS3. Funciones de transición

- Un tercer parámetro opcional de la propiedad transition es indicar una "función de transición" que nos permite seleccionar la velocidad durante la transición:

```
Elemento {  
    transition: [nombre_propiedad] [duración_transición] [función_de_transición];  
}
```

- Posibles valores:
  - **ease** : Define un efecto de transición con un comienzo lento, luego rápido y termina lento (por defecto)
  - **linear** : Misma velocidad de inicio a fin.
  - **ease-in** : Define un efecto de transición con un comienzo lento.
  - **ease-out** : Define un efecto de transición con un final lento.
  - **ease-in-out** : Define un efecto de transición con un comienzo lento y un final lento.

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transiciones en CSS3. Funciones de transición



## Tiempo de demora en iniciar la transición

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

```

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

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Transiciones. 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];  
}
```

Ejemplo: <https://css-tricks.com/almanac/properties/t/transition/>

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Animaciones en CSS3

- Las animaciones en CSS3 nos permiten hacer cosas que con las transiciones no alcanzamos o 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]  
    }  
}
```

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Animaciones en CSS3. 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"];  
}
```



# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Animaciones en CSS3. Propiedades

### ■ **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.

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3

## Animaciones en CSS3.

- **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]  
  }  
}
```

Ejemplo:

<https://css-tricks.com/almanac/properties/a/animation/>

# ANIMACIONES, TRANSFORMACIONES Y TRANSICIONES CSS3. EJEMPLOS

- Animaciones CSS3 Avanzadas: <http://www.csslab.cl/2011/06/21/animaciones-css3-avanzadas/>
- Crear un Cubo 3D con CSS:  
<https://devcode.la/tutoriales/crear-un-cubo-3d-con-css/>
- Animación en CSS3: after, before y hover:  
<http://www.cristalab.com/tutoriales/animacion-en-css3-after-before-y-hover-c1131951/>
- Crear animaciones CSS3 fácilmente con la librería Animate.css:  
<https://devcode.la/tutoriales/crear-animaciones-css3-con-animatecss/>

## 7.4 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

# ANIMACIONES MEDIANTE PROGRAMACIÓN



## 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.
- Ejemplo:  
[http://www.marcofolio.net/webdesign/creating\\_a\\_polaroid\\_photo\\_viewer\\_with\\_css3\\_and\\_jquery.html](http://www.marcofolio.net/webdesign/creating_a_polaroid_photo_viewer_with_css3_and_jquery.html)

# ANIMACIONES MEDIANTE PROGRAMACIÓN

## 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.

# ANIMACIONES MEDIANTE PROGRAMACIÓN

## HTML5, CSS3 y JavaScript

- Algunas aplicaciones para estos desarrollos
  - Google Web Designer: <https://www.google.com/webdesigner/>
  - Sencha Animator:  
<https://www.sencha.com/>
  - Tumult Hype (para Mac):  
<http://tumult.com/hype/>
  - HTML5 Maker (Web. necesita Flash Player en el navegador):  
<http://html5maker.com/#/>

# DIRECCIONES WEB DE CONSULTA

- Usando animaciones CSS:  
[https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_Animations/Usando\\_animaciones\\_CSS](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Animations/Usando_animaciones_CSS)
- Ejemplos animaciones HTML+CSS+JavaScript:  
<http://www.ajaxshake.com/es/JS/12601/animacion.html>
- Adobe Animate: <https://helpx.adobe.com/es/support/animate.html>
- <http://www.tutosytips.com/?s=animaci%C3%B3n+css>
- <https://jquery.com/>
- Animate.CSS: <https://daneden.github.io/animate.css/>
- 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 2017: <https://www.40defiebre.com/tendencias-en-diseno-web-2017/>
- Crear GIF animados: <http://aulacm.com/programas-crear-hacer-gifs-animados/>
- [https://es.wikiversity.org/wiki/Introducci%C3%B3n, Formatos y Procesamiento de la Animaci%C3%B3n](https://es.wikiversity.org/wiki/Introducci%C3%B3n,_Formatos_y_Procesamiento_de_la_Animaci%C3%B3n)