

UT6 – Contenido multimedia e interactivo

Transformaciones,
transiciones y animaciones

UT6: Contenido multimedia e interactivo

- 0. Introducción
- 1. Derechos de autor y licencias
- 2. Imágenes
- 3. Audio y vídeo
- 4. **Animaciones**
 - 1. Formatos de archivo que admiten animaciones
 - 1. GIF
 - 2. Flash
 - 2. CSS3
 - 1. Transformaciones
 - 2. Transiciones
 - 3. Animaciones

4.1.1. GIFs animados

- Formato creado en 1987.
- Consiste, en una serie de **imágenes, en formato GIF** colocadas consecutivamente y se muestran en pantalla durante un **intervalo de tiempo** determinado.
- Esta secuencia se suele hacer en modo bucle (**loop**) para que se repita indefinidamente.

4.1.1. GIFs animados

Herramientas

■ **GIMP**, *The GNU Image Manipulation Program*, el Programa para manipular imágenes GNU (<http://www.gimp.org/>). Se distribuye libremente para tareas como retoque fotográfico, composición y creación de imágenes.

■ **Picasion** (<http://picasion.com/es/>) herramienta on line para crear gif animados a partir de un máximo de 10 imágenes. Su funcionalidad es muy limitada, pero permite poner la velocidad de transmisión y la calidad final. Además, las imágenes pueden ser cogidas de repositorios de fotos como Flickr.com.

■ **GifMake** (<http://gifmake.com/>): similar a picasion y también online.

■ **GIF Construction Set Professional** (<http://www.mindworkshop.com/gifcon.html>): Permite crear GIFs y conversión entre diferentes formatos.

4.1.2. Flash

■ Ficheros .swf.

- Contienen, principalmente:
 - Gráficos vectoriales (aunque también admite bitmaps).
 - Audio y vídeo.
- Permiten la interacción con el usuario.
- Suelen ser archivos suficientemente pequeños para ser publicados en la Web.

■ Herramientas

- Adobe Flash Professional.
- La **tendencia** es dejar de utilizar flash, y crear contenidos multimedia con HTML5.

4.2. CSS3

■ Nuevas propiedades CSS3

- ☐ **transform:** **Transformaciones**
- ☐ **transition:** **Transiciones**
- ☐ **animation:** **Animaciones**

■ Compatibilidad

- ☐ Algunos navegadores exigen utilizar prefijos, por lo que se suelen utilizar las propiedades con y sin prefijo:
 - -webkit- Chrome y Safari
 - -o- Opera
 - -moz- Firefox Mozilla
 - -ms- Internet Explorer
- ☐ Se puede consultar el nivel de soporte de cada propiedad en <http://caniuse.com>.

4.2.1. CSS3 - transformaciones

- La propiedad CSS **transform** se aplica a elementos, para **trasladar**, **rotar** y **escalar, etc.** en un espacio de dos o tres dimensiones.

Sintaxis

```
elemento {  
    transform: función(valor) función(valor)..  
}
```

Ejemplo

```
img{  
    transform: rotate(25deg);  
}
```

4.2.1. CSS3 - transformaciones

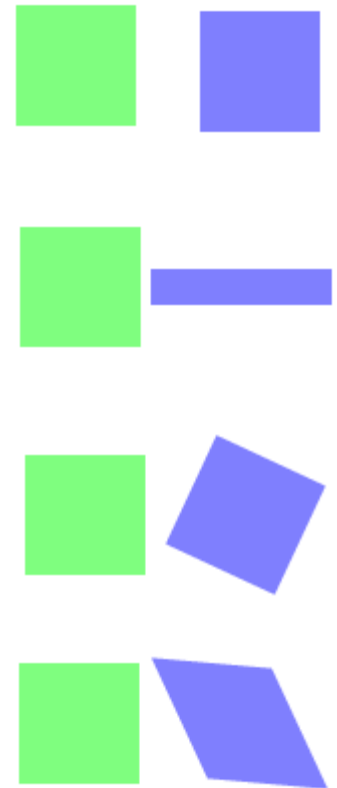
- Las transformaciones **no se heredan**.
- **No afectan al flujo del documento**. Por ejemplo, si un elemento se escala, para el resto no existe tal transformación, se comportan como si no hubiera cambiado su tamaño.
- Se pueden aplicar tanto a **elementos de bloque** como los **de línea**.
- Se pueden aplicar **varias funciones** simultáneamente.



4.2.1. CSS3 - transformaciones

Funciones 2D

- **matrix**(num [, num] {5,5})
- **translate**(*desplazamientoX* [, *desplazamientoY*]?);
 - transform: translate(12px, 19px);
- **translateX**(*desplazamiento*)
- **translateY**(*desplazamiento*)
- **scale**(*porcentajeX* [, *porcentajeY*]?);
 - transform: scale(1.5, 0.6);
- **scaleX**(*porcentaje*);
- **scaleY**(*porcentaje*);
- **rotate**(*grados*);
 - transform: rotate(25deg);
- **transform-origin**: *pos_x pos_y*;
 - transform-origin: top left;
- **skew**(*gradosX*, *gradosY*);
 - transform: skew(15deg, 3deg);
- **skewX**(*grados*);
- **skewY**(*grados*);



4.2.2. CSS3 – transiciones

- Normalmente, cuando cambia una propiedad CSS, el resultado se ve inmediatamente. Por ejemplo, si el fondo de un botón pasa de rojo a azul, el cambio se realiza de forma instantánea.
- Las transiciones permiten **controlar cómo se realiza el cambio de algunas propiedades de un elemento**. Estas transiciones se aplican a cambios realizados por cualquier medio: CSS, JavaScript...
- Aplicando transiciones, las propiedades tomarán una serie de **valores intermedios** entre el valor antiguo y el nuevo valor.
- Sólo se puede aplicar transiciones a las **propiedades CSS animables** (ver documentación HTML5-transiciones): en general, las que se pueden definir con un número: color, width, left, margin, font-size, z-index, ...

4.2.2. CSS3 – transiciones

- Las **transiciones** se utilizan para indicar cómo se tiene que realizar un cambio en el valor de una propiedad CSS.
Las transiciones **se disparan de forma "implícita"** cuando cambian las propiedades.
Lo único que **se puede definir** es el tiempo, el retraso, y una función del tiempo.
- Existen varias propiedades CSS para definir las transiciones sobre un objeto; también existe la función shorthand.
 - **transition-property**: propiedad para la que se define la transición.
 - **transition-duration**: duración en segundos de la transición.
 - **transition-timing-function**: función matemática aplicada a la transición.
 - **transition-delay**: retardo en empezar.
 - **transition**: propiedad shorthand.

4.2.2. CSS3 – transiciones

■ **transition-property**: none | <propiedad> [, <propiedad>]

Donde <propiedad> puede ser:

- Propiedad para la que se va a definir la transición.
- all: palabra reservada que indica todas las propiedades animables.
- El valor por defecto es **all**.

■ **transition-duration**: <tiempo> [, <tiempo>]

- Tiempo que dura la transición.
- El tiempo se define con un número y la unidad de tiempo (ej. 1s).
- El valor por defecto es **0s**.

Ejemplo:

```
transition-property: background-color;  
transition-duration: 2s;  
transition-timing-function: linear;  
transition-delay: 0.5s;
```

4.2.2. CSS3 – transiciones

■ **transition-timing-function**: <f_tiempo> [, <f_tiempo>]

Donde <f_tiempo>:

- Mide el grado de cambio de la propiedad en función del tiempo.
- El valor por defecto es **ease**.

https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_transition-timing-function

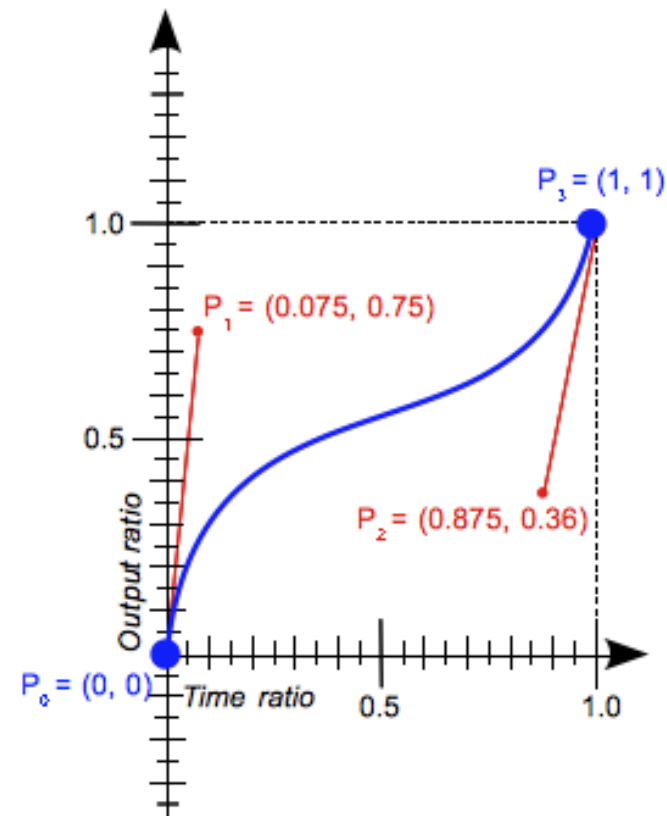
■ **transition-delay**: <tiempo> [, <tiempo>]

- Tiempo que tarda en empezar la transición.
- El tiempo se define con un número y la unidad de tiempo (ej. 1s)
- El valor por defecto es **0s**.

4.2.2. CSS3 – transiciones

Funciones del tiempo:

- Funciones basadas en una curva de Bézier:
 - **ease**: empieza despacio, sigue a velocidad uniforme y termina despacio.
 - **ease-in**: empieza despacio, después sigue a velocidad uniforme.
 - **ease-out**: al principio va a velocidad uniforme, termina despacio.
 - **ease-in-out**: es como ease, pero más lento al principio y al final.
 - **linear**: cambia a velocidad constante.
 - **cubic-bezier(<x1>, <y1>, <x2>, <y2>)** : permite definir exactamente cómo tiene que ser el cambio. Se dan las coordenadas de P1 y P2 en un rango de 0 a 1.
- Funciones por pasos.
- El valor por defecto es **ease**.

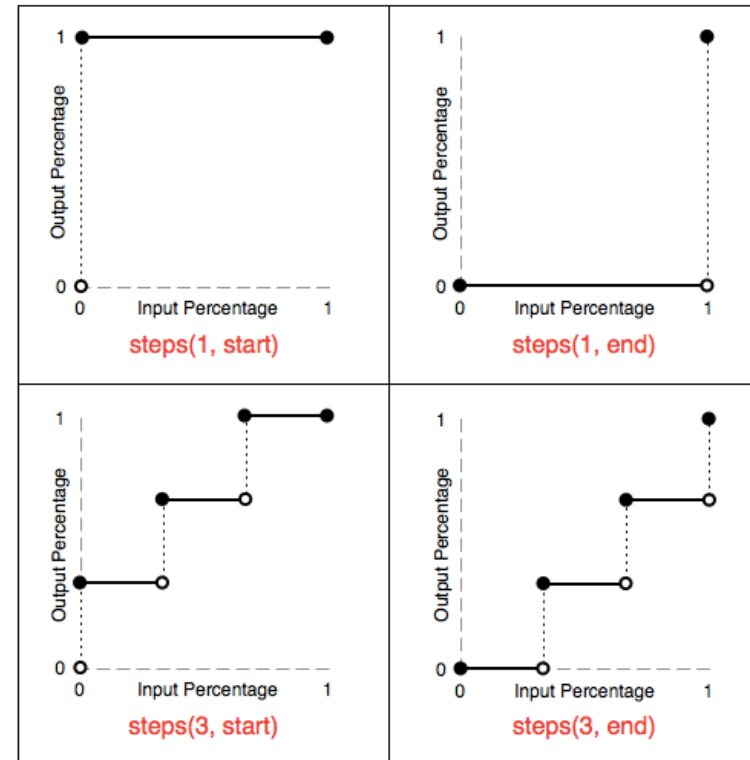


4.2.2. CSS3 – transiciones

Funciones del tiempo:

■ Funciones por pasos:

- `steps(<num>[, [start | end]]?)`: cambia al principio o al final de cada paso, dependiendo del segundo parámetro, si existe, en <num> pasos.
- `step-start`: cambia al principio, en un paso.
- `step-end`: cambia al final, en un paso.



4.2.2. CSS3 – transiciones

Aplicar transiciones a varias propiedades

- En cada propiedad transition- hay una lista de valores separados por comas
- A la primera transition-property se le aplican la función, tiempo y retardo definidos en la primera posición de las otras propiedades; a la segunda, las que están en la segunda posición, etc.
- **Si no hubiera bastantes propiedades**, funciona como un bucle: toma la primera posición y empieza desde ahí.

Ejemplo: Los cambios tienen diferente duración: para el background-color, 2s; para el border-color, 1s; para el color 2s.

```
transition-property: background-color, border-color, color;  
transition-duration: 2s, 1s;
```


4.2.2. CSS3 – transiciones

Propiedad shorthand : transition

■ **transition**: <transicion-simple> [, <transicion-simple>]

■ **transicion-simple**: [none | <propiedad>] || <tiempo-duración> || <f-tiempo> || <tiempo-retraso>

*El orden en que aparecen estos elementos es importante: el primer valor de tiempo será la duración, y el segundo el retraso.

Ejemplo:

```
transition: background-color 2s, border-color 1s;
```

4.2.2. CSS3 – transiciones

Definir diferentes valores para la transición “hacia adelante” y “hacia atrás”

- Cuando cambia el valor de una propiedad “animable”, el navegador consulta el valor de las propiedades “transition-” para decidir cómo se realizará la transición.
- Si el valor de alguna propiedad “transition-” ha cambiado, se tomará su nuevo valor antes de realizar la transición.
- Esto permite definir transiciones “hacia adelante” y “hacia atrás”

Ejemplo

```
li {  
    transition: background-color linear 1s; background: blue; }  
  
li:hover {  
    background-color: green;  
    transition-duration: 2s; /* Se aplica al cambiar al estado :hover*/ }
```

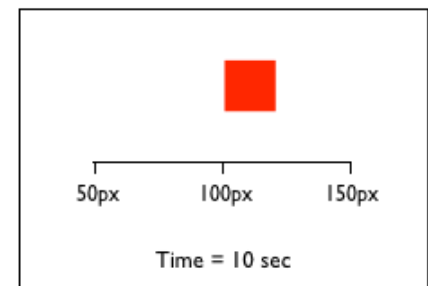
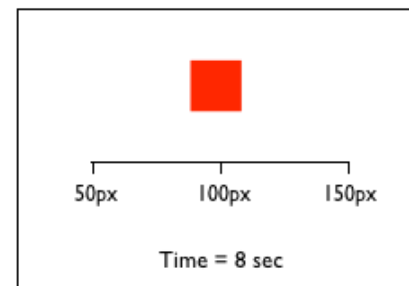
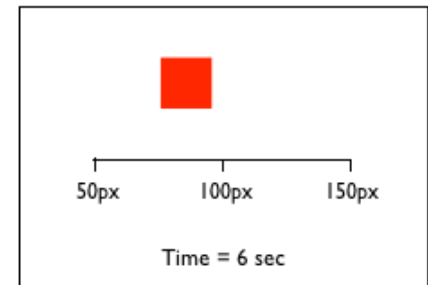
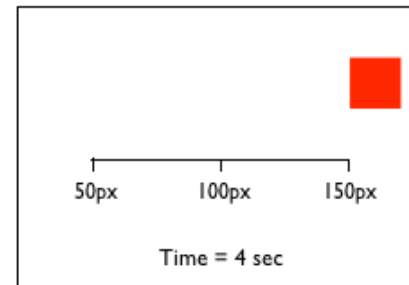
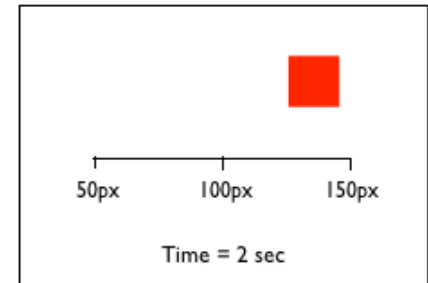
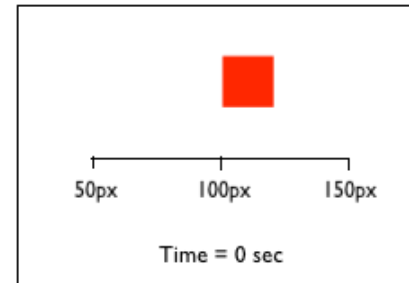
4.2.3. CSS3 – animaciones

- Las **animaciones** permiten especificar los cambios en propiedades CSS a lo largo del tiempo a través de un conjunto de keyframes.
- Las animaciones **se disparan de forma "explícita"** cuando se aplican las propiedades "animation". Por ejemplo, al cargar la página o con JavaScript.
- En las animaciones **se puede especificar**, como en las transiciones:
 - ☐ **Duración**
 - ☐ **Tiempo de retardo**
 - ☐ **Función del tiempo**Además, se pueden especificar otras características de la animación, como:
 - ☐ **Número de veces que se ejecuta la animación**
 - ☐ **Dirección**
- En las animaciones, se definen las propiedades que se pueden animar en los @keyframes:
 - ☐ **Pasos intermedios**Al igual que en las transiciones, sólo se pueden animar algunas propiedades CSS.

4.2.3. CSS3 – animaciones

Ejemplo

```
@keyframes wobble {  
    0% {    left: 100px;  
    }  
    40% {    left: 150px;  
    }  
    60% {    left: 75px;  
    }  
    100% {   left: 100px; }  
}  
div {  
    width: 50px;  
    height: 50px;  
    background-color: red;  
    position: relative;  
    left: 100px;  
  
    animation-name: wobble;  
    animation-duration: 10s;  
    animation-timing-function:  
        linear;  
}
```



4.2.3. CSS3 – animaciones

Regla **@keyframes**

- Se usa para especificar los valores de las propiedades a animar en varios puntos durante la animación. Los *keyframes* especifican el comportamiento de un ciclo de animación (aunque la animación se puede ejecutar varias veces).
- Las *keyframes* se especifican con una regla arroba: **@keyframes** seguida del nombre de la animación, seguida de un conjunto de reglas de estilo entre llaves, que definen los valores de las propiedades que van a ser animadas.

Sintaxis

```
@keyframes nombre_animacion {  
    selector {  
        declaraciones css  
    }  
    ...  
}
```

Ejemplo

```
@keyframes mover {  
    from {  
        left: 0%;  
    }  
    to {  
        left: 100%;  
    }  
}
```

4.2.3. CSS3 – animaciones

Regla @keyframes

Los **selectores**, son una lista de valores de porcentaje separados por comas, o las palabras clave "**from**" o "**to**". "**from**" equivale a "**0%**" y "**to**" equivale a "**100%**".

- Si no se especifica "**from**" o "**0%**", el navegador construye un keyframe '0%' utilizando el valor computado de las propiedades CSS animadas. Lo mismo ocurre si no se especifican "**to**" o "**100%**".
- El **bloque de declaraciones CSS** está formado por las propiedades y los valores.
- Dentro del bloque de declaraciones sólo pueden aparecer
 - propiedades "animables"
 - la propiedad "animation-timing-function"
- Si hay **varias declaraciones @keyframes con el mismo nombre**, sólo se tendrá en cuenta la última que aparece.

4.2.3. CSS3 – animaciones

Regla @keyframes y compatibilidad

- Se suelen definir varias reglas @keyframes, la primera genérica, y se repite con los prefijos específicos de cada navegador:
 - @keyframes
 - @-webkit-keyframes
 - @-moz-keyframes
 - @-o-keyframes
 - @-ms-keyframes

4.2.3. CSS3 – animaciones

Propiedades animation-

- Se utilizan para aplicar una animación en una regla CSS.
- Existen las siguientes propiedades:
 - ☐ **animation-name**
 - ☐ **animation-duration**
 - ☐ **animation-timing-function**
 - ☐ **animation-delay**
 - ☐ **animation-iteration-count**
 - ☐ **animation-direction**
 - ☐ **animation-play-state**
 - ☐ **animation-fill-mode**
 - ☐ **animation**

4.2.3. CSS3 – animaciones

- **animation-name:** <nombre-animacion> [, <nombre-animacion>]

Define una lista de animación a aplicar. La animación se habrá definido con una regla @keyframes.

- **animation-duration:** <tiempo> [, <tiempo>]

- ☐ Tiempo que dura la animación.
- ☐ El valor por defecto es **0s**.

- **animation-timing-function:** <f_tiempo> [, <f_tiempo>]

- ☐ Función que describe cómo se realiza la animación.
- ☐ Las funciones son las mismas que se utilizan en las transiciones.
- ☐ Si se define dentro del @keyframes (para un selector), será la función a aplicar hasta el siguiente *keyframe*.
- ☐ El valor por defecto es **ease**

https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_animation-timing-function

4.2.3. CSS3 – animaciones

- **animation-iteration-count**: <num_iteraciones> [, <num_iteraciones>]
 - Número de veces que se ejecuta la animación.
 - El valor por defecto es **1**.
 - <num_iteraciones> puede tomar los siguientes valores:
 - Un número
 - **infinite**: para que se repita indefinidamente
- **animation-direction**: <dirección> [, <dirección>]
 - Define si la animación se ejecutará en dirección inversa en alguno o todos los ciclos
 - El valor por defecto es **normal**.
 - <dirección> puede tomar los siguientes valores:
 - **normal**: la animación se ejecuta como está especificada
 - **reverse**: la animación se ejecuta en orden inverso a como está especificada
 - **alternate**: si la animación se ejecuta varias veces, la primera vez se ejecuta en orden normal, la siguiente en orden inverso y así sucesivamente.
 - **alternate-reverse**: como alternate, pero empieza por el orden inverso.

4.2.3. CSS3 – animaciones

- **animation-play-state:** <estado-animación> [, <estado-animación>]
 - Define si la animación se está ejecutando o está pausada.
 - <estado-animación> puede tomar los siguientes valores:
 - **running** (valor por defecto)
 - **paused**
- **animation-delay:** <tiempo> [, <tiempo>]
 - Tiempo que tarda en ejecutarse la animación.
 - El valor por defecto es **0s**.
- **animation-fill-mode:** <valor> [, <valor>]
 - Especifica un estilo para el elemento cuando la animación no se está reproduciendo (antes de que comience, después de que termine, o ambos).
 - Por defecto, cuando la animación termina, las propiedades tomarán de nuevo los valores que tenían antes de la animación. Con esta propiedad se puede cambiar ese comportamiento (forwards).
 - <valor> puede tomar los siguientes valores:
 - **none, forwards, backwards, both**

4.2.3. CSS3 – animaciones

Propiedad shorthand : animation

- **animation**: <animación-simple> [, < animación-simple >]
- **<animación-simple>**: <nombre-animación> || <tiempo> || <f-tiempo> || <tiempo> || <num-iteraciones> || <dirección> || <fill-mode> || <play-state>
 - El orden en que aparecen estos elementos es importante: el primer valor de tiempo será la duración, y el segundo el retraso.

Ejemplo:

```
animation: wobble 10s linear;
```

4.2.3. CSS3 – animaciones

Propiedades animation y compatibilidad

- Se suelen definir varias propiedades animation, la primera genérica, y se repite con los prefijos específicos de cada navegador:
 - ☐ animation
 - ☐ -webkit-animation
 - ☐ -moz-animation
 - ☐ -o-animation
 - ☐ -ms-animation

4.2.3. CSS3 – animaciones

- Mientras se ejecuta una animación, los valores de las propiedades animadas, son controladas por la animación.
- Cuando termina la animación, la propiedad tendrá el valor que se le haya asignado con otras reglas CSS.
- Las animaciones tienen más prioridad que los estilos normales y menos que los estilos !important.

