

# ¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.  
 Ese apoyo que siempre quiso tener...

## 1. Desarrollo de componentes y proyectos a medida



## 2. Auditoría de código y recomendaciones de mejora

## 3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



## 4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,  
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)  
 Gestor de contenidos (Alfresco)  
 Aplicaciones híbridas

Tareas programadas (Quartz)  
 Gestor documental (Alfresco)  
 Inversión de control (Spring)

Control de autenticación y  
 acceso (Spring Security)  
 UDDI  
 Web Services  
 Rest Services  
 Social SSO  
 SSO (Cas)

JPA-Hibernate, MyBatis  
 Motor de búsqueda empresarial (Solr)  
 ETL (Talend)

Dirección de Proyectos Informáticos.  
 Metodologías ágiles  
 Patrones de diseño  
 TDD

BPM (jBPM o Bonita)  
 Generación de informes (JasperReport)  
 ESB (Open ESB)

AdictosAlTrabajo



Entra en Adictos a través de

Entrar [Deseo registrarme](#)  
[Olvíde mi contraseña](#)
[Inicio](#) [Quiénes somos](#) [Formación](#) [Comparador de salarios](#) [Nuestro libro](#) [Más](#)
» [Estás en: Inicio](#) [Tutoriales](#) [Transiciones y animaciones con CSS3](#)**Miguel Arlandy Rodríguez**

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

[Ver todos los tutoriales del autor](#)
**Fecha de publicación del tutorial: 2012-07-09**  
**Transiciones y animaciones con CSS3.**
Tutorial visitado 1 veces [Descargar en PDF](#)

## 0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Las transiciones.
  - 3.1. Propiedades de las transiciones.
  - 3.2. Ejemplo 1: botones.
  - 3.2. Ejemplo 2: zoom.
  - 3.3. Ejemplo 3: Niño bailando un yoyo.
- 4. Las animaciones.
  - 4.1. Keyframes.
  - 4.2. Propiedades de las animaciones.
  - 4.3. Ejemplo simple.
  - 4.4. Ejemplo complejo.
- 5. Referencias.
- 6. Conclusiones.

## 1. Introducción

Los nuevos estándares de CSS proporcionan a los diseñadores web cada vez más recursos para el desarrollo de sus sitios. Todos recordamos las webs de primeros de siglo, esas de las que normalmente salimos huyendo cuando nos las encontramos y que en nada se parecen a las de ahora.

CSS3 pretende estandarizar dos nuevas características: las transiciones y las animaciones que actúan sobre el estado de un elemento transformándolo de manera gradual. Con esto se consiguen efectos muy agradables que mejoran notablemente la experiencia del usuario al interactuar con la página.

Sin embargo estas características, a fecha de hoy (Julio 2012), son muy modernas y todavía no están estandarizadas, por lo que los navegadores las van incluyendo poco a poco en sus nuevas versiones. De hecho, Internet Explorer todavía no las soporta en su versión 9, aunque Microsoft asegura que las incluirá a partir de la versión 10.

En este tutorial vamos a intentar explicar qué son las transiciones y animaciones en CSS. Veremos sus características y aprenderemos a transformar y animar elementos de nuestras páginas web únicamente con CSS. Un hecho que antes era impensable sin ayuda de Flash o Javascript.

## 2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.2 Ghz Intel Core I7, 8GB DDR3).
- Sistema Operativo: Mac OS Snow Leopard 10.6.7
- Entorno de desarrollo: IntelliJ Idea 11.1 Ultimate.
- Mozilla Firefox 13
- Google Chrome 20
- Safari 5.1
- Gimp 2.6.1

## 3. Las transiciones.

Vamos a empezar este tutorial explicando qué son las transiciones. Las transiciones no son otra cosa que el efecto de un cambio en el estado de un elemento de forma gradual en un intervalo de tiempo. Cuando hablamos de cambio de estado nos referimos a cambios en los valores de sus propiedades.

## Catálogo de servicios Autentia

Soporte a  
Desarrollo, Factoría  
y Formación



Síguenos a través de:



## Últimas Noticias

» [Autentia colabora con la ONG Proyecto Ciclista Solidario](#)» [Curso de Kanban Core en Madrid con Masa K. Maeda](#)» [Failure demand](#)» [Comentando el Libro: Lean StartUp \(el método\)](#)» [VIII Autentia Cycling Day](#)[Histórico de noticias](#)

## Últimos Tutoriales

» [Configuración y tuning de servidores de producción](#)» [Revisión de jBPM5](#)» [Database MessageSource: obtener los literales de una base de datos](#)» [Comparando diferencias entre ficheros con java-diff-utils](#)» [Mejorando la calidad de nuestro código PHP](#)

## Últimos Tutoriales del Autor

Por ejemplo, supongamos que tenemos un párrafo (elemento <p> de un documento HTML) cuyo color de fuente es rojo. El color de la fuente sería una propiedad del elemento párrafo (propiedad "color"). Imaginemos que queremos que el color de la fuente en un momento determinado cambie a azul. Con Javascript fácilmente podríamos cambiar la propiedad 'color' del elemento párrafo a azul y el usuario inmediatamente vería que el color del texto del párrafo ya no es rojo sino azul. Sin embargo, eso NO es una transición. La transición se basa en que el elemento tiene que cambiar el valor de la propiedad de forma gradual durante un intervalo de tiempo. Es decir, si decidimos que el intervalo de tiempo sea de 2 segundos, la transición sería el efecto que se produce al ver como, en esos 2 segundos, el color de la letra va cambiando progresivamente de rojo a azul. Podríamos decir que es una especie de 'morphing'.

Este tipo de efectos producen en una página la sensación de que está "viva" :).

A fecha de hoy (ver fecha de publicación del tutorial), las transiciones únicamente son soportadas por los navegadores: Firefox, Chrome, Safari y Opera. Internet Explorer (al menos hasta su versión 9) no las soporta.

Visto esto, vamos a empezar a explicar cómo usar transiciones en nuestras webs.

### 3.1 Propiedades de las transiciones.

Para poder aplicar transiciones sobre un determinado elemento contamos con las siguientes propiedades:

- **transition-property:** indica sobre qué propiedad del elemento se aplicará la transición. Pueden especificarse varias propiedades separándolas por comas. El valor por defecto es 'all' que indica que la transición se aplicará sobre todas las propiedades del elemento. Un valor válido para esta propiedad sería, por ejemplo, 'width'.
- **transition-duration:** indica el tiempo que tardará la transición en completarse. El valor por defecto es 0. Ejemplos de valores válidos para esta propiedad pueden ser: 2s (2 segundos), 300ms (300 milisegundos), etc...
- **transition-timing-function:** indica cómo será la progresión de la transición. El valor por defecto es 'ease'. Los valores que acepta esta propiedad son los siguientes:
  - **linear:** la transición se produce en forma de progresión lineal. Tarda lo mismo desde que empieza hasta que acaba.
  - **ease:** la transición se produce de forma lenta al principio, luego más rápido, y termina otra vez lenta (valor por defecto).
  - **ease-in:** la transición comienza lenta y luego va más rápido.
  - **ease-out:** la transición comienza rápida y termina lenta.
  - **ease-in-out:** la transición comienza y acaba lenta. Igual que 'ease' pero de forma más notable.
  - **cubic-bezier(n,n,n,n):** permite definir nuestro propio intervalo de progresión. n: valor entre 0 y 1.
- **transition-delay:** es el tiempo que tardará la transición en comenzar. Por defecto es 0.
- **transition:** permite unificar todas las propiedades anteriores en una sola. Un ejemplo sería `transition: width 3s ease-in 200ms`

**Importante:** por el momento los navegadores que proveen estas propiedades requieren que se especifiquen anteponiendo el prefijo de cada navegador a la propiedad. Esto es, para Firefox hay que anteponer el prefijo -moz- a la propiedad (ej: -moz-transition-duration:3s), para Chrome o Safari sería -webkit- y para Opera -o-. ¿Que por qué sucede esto? Pues porque estas propiedades todavía no están estandarizadas, por lo que los proveedores (navegadores) las consideran experimentales. Pero tranquilos que es cuestión de tiempo que se estandarice... :)

### 3.2 Ejemplo 1: botones.

Vamos a ver un primer ejemplo de transiciones aplicado a unos botones (realmente son enlaces). Lo que haremos será hacer que se produzca una transición en las propiedades de los botones cuando el usuario pase el ratón por encima.

Como en toda transición tenemos un elemento con una propiedad o propiedades iniciales y unas propiedades finales. Nuestros botones tendrán el siguiente aspecto:



Si nos fijamos en el botón 1 observamos que cambian dos propiedades: el color del texto y el color de fondo. En el botón 2 cambian el color del texto, el color de fondo, el color del borde y el margen interno (padding).

Ahora definiremos las propiedades de nuestra transición.

Queremos que para el botón 1:

- La transición se aplique a todos los elementos que cambien (`transition-property:all`).
- La transición dure 1 segundo (`transition-duration: 1s`).
- La transición se produzca de forma lenta al principio, luego rápido y acabe lenta (`transition-timing-function: ease`).
- La transición se produzca con un retardo de 200ms tras pasar el ratón por encima (`transition-delay:0.2s`).

Queremos que para el botón 2:

- La transición se aplique a todos los elementos que cambien (`transition-property:all`).
- La transición dure 300 milisegundos (`transition-duration: 300ms`).
- La transición se produzca de forma rápida pero lenta al final (`transition-timing-function: ease-out`).
- La transición se produzca con un retardo de 20ms tras pasar el ratón por encima (`transition-delay:20ms`).

Las reglas y propiedades en la css serían las siguientes:

```

1  a.link1, a.link2 {
2      padding: 10px 20px;
3      border-radius: 5px;
4  }
5
6  a.link1 {
7      color: #FFF;
8      background: #9ab8d2;
9      border: 1px solid #5380d2;
10
11     /* Firefox */
12     -moz-transition-property: all;
13     -moz-transition-duration: 1s;
14     -moz-transition-timing-function: ease;
15     -moz-transition-delay: 0.2s;
16
17 
```

» Comparando diferencias entre ficheros con java-diff-utls

» JQuery Waypoints: realizando acciones al llegar a un punto de la página con el scroll.

» Creando un videojuego con HTML5 y Javascript

» Trabajando con IntelliJ IDEA 11

» JsTestDriver: testea tu código Javascript

### Últimas ofertas de empleo

2011-09-08

Comercial - Ventas - MADRID.

2011-09-03

Comercial - Ventas - VALENCIA.

2011-08-19

Comercial - Compras - ALICANTE.

2011-07-12

Otras Sin catalogar - MADRID.

2011-07-06

Otras Sin catalogar - LUGO.

```

18  /* Chrome / Safari */
19  -webkit-transition-property: all;
20  -webkit-transition-duration: 1s;
21  -webkit-transition-timing-function: ease;
22  -webkit-transition-delay: 0.2s;
23
24
25  /* Opera */
26  -o-transition-property: all;
27  -o-transition-duration: 1s;
28  -o-transition-timing-function: ease;
29  -o-transition-delay: 0.2s;
30
31
32  /* Default */
33  transition-property: all;
34  transition-duration: 1s;
35  transition-timing-function: ease;
36  transition-delay: 0.2s;
37  }
38
39  a.link1:hover {
40  background-color:#5380d2;
41  color: #a8bfd2;
42  }
43
44  a.link2 {
45  color: #9ca3a3;
46  background: #d4dbdb;
47  border: 2px solid #b3baba;
48
49  /* Firefox */
50  -moz-transition-property: all;
51  -moz-transition-duration: 300ms;
52  -moz-transition-timing-function: ease-out;
53  -moz-transition-delay: 20ms;
54
55  /* Chrome / Safari */
56  -webkit-transition-property: all;
57  -webkit-transition-duration: 300ms;
58  -webkit-transition-timing-function: ease-out;
59  -webkit-transition-delay: 20ms;
60
61
62  /* Opera */
63  -o-transition-property: all;
64  -o-transition-duration: 300ms;
65  -o-transition-timing-function: ease-out;
66  -o-transition-delay: 20ms;
67
68
69  /* Default */
70  transition-property: all;
71  transition-duration: 300ms;
72  transition-timing-function: ease-out;
73  transition-delay: 20ms;
74  }
75
76  a.link2:hover {
77  background-color:#9ca3a3;
78  color: #333;
79  border: 2px solid #636a6a;
80  padding: 10px 30px;
81  }

```

Recordemos que la propiedad "transition" permite unificar las propiedades en una sola, por lo que para el botón 1 sería válida la propiedad: **transition: all 1s ease 0.2s** y para el botón 2: **transition: all 300ms ease-out 20ms**. Siempre anteponiendo el prefijo del navegador (-moz-, -webkit-, -o-).

[Podéis ver el ejemplo en acción aquí](#)

### 3.3 Ejemplo 2: zoom.

Vamos a ver un segundo ejemplo de transiciones aplicado a lo que sería un zoom en un texto.

Queremos que cuando un usuario pase el ratón por encima de un texto contenido en un párrafo, el tamaño de letra se incremente.

Queremos que para nuestro párrafo:

- La transición se aplique a la propiedad font-size (transition-property:font-size).
- La transición dure 300 milisegundos (transition-duration: 300s).
- La transición se produzca de forma siempre a la misma velocidad (transition-timing-function: linear).
- La transición se produzca sin retardo (transition-delay:0s).

Esta vez estableceremos el valor de la transición en una única propiedad "transition". La css quedaría de la siguiente forma:

```

1  p.zoom {
2  font-size: 12px;
3  width: 300px;
4
5  /* Firefox */
6  -moz-transition: font-size 300ms linear 0s;
7  /* Chrome / Safari */
8  -webkit-transition: font-size 300ms linear 0s;
9  /* Opera */
10 -o-transition: font-size 300ms linear 0s;
11 /* Default */
12 transition: font-size 300ms linear 0s;
13 }
14
15 p.zoom:hover {
16 font-size: 16px;
17 }

```

[Podéis ver el ejemplo en acción aquí](#)

### 3.4 Ejemplo 3: Niño bailando un yoyo.

En este último ejemplo vamos a simular un niño bailando un yoyo. Para ello contaremos con dos imágenes:

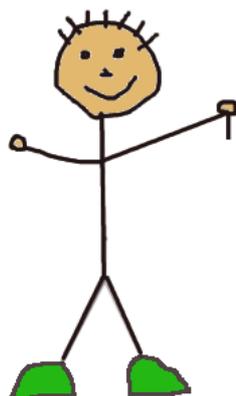


Imagen del niño



Imagen del yoyo

Más de uno ya se habrá dado cuenta de que pintar no es mi fuerte :-S. Si alguien me pudiera recomendar un buen manual sobre Gimp se lo agradeceré eternamente... :-D

Pues bien, lo que haremos será establecer las dos imágenes como imagen de fondo de dos capas (div). Ajustaremos las posiciones de cada capa de tal forma que la imagen del yoyo "case" con la mano del niño. Por último aplicaremos la transición al alto (height) de la capa que contiene la imagen de fondo del yoyo para que se produzca el efecto de movimiento cuando el usuario pase el botón por encima.

Por tanto, queremos que la capa que contiene la imagen del yoyo tenga las siguientes propiedades:

- La transición se aplicará a la propiedad height (transition-property:height).
- La transición durará 1s segundo (transition-duration: 1s).
- La transición se producirá siempre a la misma velocidad (transition-timing-function: linear).
- La transición se producirá con un retardo de 200ms (transition-delay:0.2s).

Por comodidad volveremos a establecer el valor de la transición en una única propiedad "transition". La css quedaría de la siguiente forma:

```

1  div#kid {
2      box-shadow: 0 0 15px #000;
3      width: 210px;
4      height: 300px;
5      padding: 0;
6      background: url("../img/nino-yoyo.png") no-repeat top left;
7  }
8
9  div#yoyo {
10     padding: 0;
11     background: url("../img/yoyo.png") no-repeat bottom right;
12     width: 65px;
13     height: 75px;
14     position: relative;
15     right: 100%;
16     left: 145px;
17     top: 100px;
18
19     /* Firefox */
20     -moz-transition: height 1s linear 0.2s;
21     /* Chrome / Safari */
22     -webkit-transition: height 1s linear 0.2s;
23     /* Opera */
24     -o-transition: height 1s linear 0.2s;
25     /* Default */
26     transition: height 1s linear 0.2s;
27 }
28
29 div#kid:hover div#yoyo {
30     height: 200px;
31 }

```

Podéis ver el ejemplo en acción [aquí](#)

## 4. Las animaciones.

Las animaciones son bastante parecidas a las transiciones. Ambas tienen en común que actúan sobre los valores de las propiedades de un elemento transformándolos gradualmente en otros valores. Sin embargo, las animaciones ofrecen bastante más control que las transiciones en el proceso de transformación del elemento.

Las transiciones se ejecutan cuando se produce un cambio en las propiedades de un elemento. Ya lo vimos en el punto anterior, cuando el usuario pasa el ratón por encima de un elemento y la css indica que el valor de determinadas propiedades cambia (:hover), entonces se lanza la transición.

Las animaciones son algo distintas ya que se lanzan explícitamente. No se espera a que suceda un cambio en el estado del elemento como pasaba con las transiciones. Es por ello que para usar animaciones necesitamos "keyframes". Estos "keyframes" describirán el comportamiento que tendrá la animación durante su periodo de ejecución.

A diferencia de las transiciones, las animaciones (de momento) no son soportadas por el navegador Opera.

### 4.1 Keyframes.

Como hemos dicho, los keyframes indicarán el comportamiento que tendrá un elemento durante el tiempo que dure la animación.

Para definir el comportamiento de una animación mediante un keyframe lo hacemos de la siguiente manera:

```

1  @keyframes nombre-animacion {
2      0% {
3          left: 10px;
4      }
5
6      25% {
7          left: 150px;
8      }
9
10     50% {
11         left: 160px;
12     }
13
14     75% {
15         left: 300px;
16     }
17
18     100% {
19         left: 310px;
20     }
21 }
```

Como vemos, nombre-animacion será el nombre que le demos a nuestra animación y que será referenciada por el elemento que vayamos a animar (lo veremos más adelante). Los porcentajes que hay dentro de la regla indicarán el estado de la propiedad (o propiedades del elemento) en ese % de duración de la animación. Si por ejemplo, nuestra animación durase 10 segundos, nuestro elemento comenzaría con una distancia a la izquierda del elemento que lo contiene (propiedad left) igual a 10px. Transcurrido el 25% del tiempo (2.5 segundos) su posición sería de 150px, transcurrido el 50% del tiempo (5 segundos) su posición sería de 160px, y así sucesivamente...

Por supuesto, podemos añadir tantos porcentajes como queramos. Notese que el 0% se puede sustituir por "from" y el 100% por "to".

#### 4.2 Propiedades de las animaciones.

Las propiedades de las animaciones son bastante parecidas a las de las transiciones. Para poder aplicar animaciones sobre nuestros elementos contamos con las siguientes propiedades:

- animation-name: es el nombre de la animación que actuará sobre el elemento. Recordemos que en el punto anterior (keyframes) para definir una animación mediante un keyframe debíamos añadir también el nombre de la animación. Pues es ese nombre.
- animation-duration: el tiempo que tardará la animación desde su inicio hasta su fin.
- animation-timing-function: del mismo modo que en las transiciones, esta propiedad indica cómo progresará la animación. Acepta los mismos valores que la propiedad transition-timing-function.
- animation-iteration-count: indica el número de veces que se repetirá la animación. Su valor por defecto es 1. El valor "infinite" indica que se repetirá indefinidamente.
- animation-direction: indica el orden en el que ejecutará la animación. Acepta los valores "normal" y "alternate". Si tuviésemos dos repeticiones de la animación (animation-iteration-count: 2) y el valor de esta propiedad fuese alternate, primero se ejecutaría la animación en orden normal 0 a 100% y luego al revés 100% a 0. Si fuese normal, las dos veces se ejecutaría de 0 a 100%. El valor por defecto es normal.
- animation-play-state: indica si la animación está corriendo o pausada. Acepta los valores running (valor por defecto) y paused.
- animation-delay: igual que en las transiciones, indica el retardo con el que se iniciará la animación. Por defecto 0.
- animation-fill-mode: pues básicamente lo que hace esta propiedad es decidir si los valores de las propiedades que ha modificado la animación vuelven a su estado original (antes de la animación) o no. Realmente vale para ver qué se hace con los valores de las propiedades fuera del periodo de duración de la animación. Acepta los valores: "none", "forwards", "backwards" y "both". El valor por defecto es none que indica que cuando acabe la animación el elemento recuperará los valores que tenían de sus propiedades antes de iniciarse la animación.
- animation: permite unificar las propiedades anteriores en una sola. Sería algo así: animation: mi-animacion 3s ease 0.5s 1 normal none.

**Importante:** como sucedía con las transiciones es necesario anteponer el prefijo de cada navegador a cada propiedad. Además las animaciones solo son soportadas por Firefox (prefijo -moz-), por Chrome (prefijo -webkit-) y Safari (-webkit-). Lo mismo sucede con los keyframes hay que definirlos de la forma @-moz-keyframe mi-animacion (caso de Firefox). Tranquilos, que como pasa con las transiciones, pronto se estandarizarán.

#### 4.3 Ejemplo simple.

Vamos a ver primero un sencillo ejemplo para hacernos una idea de cómo hacer animaciones y luego veremos otro bastante más complejo.

En este ejemplo vamos a animar un cuadro (div) de color rojo y haremos que se mueva hacia la derecha. En mitad del camino cambiará de color (lo hará de forma progresiva) a azul. Luego volverá a su color original. La animación durará 2 segundos y, en cada intervalo definido, el cuadro progresará de forma más rápida al principio y más lenta al final. La animación se repetirá 10 veces.

```

1  div#simple {
2      width: 50px;
3      height: 50px;
4      background-color: red;
5      position: absolute;
6
7      -moz-animation-name: simple-animacion;
8      -moz-animation-duration: 2s;
9      -moz-animation-delay: 0s;
10     -moz-animation-iteration-count: 10;
11     -moz-animation-timing-function: ease-out; /* rápido al principio lento al final */
12     -moz-animation-direction: normal;
13
14     -webkit-animation-name: simple-animacion;
15     -webkit-animation-duration: 2s;
16     -webkit-animation-delay: 0s;
17     -webkit-animation-iteration-count: 10;
18     -webkit-animation-timing-function: ease-out;
19     -webkit-animation-direction: normal;
20 }
21 }
```

```

22 @-moz-keyframes simple-animation {
23     0% { left: 0;
24         }
25     }
26
27     50% {
28         left: 200px;
29         background-color: blue;
30     }
31
32     100% {
33         left: 400px;
34     }
35 }
36
37 @-webkit-keyframes simple-animation {
38     0% { left: 0;
39         }
40     }
41
42     50% {
43         left: 200px;
44         background-color: blue;
45     }
46
47     100% {
48         left: 400px;
49     }
50 }

```

El resultado sería algo así:



Estado del elemento al iniciar la animación (0%).



Estado del elemento en la mitad de la animación (50%).



Estado del elemento al finalizar la animación (100%).

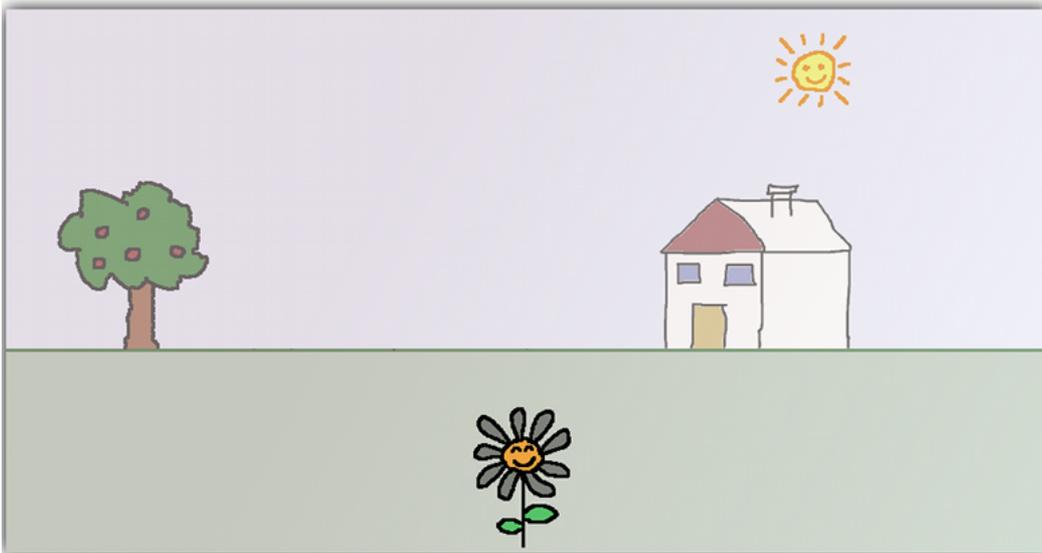
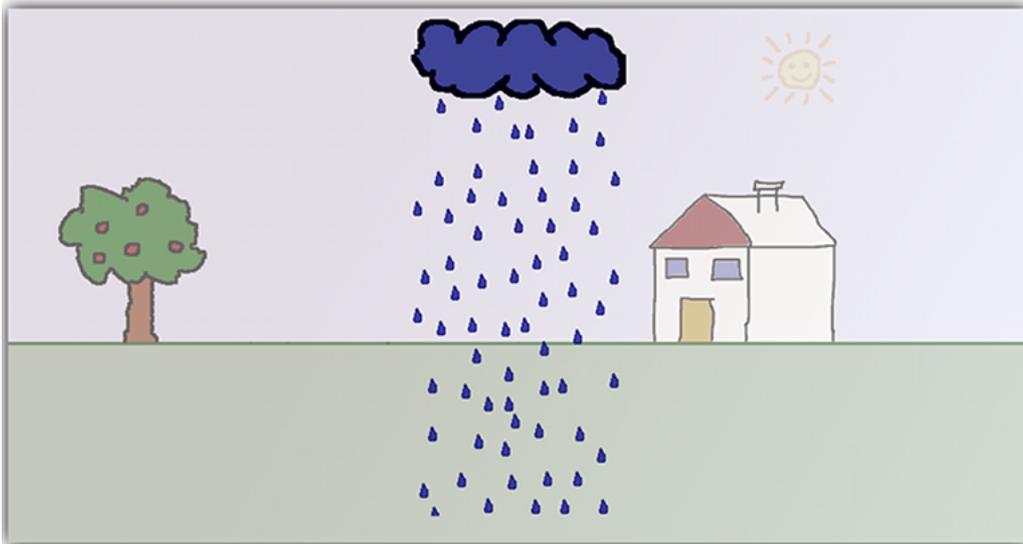
#### 4.4 Ejemplo complejo.

Pues llegados a este punto y con un poquito de imaginación ya estamos en disposición de hacer un ejemplo más complejo. Vamos a ver un ejemplo de animaciones con CSS3 donde se ve como una nube descarga agua sobre un prado, luego luce el sol y sale una bonita flor. Y alguno estará diciendo "**¡vaya ejemplo más cursi!**". Pues sí, tiene toda la razón del mundo, pero es que todavía no sé hacer a un Espartano cortando las cabezas de sus enemigos Persas. En cuanto aprenda lo publico en otro tutorial ;-)...

El truco para realizar este tipo de animaciones está en jugar con los retardos (animation-delay) de cada uno de los elementos que entran en acción, ya que todos comienzan la animación a la vez. Recordemos que esto no es Javascript, aquí nuestros actores no interactúan, es todo "cartón-piedra".

[Aquí podéis ver el ejemplo.](#) Para ver la CSS y el HTML, simplemente explorar el código fuente de la página o [pulsar aquí para ver la CSS.](#)





## 5. Referencias.

- W3C: Transiciones con CSS3
- W3C: Animaciones con CSS3
- Ejemplo de transiciones con CSS3.
- Ejemplo de animación con CSS3.

## 6. Conclusiones.

En este tutorial hemos visto cómo realizar transiciones y animaciones sin escribir ni una sola línea de Javascript, únicamente con CSS. Este tipo de efectos, mejoran notablemente la experiencia del usuario al interactuar con la página.

Además de los ejemplos que hemos visto en el tutorial, puede ser muy interesante su uso en otros escenarios como galerías de imágenes o menús de navegación con sub-menús.

El problema es un poco el de siempre, son propiedades relativamente nuevas que todavía no están estandarizadas y no todos los navegadores las soportan (véase Internet Explorer). Además, los que si que las soportan las consideran todavía experimentales, por lo que tenemos que añadir prefijos concretos de cada navegador en la definición de nuestras propiedades y, como consecuencia, la gran cantidad de código duplicado que eso genera. Sin embargo, pronto se estandarizarán y dejarán de ser necesarios los prefijos. Además, Internet Explorer tiene pensado incluirlas en su versión 10.

Espero que este tutorial os haya sido de ayuda. Un saludo.

Miguel Arlandy

[marlandy@autentia.com](mailto:marlandy@autentia.com)

Twitter: [@m\\_arlandy](https://twitter.com/m_arlandy)

**A continuación puedes evaluarlo:**

[Regístrate para evaluarlo](#)

**Por favor, vota +1 o compártelo si te pareció interesante**

Share |

0

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA

Impulsores

Comunidad

¿Ayuda?

----  
sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by [kamacracy](#)

Copyright 2003-2012 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

