

UD02. CSS. Animaciones

Diseño de interfaces WEB
2º Curso

- 
- Animaciones
 - Fotogramas claves
 - Asignación
 - Reproducción
 - Encadenar animaciones

Animaciones

La idea de la animación consiste en encadenar varias transiciones a una misma propiedad.

Constan de dos componentes:

- Un estilo que describe la animación
- Un conjunto de fotogramas que indican varios de los estados de la misma

Animaciones

CSS vs JS

Tradicionalmente las animaciones se realizaban con JS. CSS aporta algunas ventajas

- Es el navegador el que controla la visualización, por lo que puede optimizar de manera automática el rendimiento de la animación en función de los recursos disponibles
- No es necesario (en animaciones sencillas) saber JS

Fotogramas claves

Fotograma clave: descripción del elemento animado en un momento dado de la animación.

Mínimo hay dos: el punto inicial (0% o `from`) y el punto final (100% o `to`).

Se definen mediante el tag `@keyframes`

```
@keyframes miAnimacion {  
  from {  
    width: 100px;  
  }  
  to {  
    width: 300px;  
  }  
}
```

Fotogramas claves

Es posible definir puntos intermedios, indicando el porcentaje de la reproducción al que hace referencia:

```
@keyframes miAnimacion {  
    from {  
        width: 100px;  
    }  
    50% {  
        width: 150px;  
    }  
    to {  
        width: 300px;  
    }  
}
```

Asignación

Para realizar la animación a un elemento se utilizan las propiedades *animation*-*

- *animation-delay*: tiempo de retardo en segundos entre el momento en que el elemento se carga y el comienzo de la secuencia de la animación.
- *Animation-direction*.
 - normal: desde el inicio hasta el final.
 - reverse: desde el final hasta el inicio.
 - alternate: de inicio a fin en las iteraciones impares y hacia atrás en las pares.
 - alternate-reverse: hacia atrás en las iteraciones impares y hacia delante en las pares.

Asignación

Para realizar la animación a un elemento se utilizan las propiedades *animation*-*

- *animation-iteration-count*: número de veces que se repite. *infinite* = indefinidamente.
- *animation-duration*: duración de la animación (por ciclo)
- *animation-name*: nombre de la regla @keyframes que describe los fotogramas de la animación.
- *animation-play-state*: Estado de la animación: *running* o *paused*. Puede ser utilizado desde JS
- *animation-timing-function*: ritmo de la animación: *ease*, *linear*, *ease-in*, *ease-out*, *ease-in-out*.
- *animation-fill-mode*: qué valores tendrán las propiedades después de finalizar la animación: *none* (ningún estilo aplicado), *forwards* (se queda con lo estilos del final), *backwards* (se queda con el estilo del principio).

Asignación

Estas propiedades se definen en un estilo, que al ser aplicado hará que la animación empiecen:

```
.divAnimado {  
    animation-duration: 3s;  
    animation-name: miAnimacion;  
    animation-iteration-count: 1;  
}
```

Asignación

El estilo será aplicado cuando se produzca un evento.

Ese evento puede ser lanzado de muchas maneras: mediante la aplicación de un hover, o un script de JS que cambie la clase aplicada

```
<button id="iniciaAnimacion">Run</button>
<div id="miDiv"></div>
function iniciar(){
    var iniAni=document.getElementById("iniciaAnimacion");
    iniAni.addEventListener("click", accIniAni, false); }

function accIniciaAnimacion() {
    var div=document.getElementById("miDiv");
    div.className = "divAnimado"; }

window.addEventListener("load", iniciar, false);
```

Encadenar animaciones

Es posible crear animaciones para varios elementos y hacer que una empiece cuando acabe otra.

Para ello se utilizan eventos que permiten controlar el estado de las animaciones:

- *animationstart*: se inicia la primera iteración de la animación
- *animationiteration*: ha iniciado una iteración (excepto la primera)
- *animationend*: la animación ha finalizado.

Encadenar animaciones

```
.divAnimado {  
    animation-duration: 3s;  
    animation-name: miAnimacion;  
    animation-iteration-count: 1;  
}
```

```
.divAnimado2 {  
    animation-duration: 3s;  
    animation-name: miAnimacion2;  
    animation-iteration-count: 1;  
}
```

Encadenar animaciones

```
function iniciar(){  
    var iniAni=document.getElementById("iniciaAnimacion");  
    iniAni.addEventListener("click", accIniAni, false); }  
  
function accIniAni() {  
    var div=document.getElementById("miDiv");  
    div.className = "divAnimado";  
    div.addEventListener("animationend", accIniAni2, false); }  
  
function accIniAni2() {  
    var div2=document.getElementById("miDiv2");  
    div2.className = "divAnimado2"; }  
  
window.addEventListener("load", iniciar, false);
```

Compatibilidad

Tanto las transformaciones como las transiciones pueden tener problemas de compatibilidad entre diferentes navegadores.

- Para alcanzar la mayoría de ellos es recomendable utilizar los prefijos de cada navegador (-webkit-, -moz-, -o-, -ms-)
- Es conveniente conocer usar una tabla de compatibilidad para saber qué y donde podemos usar cada propiedad

<http://caniuse.com/>