

{css}



Clase 34

Clase 34

<índice>

Animaciones en CSS

¿Qué son?

Transiciones

Animaciones y parámetros

@keyframes

Funciones de tiempo

¿Qué son?

¿Qué son?

Cambios de estilos

Por ejemplo con :hover

El cambio se hace de forma instantánea

Cambiemos esto...

¿Qué son?

Interactividad
Dinamismo
Pero sin pasarse...

Transiciones

Transiciones

Utilizando transiciones tenemos a nuestra disposición una gran flexibilidad que nos permitirá dotar de atractivos efectos de transición que harán que nuestros diseños sean más elegantes y agradables.

Transiciones

Las transiciones se basan en un principio muy básico: conseguir un efecto suavizado entre un estado inicial y un estado final al realizar una acción.

Transiciones

Las propiedades CSS que podemos utilizar relacionadas con las transiciones son las siguientes

Transiciones

Propiedades de transición

Propiedades	Descripción	Valor
transition-duration	Tiempo de duración.	0 TIME
transition-property	Propiedades CSS afectadas por la transición.	all none propiedad css
transition-timing-function	Ritmo de la transición.	Ver funciones de tiempo
transition-delay	Tiempo de retardo inicial.	0 TIME

Transiciones

Ten en cuenta que por defecto las transiciones están habilitadas para todas las propiedades CSS involucradas. Sin embargo, tienen una duración de 0s. Salvo que cambiemos esta duración, no se realizará transición.

Transiciones

transition-duration

Con la propiedad transition-duration especificaremos la duración de la transición, desde el inicio de la transición, hasta su finalización. Recuerda que por defecto, las transiciones tienen una duración de 0s, por lo que si no cambias este valor, cambiará de golpe y será lo mismo que no tener transición.

Transiciones

Ojo: Si establecemos una duración demasiado alta, el navegador realizará la transición a trozos intermitentes, lo que hará que se vea a golpes. Se recomienda siempre comenzar con valores cortos, para que las transiciones sean rápidas y elegantes.

Transiciones

```
.element {  
    transition-duration: 0.5s;  
    width: 200px;  
    height: 200px;  
    background: grey;  
}  
  
.element:hover {  
    background: indigo;  
}
```

Demo

Transiciones

transition-property

En primer lugar, la propiedad transition-property se utiliza para especificar la propiedad a la que afectará la transición. Podemos especificar la propiedad concreta (width o color, por ejemplo) o simplemente especificar all para que se aplique a todos los elementos con los que se encuentre. Por otro lado, none hace que no se aplique ninguna transición.

Transiciones

Propiedades	Valor
all	Aplica la transición a todas las propiedades css.
none	No aplica transición. El cambio se producirá de golpe (brusco).
propiedad css	Aplica la transición sólo a la propiedad css especificada.

Transiciones

Ten en cuenta que puedes animar todas las propiedades CSS afectadas utilizando all

Transiciones

```
.element {  
    /* Primer ejemplo: Anima todas las propiedades que cambien de estado */  
    transition-property: all;  
    transition-duration: 1s;  
  
    /* Segundo ejemplo: Anima sólo el ancho, el resto no transicionan, cambian de golpe */  
    transition-property: width;  
    transition-duration: 1s;  
}
```

Transiciones

Nota: Debes saber que no todas las propiedades CSS son animables (o transicionables). Por norma general, la mayoría de las propiedades que son cuantificables (valores numéricos, porcentajes, valores hexadecimales como colores, etc...) son animables, sin embargo, otras propiedades no lo son. Por ejemplo, la propiedad background-image para gradientes no se puede animar actualmente.

Transiciones

Demo

Transiciones

transition-delay

Por último, la propiedad transition-delay nos ofrece la posibilidad de retrasar el inicio de la transición un número de segundos determinado.

Si se omite, la transición comienza inmediatamente.

Transiciones

Veamos un pequeño ejemplo de todas estas propiedades de transición. Utilizaremos la función de tiempo linear, que luego te la cuento

Transiciones

```
a {  
    display: inline-block;  
    background: #ddd;  
    color: #222;  
    padding: 8px 16px;  
  
    transition-property: all;  
    transition-duration: 0.5s;  
    transition-timing-function: linear;  
}  
  
a:hover {  
    background: #fff;  
    color: darkred;  
    padding: 32px;  
    border: 1px solid darkred;  
}
```

Demo

Transiciones

- La propiedad background de color de fondo cambiará de #ddd a #fff
- La propiedad color de color de texto cambiará de #222 a #666
- La propiedad padding del tamaño del relleno cambiará de 8px a 25px
- La propiedad border cambiará de 0 a 1px solid #888.

Transiciones

Este último es un caso especial, puesto que cambia de estilos porque toma el estilo inicial por defecto, que es un borde de 0px de grosor. Cada una de estas transiciones se realizarán a un ritmo lineal, durante 0.5s de duración.

Transiciones

transition (shorthand)

Como siempre, podemos resumir todas estas operaciones en una propiedad de atajo denominada transition. Los valores del ejemplo superior, se podrían escribir como se puede ver a continuación (si no necesitas algún valor, se puede omitir)

Transiciones

```
.box {  
  /* transition: <property> <duration> <timing-function> <delay> */  
  transition: all 0.5s linear;  
  
  background: grey;  
  width: 200px;  
  height: 200px;  
}  
  
.box:hover {  
  background: indigo;  
}
```

Transiciones

Observa que estamos omitiendo el último valor. Si quisieramos un retardo de 2s a la hora de hacer una transición, bastaría con añadir un 2s después de linear.

Transiciones

Transiciones de entrada y de salida

Transiciones

Si nos fijamos bien, la transición del ejemplo anterior se aplica sólo al mover el ratón sobre el elemento (transición de entrada). Sin embargo, si movemos el ratón fuera del enlace, no se produce transición, sino que realiza el cambio de forma brusca. Esto ocurre porque le estamos diciendo que solo realice la transición cuando tenemos el ratón encima (:hover), en caso contrario no lo hará porque no hay definidas propiedades de transición.

Transiciones

```
a {  
background: black;  
color: white;  
padding: 8px;  
transition: background 0.2s linear;  
}  
  
a:hover {  
background: red;  
transition: background 2s linear;  
}
```

Transiciones

También es posible que queramos indicar, por ejemplo, diferentes duraciones dependiendo de la propiedad CSS. Por ejemplo, que tarde mucho en cambiar el ancho del elemento, pero muy poco en cambiar el color de fondo.

Para ello, podemos separar con comas las diferentes propiedades que queremos transicionar

Transiciones

```
.box {  
    width: 200px;  
    height: 200px;  
    background: grey;  
    transition:  
        width 3s,  
        background-color 0.5s;  
}  
  
.box:hover {  
    width: 400px;  
    background: deeppink;  
}
```

@keyframes

Cuando ya hemos utilizado propiedades CSS como animation (o de su familia, animation-*), nos falta una parte muy importante: definir los fotogramas de la animación.

Una animación esta formada por varios fotogramas, una secuencia de imágenes (30-60 fotogramas por segundo, por ejemplo) que mostradas una detrás de otra generan el efecto de movimiento que conocemos de una animación.

@keyframes

En CSS, los fotogramas se crean a partir de propiedades CSS, y no hace falta definir tantos fotogramas. Sólo crearemos fotogramas clave y el resto de fotogramas los generará el navegador.

Para definir esos fotogramas clave, utilizaremos la regla `@keyframes`, la cuál es muy sencilla de utilizar. Se basa en el siguiente esquema

@keyframes

```
@keyframes nombre-animation {  
    time-selector {  
        propiedad : valor ;  
        propiedad : valor  
    }  
}
```

@keyframes

Cada uno de estos time-selector será un momento clave de cada uno de los fotogramas clave de nuestra animación, y ya veremos que pueden definirse muchos en una misma animación.

@keyframes

Selectores from y to

Parte	Descripción
@keyframes STRING name	Regla para darle un nombre y definir los fotogramas clave de una animación.
from	Fotograma clave inicial con los estilos CSS a aplicar. Equivalente a 0%.
to	Fotograma clave final con los estilos CSS a aplicar. Equivalente a 100%.
PERCENT	Porcentaje específico de la animación con los estilos CSS a aplicar. Permite decimales.

```
@keyframes change-color {  
  from { background: red; } /* Primer fotograma */  
  to { background: green; } /* Segundo y último fotograma */  
}
```

Como vemos, hemos nombrado change-color la animación, que parte de un primer fotograma clave con el fondo rojo hasta un último fotograma clave con fondo verde.

@keyframes

Recuerda que no basta con definir la animación mediante @keyframes, también que hay que asociar la animación al elemento o etiqueta HTML al que queremos aplicársela

@keyframes

```
.element {  
    width: 100px;  
    height: 100px;  
    animation: change-color 1.5s linear infinite;  
}  
  
@keyframes change-color {  
    from { background: red; } /* Primer fotograma */  
    to { background: green; } /* Segundo y último fotograma */  
}
```

@keyframes

La magia de la regla @keyframes es que nosotros solo declaramos los fotogramas clave, mientras que el navegador irá generando los fotogramas intermedios para conseguir una animación fluida.

Los selectores from y to son muy similares a colocar 0% y 100%, así que los modificaremos y de esta forma podremos ir añadiendo nuevos fotogramas intermedios. Vamos a añadir un fotograma intermedio e indentando, ahora sí, correctamente el código

@keyframes

```
@keyframes change-color {  
    0% {  
        background: red;          /* Primer fotograma */  
    }  
    50% {  
        background: yellow;     /* Segundo fotograma */  
        width: 400px;  
    }  
    100% {  
        background: green;      /* Último fotograma */  
    }  
}  
  
.animated {  
    background: grey;  
    color: #FFF;  
    width: 150px;  
    height: 150px;  
    animation: change-color 2s ease 0s infinite;  
}
```

Ejemplo de %

@keyframes

En este caso, la animación va a progresar por tres fotogramas clave, por lo que irá desde el color rojo, al color amarillo, para finalizar en el color verde. Esta animación funcionará durante 2 segundos y se realizará una sola vez (por defecto las animaciones solo se realizan una vez, salvo que le indiques infinite o el número deseado)

Funciones de tiempo

Funciones de tiempo

Tanto cuando hablamos de transiciones como cuando hablamos de animaciones tenemos dos propiedades que se aplican exactamente igual y se encargan de definir el ritmo o transcurso de la animación o transición en cuestión.

Funciones de tiempo

Propiedades	Descripción	Valor
transition-timing-function	Ritmo de la transición	FUNCTION cubic-bezier(A, B, C, D)
animation-timing-function	Ritmo de la animación	FUNCTION cubic-bezier(A, B, C, D)

Funciones de tiempo

Función predefinida

Quizás, la forma más sencilla de indicar el ritmo de una animación, sería especificar una de las siguientes palabras clave (valores) que existen para indicar en las propiedades anteriores. Cada una de ella, realiza la animación a un ritmo diferente

Funciones de tiempo

Si no se indica ninguna función de tiempo concreta, CSS utilizará la función de tiempo ease.

Sin embargo, podemos utilizar cualquiera de las anteriores.

Funciones de tiempo

```
.element {  
    width: 200px;  
    height: 50px;  
    margin: 5px;  
    background: indigo;  
    color: white;  
    animation: animated 2s alternate infinite;  
}  
  
@keyframes animated {  
    to { width: 500px; }  
}  
  
.ease { animation-timing-function: ease; }  
.linear { animation-timing-function: linear; }  
.ease-in { animation-timing-function: ease-in; }  
.ease-out { animation-timing-function: ease-out; }  
.ease-in-out { animation-timing-function: ease-in-out; }
```

```
<div class="element ease">ease</div>  
<div class="element linear">linear</div>  
<div class="element ease-in">ease-in</div>  
<div class="element ease-out">ease-out</div>  
<div class="element ease-in-out">ease-in-out</div>
```

Funciones de tiempo

steps

Las funciones de salto permiten establecer una transición o animación en un número concreto de pasos muy específico, que se pasa por parámetro n. En este ejemplo, estamos indicando que la animación debe tener exactamente 5 pasos, por lo que en lugar de ser una transición o animación fluida, será una escalonada en el número de pasos dados

Funciones de tiempo

```
.container {  
    width: 600px;  
    background: grey;  
}  
  
.element {  
    width: 200px;  
    height: 200px;  
    background: indigo;  
    animation: move 5s infinite;  
    animation-timing-function: steps(5);  
}  
  
@keyframes move {  
    to { transform: translateX(500px); }  
}
```

```
<div class="container">  
    <div class="element"></div>  
</div>
```

Ejemplos de librerías

AOS

Animate On Scroll Library

Enlace

Animate.css

Just-add-water CSS animations

Enlace

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez