

pseudoclasses, transformaciones, transiciones y animaciones

pseudoclasses

Las *pseudoclasses* en CSS son palabras clave precedidas por dos puntos (😊) que se añaden a los selectores para aplicar estilos a elementos HTML según su estado, posición o interacción, sin necesidad de modificar el HTML. Permiten, por ejemplo, cambiar el aspecto de un enlace cuando el usuario pasa el cursor sobre él, resaltar el primer elemento de una lista, o dar estilos a campos de formulario según si están activos o validados, facilitando así la creación de interfaces más dinámicas y accesibles.

```
selector:pseudoclase {
  propiedad: valor;
}
```

Pseudoclase	Descripción
:hover	Cuando el cursor está sobre el elemento
:active	Elemento activo (por ejemplo, durante el clic)
:focus	Elemento enfocado (por teclado o clic)
:visited	Enlace visitado
:link	Enlace no visitado
:checked	Elemento de formulario marcado
:disabled	Elemento de formulario deshabilitado
:enabled	Elemento de formulario habilitado
:first-child	Primer hijo de su padre
:last-child	Último hijo de su padre
:nth-child(n)	Selecciona el n-ésimo hijo de su padre, sin importar el tipo de elemento
:nth-of-type(n)	Selecciona el n-ésimo hijo de su padre, pero solo si es del mismo tipo de elemento (por ejemplo, el segundo `` entre varios hijos)
:only-child	Único hijo de su padre
:not(selector)	Elementos que no coinciden con el selector dado
:required	Campo de formulario obligatorio
:optional	Campo de formulario opcional
:valid	Campo de formulario con valor válido

Pseudoclase	Descripción
:invalid	Campo de formulario con valor no válido
:empty	Elemento sin contenido

transform

Las *transformaciones* en CSS permiten modificar la apariencia y posición de los elementos HTML en el espacio, mediante operaciones como mover, girar o escalar. Estas transformaciones se aplican usando la propiedad **transform**, que acepta funciones como **translate**, **rotate** y **scale**. A diferencia de las transiciones, las transformaciones ocurren de manera inmediata, pero pueden combinarse con transiciones o animaciones para lograr efectos suaves y dinámicos.

Función	Descripción	Ejemplo de valor
translate	Desplaza el elemento en el eje X, Y (y opcionalmente Z)	translate(50px, 20px)
rotate	Rota el elemento en torno a su centro (o un eje 3D)	rotate(45deg), rotateZ(1rad)
scale	Escala el tamaño del elemento en X, Y (y opcionalmente Z)	scale(1.5), scale(2, 0.5)
transform	Propiedad abreviada para aplicar una o varias transformaciones	transform: rotate(30deg) scale(1.2)

transiciones

Las *transiciones* en CSS permiten suavizar los cambios de estilo de los elementos HTML, haciendo que estos pasen de un estado a otro de forma gradual en lugar de instantánea. Por ejemplo, al cambiar el color de fondo de un botón al pasar el cursor, la transición hace que el cambio sea progresivo y visualmente más atractivo. Las transiciones se configuran con propiedades como **transition-property**, **transition-duration**, **transition-timing-function** y **transition-delay**, y se pueden definir de forma abreviada para simplificar el código.

Propiedad	Descripción	Ejemplo de valor
transition-property	Propiedad CSS a la que se aplica la transición	background-color, width
transition-duration	Duración de la transición	0.5s, 2s
transition-timing-function	Curva de velocidad de la transición	ease, linear, ease-in
transition-delay	Tiempo de espera antes de iniciar la transición	0s, 0.3s
transition	Propiedad abreviada para definir todas las anteriores en una línea	all 0.5s ease-in 0.2s

animaciones

Las *animaciones* en CSS permiten crear efectos visuales complejos haciendo que los estilos de un elemento cambien de forma progresiva a lo largo del tiempo. Para definir una animación, se utiliza la regla `@keyframes`, que especifica los estilos en distintos puntos del proceso (por ejemplo, inicio, mitad y final). Luego, se aplica la animación al elemento usando propiedades como `animation-name`, `animation-duration`, `animation-timing-function`, `animation-direction` o `animation-iteration-count`.

Ejemplo básico con keyframes:

```
/* Definición de la animación */
@keyframes mover {
  from {
    transform: translateX(0);
    background-color: red;
  }
  to {
    transform: translateX(100px);
    background-color: blue;
  }
}

/* Aplicación de la animación a un elemento */
.caja {
  width: 100px;
  height: 100px;
  background: red;
  animation-name: mover;
  animation-duration: 2s;
}
```

En este ejemplo, el elemento con clase `.caja` se moverá 100 píxeles a la derecha y cambiará de color de rojo a azul en 2 segundos.

Propiedad	Valor	Descripción
animation-name	none (por defecto)	No se aplica ninguna animación.
	nombre (ej: corre, gira)	Nombre de la animación definida en una regla <code>@keyframes</code> .
	initial	Usa el valor inicial por defecto de la propiedad.
	inherit	Hereda el valor de su elemento padre.
	unset	Resetea la propiedad a su valor heredado o inicial.
animation-duration	0s (por defecto)	La animación no tiene duración (no se ejecuta).

Propiedad	Valor	Descripción
	Ns (ej: 2s, 0.5s, 400ms)	Duración de la animación en segundos (s) o milisegundos (ms).
animation-timing-function	linear	Velocidad constante durante toda la animación.
	ease (por defecto)	Comienza lento, acelera y termina lento.
	ease-in	Comienza lento y acelera.
	ease-out	Comienza rápido y desacelera.
	ease-in-out	Comienza y termina lento, acelera en el medio.
	step-start	Salto brusco al inicio de cada ciclo.
	step-end	Salto brusco al final de cada ciclo.
	steps(N, start/end)	Divide la animación en N pasos discretos.
	cubic-bezier(x1, y1, x2, y2)	Curva personalizada para el ritmo de la animación.
animation-direction	normal (por defecto)	La animación se ejecuta siempre hacia adelante.
	reverse	La animación se ejecuta siempre hacia atrás.
	alternate	Alterna el sentido en cada ciclo (adelante y atrás).
	alternate-reverse	Alterna el sentido en cada ciclo, comenzando hacia atrás.
animation-iteration-count	1 (por defecto)	La animación se ejecuta una sola vez.
	N (ej: 2, 3, 4...)	La animación se repite N veces.
	decimal (ej: 2.5)	La animación se repite parcialmente (ej: 2.5 = dos veces completas y media vez más).
	infinite	La animación se repite infinitamente.
animation	Propiedad abreviada para definir todas las anteriores en una línea	mover 2s linear normal infinite