

CSS3

ALGUNOS NUEVOS SELECTORES

Puedes ver todos los selectores en el siguiente enlace:

https://www.w3schools.com/cssref/css_selectors.asp

`elemento::before`

Añade un elemento vacío de línea justo antes del elemento. Normalmente es para añadirle contenido con la propiedad `content`.

`elemento::after`

Lo mismo pero después del elemento.

`elemento[atributo^="valor"]`

Selecciona los elementos con ese atributo cuyo valor comienza por la cadena de texto indicada en valor.

`elemento[atributo$="valor"]`

Selecciona los elementos con ese atributo cuyo valor termina por la cadena de texto indicada en valor.

`elemento[atributo*="valor"]`

Selecciona los elementos con ese atributo cuyo valor contenga la cadena de texto indicada en valor.

Ver: **`ejemplo1css3.html`**

PROPIEDAD BOX-SIZING

Incluye en el alto y ancho de un elemento su padding y su borde.

`box-sizing: border-box;`

Si quiero aplicar esta propiedad a todos los elementos:

*{

Box-sizing: border-box}

Ver: [ejemplobox-sizing](#)

TEXTO EN COLUMNAS (No lo vemos)

Para formatear el texto en columnas se utilizan los siguientes atributos:

column-count: Indica el número de columnas.

column-width: Ancho de cada columna.

column-gap: Separación entre columnas.

column-rule: Crea una línea de separación entre columnas (se indica ancho, estilo y color)



Ver: [ejemplo2css3.html](#)

TRANSFORMACIONES 2D

Para modificar la rotación, inclinación o escala de un elemento.

- Girar un elemento en sentido horario un número de grados:

```
/*transform: rotate(grados);*/
```

```
transform: rotate(25deg)
```

- Inclinar un elemento en coordenadas X como Y.

```
/*transform: skew(gradosX, gradosY);*/
```

```
transform: skew(15deg, 3deg);
```

- Escalar nuestro elemento en X como en Y en tantos por uno:

```
/*transform: scale(escalaX, escalaY);*/
```

```
transform: scale(1.5, 0.6);
```

- Desplazar el elemento tanto en X como en Y:

```
/*transform: translate(desplazamientoX, desplazamientoY);*/
```

```
transform: translate(12px, 19px);
```

Se pueden aplicar diferentes transformaciones a un mismo elemento simplemente escribiéndolas de manera consecutiva:

Ejemplo:

```
transform: scale(1.6) skew(10deg) translate(5px) rotate(12deg);
```

Ver [ejemplo3css3.html](#)

TRANSFORMACIONES 3D

Lo usamos más adelante con las animaciones.

https://www.w3schools.com/css/css3_3dtransforms.asp

COLOR

Podemos añadir un canal alfa a los colores rgb utilizados en css, de esta forma los colores se representan usando **rgba** con 4 parámetros en vez de rgb con tres. Este cuarto parámetro que se añade indica el grado de opacidad de 0 a 1 (0 es transparente, 1 es opacidad total):

Ejemplo: El color de fondo tiene una opacidad intermedia.

```
Body { background-color: rgba ( 100, 200, 40, 0.5); }
```

Ver **ejemplo4css3.html**.

También se puede usar el atributo **opacity** con valores del 0 a 1, la diferencia con lo anterior es que afecta a todo lo incluido dentro del elemento:

Ejemplo: Toda la página tiene una opacidad del 0.5

```
Body{ background-color:rgb( 100, 200, 40); opacity:0.5; }
```

Ver **ejemplo5css3.html**

GRADIENTE

Te permite definir transiciones entre uno o más colores.

Puede ser sólo de transparencia.

Ver **ejemplogradientetrans.html**

Puede ser lineal o diagonal.

Ver **ejemplolinealdiagonal.html**

TRANSICIONES

Consiste en que cuando cambie una propiedad css o más de una de un elemento, se realice de forma gradual, en un tiempo determinado.

Ver **ejemplotransition.html**

ANIMACIONES

Consiste en cambiar las propiedades de los objetos en el tiempo.

Fotograma clave: Es un punto destacado en el tiempo de nuestra animación. Cualquier animación consta al menos de dos fotogramas claves: el punto inicial y el punto final.

En CSS3 se crean animaciones completas mediante **@keyframes**, que son un conjunto de fotogramas clave.

Un ejemplo de animación que desplaza un elemento 30 px hacia la derecha.

```
@keyframe mianimacion{
```

```
    /* fotograma inicial , también 0% */
```

```
    from{
```

```
        left:0px;
```

```
    }
```

```
    /* fotograma final, también 100% */
```

```
    to{
```

```
        left:30px;
```

```
    }
```

```
}
```

Podemos establecer fotogramas claves intermedios para hacer la animación más compleja mediante porcentajes:

```
@keyframe mianimacion{
```

```
    /* fotograma inicial */
```

```
    from{
```

```
        left:0px;
```

```
    }
```

```
    /* fotograma al 50% de su reproducción */
```

```
    50%{
```

```
        left:10px;
```

```
    }
```

```
    /* fotograma final */
```

```
    to{
```

```
        left:30px;
```

}

}

Podemos modificar varias propiedades en cada fotograma, separándolas por punto y coma:

@keyframe mianimacion{

/* fotograma inicial , también 0% */

from{

left:0px; background-color:red;

}

/* fotograma final, también 100% */

to{

left:30px;background-color:blue;

}

}

Una vez creada una animación se la aplicamos a un elemento determinado de la siguiente forma:

#elemento{

animation-name: mianimacion; /*nombre de la animación, se puede definir más de una separándolas por comas */

animation-duration: 3s; /*duración en segundos */

animation-iteration-count: 1; /*número de veces que se repite, infinite si es indefinidamente */

animation-direction: normal; /* Hacia adelante. Con el valor alternate después se vuelve a reproducir en sentido opuesto.*/

animation-delay: 2s; /*Indica en segundos si se produce un retardo en el inicio de la animación */

animation-timing-function:ease; /* Efecto de suavizado (ease, linear, ease-in, ease-out, ease-in-out)*/

position:relative; /*obligatorio que el posicionamiento sea relativo si la animación consiste en realizar un movimiento*/

animation-fill-mode:none; /*La animación no añade ningún estilo al elemento. Otros valores: forwards(el elemento se queda con el estilo del último fotograma), backwards(el elemento se queda con el estilo del primer fotograma reteniéndole durante el delay), both(forwards y backwards a la vez) */

}

Ver [ejemplo6css3.html](#), [ejemplo7css3.html](#) y [ejemplo8css3.html](#).