



Carmelo José Jaén Díaz



Resumen de propiedades CSS avanzado

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Ene - Feb
 - iPasen - cjaedia071@g.educaand.es

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- **Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.**
- **Utilizar marcos y tablas para presentar la información de manera ordenada.**
- **Identificar nuevos elementos y etiquetas en HTML5.**
- **Reconocer las posibilidades de modificar etiquetas HTML.**
- **Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.**

GLOSARIO



Formularios. Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

Fuentes seguras. Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

Guías de estilo. Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

HTML. Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

GLOSARIO



HTML5. Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

Legibilidad. Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

Marcos. Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

Tipografía. Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa "modo o estilo en que está impreso un texto" o "clase de tipos de imprenta".

INTRODUCCIÓN



A continuación, se presenta una recopilación de CSS en forma de tabla resumen con sus propiedades y valores correspondientes.



DEFINICIÓN DE VARIABLES CSS O PROPIEDADES PERSONALIZADAS



Para definir una propiedad personalizada pondremos dos guiones `--` delante del nombre que queramos utilizar. Para definir la propiedad personalizada para todo el documento HTML la incluiremos en el pseudoelemento `:root`. Ejemplo:

```
:root {  
  --principal-color: black;  
  /* Variable personalizada --principal-color */  
  --font: "Arial";  
}
```

USO DE PROPIEDADES PERSONALIZADAS



Para utilizar una propiedad personalizada hay que insertar su nombre dentro de la expresión `var()`:

```
.clase {  
  background-color: var(--principal-color);  
  font-family: var(--font);  
}
```


GRADIENTES CSS



Propiedad	Valores	Ejemplo
background	<i>linear-gradient</i> (dirección{ <i>to top</i> , <i>to bottom</i> , <i>to left</i> y <i>to right</i> } ángulo, color, color, color...)	background: linear-gradient(to right, red, orange);
background	<i>linear-gradient</i> (dirección{ <i>to top</i> , <i>to bottom</i> , <i>to left</i> y <i>to right</i> } ángulo, color posición, color posición, color posición...)	background: linear-gradient(to right, red 20%, orange 40%);
background	<i>linear-gradient</i> (color, color, color...)	background: linear-gradient(red, orange);
background	<i>radial-gradient</i> (forma{ <i>circle</i> y <i>ellipse</i> }, color, color, color...)	background: radial-gradient(circle closest-side, red, orange);

GRADIENTES CSS



Propiedad	Valores	Ejemplo
background	<i>radial-gradient(forma{circle y ellipse}, color posición, color posición, color posición...)</i>	<code>background: radial-gradient(circle at top left, red 20%, orange 40%);</code>
background	<i>radial-gradient(color, color, color...)</i>	<code>background: radial-gradient(red, orange, blue);</code>
background	<i>radial-gradient(farthest-side, color, color, ...)</i>	<code>background: radial-gradient(farthest- side, red, blue);</code>

TRANSICIONES EN CSS: SINTAXIS



Todos los parámetros para aplicar las transiciones se pueden establecer en una sola línea y también mediante propiedades por separado. Veamos cómo se implementa en una sola línea mediante la propiedad abreviada `transition`.

Formato:

`transition:[propiedad a modificar] [Duración] [Tipo de animación] [Retardo];`

TRANSFORMACIONES CSS



Propiedad	Descripción	Ejemplo
Scale	Se establece con uno o dos valores, que representan la cantidad de escala que se aplica en cada dirección: <code>scale(x)</code> o <code>scale(x, y)</code> . Cuando el valor está fuera del rango <code>[-1, 1]</code> , el elemento crece a lo largo de esa dimensión. Cuando está dentro del rango el elemento se encoge.	<code>transform: scale(0.5)</code>
Translate	Cambia la posición del elemento hacia la izquierda, derecha, arriba o abajo. La función <code>translate()</code> se establece con uno o dos valores: <code>translate(x)</code> o <code>translate(x, y)</code> . Sus valores pueden estar definidos en píxeles, porcentajes,...	<code>transform: translate(10px)</code>

TRANSFORMACIONES CSS



Propiedad	Descripción	Ejemplo
Rotate	Gira o rota los elementos en grados: <code>rotate(v)</code> .	<code>transform: rotate(45deg)</code>
Skew	Distorsiona los elementos según el ángulo en grados. La función <code>skew()</code> se establece con uno o dos valores: <code>skew(x)</code> o <code>skew(x, y)</code> .	<code>transform: skew(45deg)</code>
Matrix	Mueve o transforma los elementos con precisión de píxel. La función <code>matrix()</code> se establece con seis valores numéricos: <code>matrix(a, b, c, d, x, y)</code> . Los dos últimos valores representan la translación y los primeros la transformación lineal.	<code>transform: matrix(0.5, 0.1, 0.5, 1, 10, -2)</code>

TRANSFORMACIONES CSS



La propiedad `transform` se usa junto con la propiedad transition vista en la sección anterior para que la transformación pueda tener una transición espaciada en el tiempo:

```
.caja1{
  -webkit-transition: 1s linear;
  transition: 1s linear;
}
.caja1:hover{
  -webkit-transform: scale(.5);
  transform: scale(.5);
}
```

Recuerda utilizar la extensión que te facilita la
tarea de crear los prefijos para navegadores

ANIMACIONES CSS



La propiedad `animation` engloba todas las subpropiedades relacionadas con la animación en CSS, por lo que sería la propiedad abreviada. Las subpropiedades permiten controlar aspectos específicos de la animación, como su nombre, duración, dirección, relleno, etc.

```
selector {  
  animation: name duration timing-function delay iteration-count direction  
  fill-mode play-state;  
}
```

- `name`: Especifica el nombre de la regla `@keyframes` que describe los fotogramas de la animación.
- `duration`: Indica la cantidad de tiempo que la animación consume en completar su ciclo. Se expresa en segundos (s) o milisegundos (ms).

ANIMACIONES CSS



```
selector {  
  animation: name duration timing-function delay iteration-count direction  
  fill-mode play-state;  
}
```

- **timing-function**: Define el ritmo de la animación, es decir, cómo se muestran los fotogramas a lo largo del tiempo. Puede ser lineal, ease, ease-in, ease-out, ease-in-out, cubic-bezier, step-start, step-end, steps, etc.
- **delay**: Especifica un tiempo de espera antes de que la animación comience a ejecutarse. Se expresa en segundos (s) o milisegundos (ms).
- **iteration-count**: Indica el número de veces que se repetirá la animación. Puede ser un número entero, infinite para que se repita infinitamente, o un valor específico como 2, 3, etc.

ANIMACIONES CSS



```
selector {  
  animation: name duration timing-function delay iteration-count direction  
  fill-mode play-state;  
}
```

- **direction**: Define si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia. Puede ser normal, reverse, alternate, o alternate-reverse.
- **fill-mode**: Especifica qué valores tendrán las propiedades después de finalizar la animación. Puede ser none, forwards, backwards, o both.
- **play-state**: Permite pausar y reanudar la secuencia de la animación. Puede ser running o paused.

ANIMACIONES CSS

Subpropiedades de *animation*



Si optamos por usar las subpropiedades, comenzamos añadiendo la propiedad `animation-name` al selector, a la cual le asignamos un nombre identificativo, denominado `identifier`. Luego, podemos utilizar las demás subpropiedades para ajustar diferentes aspectos de la animación, como la duración, dirección, entre otros.

Sintaxis:

```
selector{  
    animation-name: identifier;  
    subproperty: value;  
}
```

ANIMACIONES CSS

Subpropiedades de *animation*



Las subpropiedades de `animation` son:

- `animation-name`: Especifica el nombre de la regla `@keyframes` que describe los fotogramas de la animación.
- `animation-delay`: Tiempo de retardo entre el momento en que el elemento se carga y el comienzo de la secuencia de la animación.
- `animation-direction`: Indica si la animación debe retroceder hasta el fotograma de inicio al finalizar la secuencia.
- `animation-duration`: Indica la cantidad de tiempo que la animación consume en completar su ciclo (duración).
- `animation-iteration-count`: El número de veces que se repite. Podemos indicar `infinite` para repetir la animación indefinidamente.

ANIMACIONES CSS

Subpropiedades de *animation*



Las subpropiedades de `animation` son:

- `animation-play-state`: Permite pausar y reanudar la secuencia de la animación.
- `animation-timing-function`: Indica el ritmo de la animación, es decir, como se muestran los fotogramas de la animación, estableciendo curvas de aceleración.
- `animation-fill-mode`: Especifica qué valores tendrán las propiedades después de finalizar la animación (los de antes de ejecutarla, los del último fotograma de la animación o ambos).

FUNCIONES MATEMÁTICAS EN CSS



Propiedad	Descripción	Valores Ejemplo
<code>abs()</code>	Devuelve el valor absoluto de un número	<code>abs(-10) = 10</code>
<code>sin()</code>	Devuelve el seno de un ángulo	<code>sin(45deg) ≈ 0.71</code>
<code>cos()</code>	Devuelve el coseno de un ángulo	<code>cos(60deg) = 0.5</code>
<code>tan()</code>	Devuelve la tangente de un ángulo	<code>tan(30deg) ≈ 0.58</code>
<code>sqrt()</code>	Devuelve la raíz cuadrada de un número	<code>sqrt(16) = 4</code>
<code>pow()</code>	Eleva un número a una potencia	<code>pow(2, 3) = 8</code>
<code>min()</code>	Devuelve el valor mínimo entre dos números	<code>min(10, 5) = 5</code>
<code>max()</code>	Devuelve el valor máximo entre dos números	<code>max(10, 5) = 10</code>

FUNCIONES MATEMÁTICAS EN CSS



Propiedad	Descripción	Valores Ejemplo
<code>random()</code>	Devuelve un número aleatorio entre 0 y 1	<code>random()</code> \approx 0.71
<code>floor()</code>	Devuelve el valor entero menor o igual	<code>floor(3.8)</code> = 3
<code>ceil()</code>	Devuelve el valor entero mayor o igual	<code>ceil(3.2)</code> = 4
<code>round()</code>	Devuelve el valor redondeado al entero más cercano	<code>round(3.7)</code> = 4
<code>clamp()</code>	Limita un valor dentro de un rango	<code>clamp(10, 5, 8)</code> = 8
<code>calc()</code>	Realiza cálculos matemáticos en propiedades CSS	<code>width: calc(100% - 20px)</code>
<code>attr()</code>	Obtiene el valor de un atributo HTML y lo utiliza en CSS	<code>content: attr(data-text)</code>

CSS Grid



Propiedad	Descripción	Valores Ejemplo
display	Define el contenedor como un grid container	<code>grid</code> (por defecto), <code>inline-grid</code>
grid-template-columns	Define las columnas del grid	<code>grid-template-columns: 1fr 2fr 1fr</code>
grid-template-rows	Define las filas del grid	<code>grid-template-rows: 100px 200px</code>
grid-gap	Propiedad abreviada para establecer <code>grid-column-gap</code> y <code>grid-row-gap</code>	<code>grid-gap: 10px</code>
grid-column-gap	Establece el tamaño del espacio entre columnas	<code>grid-column-gap: 20px</code>
grid-row-gap	Establece el tamaño del espacio entre filas	<code>grid-row-gap: 10px</code>

CSS Clamp()



La función CSS `clamp()` nos permite definir un rango de valores para el tamaño de la fuente, incluyendo un tamaño mínimo, máximo y un valor preferido que se ajusta dinámicamente en relación con el ancho de la pantalla.

Sintaxis: Su sintaxis básica consiste en tres valores separados por comas.

propiedad: `clamp(valor_mínimo, valor_preferido, valor_máximo);`

- **propiedad:** Es la propiedad a la que se va a aplicar `clamp()`. En el caso de trabajar con tipografías usaremos `font-size` para modificar el tamaño de las fuentes.

CSS Clamp()



propiedad: `clamp(valor_mínimo, valor_preferido, valor_máximo);`

- **valor_mínimo:** Es el tamaño mínimo que se aplicará a la propiedad, independientemente del tamaño de la pantalla o del contenedor. Si el valor preferido es menor que este, se aplicará el valor mínimo.
- **valor_preferido:** Es el valor que se aplicará de manera predeterminada a la propiedad. Este valor puede ser relativo, como en el caso de unidades como `vw` (viewport width) o `em`, lo que permite que el tamaño del texto se ajuste dinámicamente en función del tamaño de la pantalla.
- **valor_máximo:** Es el tamaño máximo que se aplicará a la propiedad. Si el valor preferido es mayor que este, se aplicará el valor máximo.

CSS *line-clamp*



```
.elemento-truncado {  
  display: -webkit-box;  
  -webkit-line-clamp: <numero_de_lineas>;  
  -webkit-box-orient: horizontal | vertical;  
  overflow: hidden;  
}
```

- `display: -webkit-box;` Esta propiedad se utiliza para establecer el tipo de caja de diseño utilizado para el elemento. En este caso, `-webkit-box` es necesario para compatibilidad con navegadores basados en WebKit, como Safari y Chrome.
- `-webkit-line-clamp: <numero_de_lineas>;` Esta es la parte principal de la propiedad `line-clamp`. Aquí se especifica el número de líneas que se mostrarán antes de truncar el texto. Por ejemplo, si estableces `-webkit-line-clamp: 3;`, solo se mostrarán las primeras tres líneas del texto y el resto será truncado.

CSS *line-clamp*



- `-webkit-box-orient: horizontal | vertical;` Esta propiedad se utiliza para establecer la orientación del contenido dentro del contenedor. En el contexto de `line-clamp`, normalmente se usa `vertical` para que los elementos de línea se apilen verticalmente dentro del contenedor flexible, permitiendo así la truncación vertical del texto.
 - `horizontal`: Los elementos se apilan horizontalmente.
 - `vertical`: Los elementos se apilan verticalmente.
- `overflow: hidden;` Esta propiedad se utiliza para ocultar el contenido que se desborda del contenedor. Es importante para asegurarse de que el texto truncado no sea visible fuera del área designada.

CSS *Scroll Snap*



Para implementar **Scroll Snap** en CSS, aplicaremos la propiedad **scroll-snap-type** al contenedor correspondiente con la siguiente sintaxis.

SINTAXIS

```
.contenedor-scroll {  
    scroll-snap-type: [x | y | block | inline] [mandatory | proximity];  
}  
.seccion {  
    scroll-snap-align: [start | center | end | none];  
}
```

CSS *Scroll Snap*



- La propiedad `scroll-snap-type` especifica cómo se debe comportar el desplazamiento dentro del contenedor. Puede definir el tipo de desplazamiento en los ejes horizontal (`x`) o vertical (`y`), así como también en bloque (`block`) o en línea (`inline`). Además, determina si el desplazamiento debe detenerse obligatoriamente en los puntos de anclaje especificados (`mandatory`) o si puede detenerse en el punto más cercano (`proximity`).
- La propiedad `scroll-snap-align`, que se aplica a los elementos hijos del contenedor. Esta propiedad determina cómo se alineará cada elemento con respecto al punto de anclaje. Los valores comunes para `scroll-snap-align` son `start`, `end`, `center`, `none`, etc. Por ejemplo, `scroll-snap-align: start;` alinea el inicio del elemento con el punto de anclaje.

CSS *scroll-behavior*



La propiedad `scroll-behavior` se aplica típicamente al elemento `html` para controlar el comportamiento de desplazamiento en toda la página. Veamos su sintaxis.

SINTAXIS:

```
html{  
    scroll-behavior: auto | smooth;  
}
```

- **auto:** Este es el valor predeterminado. Indica que el desplazamiento será instantáneo, similar al comportamiento estándar del navegador.
- **smooth:** Este valor activa el desplazamiento suave, lo que significa que el movimiento entre los diferentes elementos de la página será gradual y animado.

CSS *Blend Modes*



Para aplicar **Blend Modes** se utiliza la propiedad `mix-blend-mode`, que permite mezclar los colores de un elemento con los colores del fondo o de otros elementos superpuestos. Veamos cómo se implementa esta propiedad.

SINTAXIS

```
.contenedor-superpuesto {  
    mix-blend-mode: <modo-de-mezcla>;  
}
```

Donde `<modo-de-mezcla>` es el valor del **Blend Mode** que quieres aplicar, como `multiply`, `screen`, `overlay`, entre otros.

CSS Blend Modes



Veamos una tabla con los valores más destacados de la propiedad `mix-blend-mode`, su descripción y sus características.

Blend Mode	Descripción	Función
<code>multiply</code>	Multiplica los colores de la capa superior con los de la capa inferior.	Oscurecer la imagen resultante.
<code>screen</code>	Invierte los colores de la capa superior y los multiplica con los de la capa inferior.	Aclarar las áreas de la imagen.
<code>overlay</code>	Combina <code>Multiply</code> y <code>Screen</code> , oscureciendo o aclarando la capa inferior dependiendo de sus colores.	Producir un efecto de contraste y saturación en la imagen resultante.

CSS *Blend Modes*



Blend Mode	Descripción	Función
darken	Conserva los colores más oscuros de las dos capas.	Seleccionar el color más oscuro de cada píxel de ambas capas.
lighten	Conserva los colores más claros de las dos capas.	Seleccionar el color más claro de cada píxel de ambas capas.
color-dodge	Aclara los colores de la capa inferior dependiendo de los colores de la capa superior.	Aclarar las áreas de la imagen de la capa inferior.
color-burn	Oscurece los colores de la capa inferior dependiendo de los colores de la capa superior.	Oscurecer las áreas de la imagen de la capa inferior.
difference	Resta los colores de la capa superior de los de la capa inferior.	Producir un efecto de inversión de colores.

CSS Blend Modes



Blend Mode	Descripción	Función
exclusion	Similar a Difference, pero con una apariencia menos extrema.	Combinar los colores de ambas capas con un efecto de exclusión.
hard-light	Combina Multiply y Screen, pero de una manera más intensa.	Producir un efecto más fuerte de contraste y saturación.
soft-light	Similar a Overlay, pero con una apariencia más suave.	Producir un efecto más suave de contraste y saturación.
hue	Preserva el brillo y la saturación de la capa inferior, pero adopta el tono de la capa superior.	Aplicar el tono de la capa superior a la capa inferior.
saturation	Preserva el brillo y el tono de la capa inferior, pero adopta la saturación de la capa superior.	Aplicar la saturación de la capa superior a la capa inferior.

CSS *Blend Modes*



Blend Mode	Descripción	Función
color	Preserva el brillo de la capa inferior, pero adopta el tono y la saturación de la capa superior.	Aplicar el tono y la saturación de la capa superior a la capa inferior.
luminosity	Preserva el tono y la saturación de la capa inferior, pero adopta el brillo de la capa superior.	Aplicar el brillo de la capa superior a la capa inferior.

PROPIEDAD *hyphens* CSS:

Sintaxis y valores



La propiedad **hyphens** en CSS controla la división de palabras al final de una línea cuando el texto se desborda.

SINTAXIS

```
.elemento {  
    hyphens: auto | manual | none;  
}
```

VALORES

- **none**: Desactiva la división automática y manual de palabras. El texto puede desbordarse si no cabe en el contenedor.

VALORES

- **auto**: El navegador decide dónde dividir las palabras para evitar el desbordamiento del texto. Utiliza las reglas de separación de sílabas del idioma predeterminado del documento o del sistema operativo.
- **manual**: Las palabras no se dividirán automáticamente. Debes utilizar el carácter de guión (-) para indicar dónde se pueden dividir las palabras manualmente.

FILTROS CSS



La propiedad `filter` de CSS permite aplicar efectos visuales a elementos de la página. Estos efectos se aplican mediante funciones de filtro, que pueden tener uno o más valores según el efecto deseado. Veamos la sintaxis de la propiedad `filter` en CSS y los valores que acepta.

SINTAXIS

```
selector {  
  filter: <función-de-filtro>(<valor>) ;  
}
```

FILTROS CSS



Valores más comunes que acepta la propiedad `filter`:

Función	Descripción	Valores
<code>blur(<valor>)</code>	Aplica un desenfoque al elemento.	<code><valor></code> cantidad de desenfoque a aplicar.
<code>brightness(<valor>)</code>	Ajusta el brillo del elemento.	<code><valor></code> 0: completamente oscuro, 1: original.
<code>contrast(<valor>)</code>	Ajusta el contraste del elemento.	<code><valor></code> 0: completamente gris, 1: original.
<code>grayscale(<valor>)</code>	Convierte el elemento a escala de grises.	<code><valor></code> 0: original, 1: escala de grises.

FILTROS CSS



Función	Descripción	Valores
hue-rotate(<valor>)	Gira el espectro de colores del elemento.	<valor> es la cantidad de grados en sentido horario 0 a 360deg.
invert(<valor>)	Invierte los colores del elemento.	<valor> 0: original, 1: invertido.
opacity(<valor>)	Ajusta la opacidad del elemento.	<valor> 0: transparente, 1: original.
saturate(<valor>)	Ajusta la saturación del elemento.	<valor> 0: grisáceo , 100%: original.
sepia(<valor>)	Aplica un tono sepia al elemento.	<valor> 0: original, 1: sepia completo.

PREFIJOS CSS DE LOS NAVEGADORES



Prefijo	Navegador
-moz-	Firefox
-webkit-	Safari y Chrome
-o-	Opera
-khtml-	Konqueror
-ms-	Internet Explorer y Microsoft Edge