



Carmelo José Jaén Díaz



CSS Clamp: Textos según el tamaño de pantalla

-
- 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.

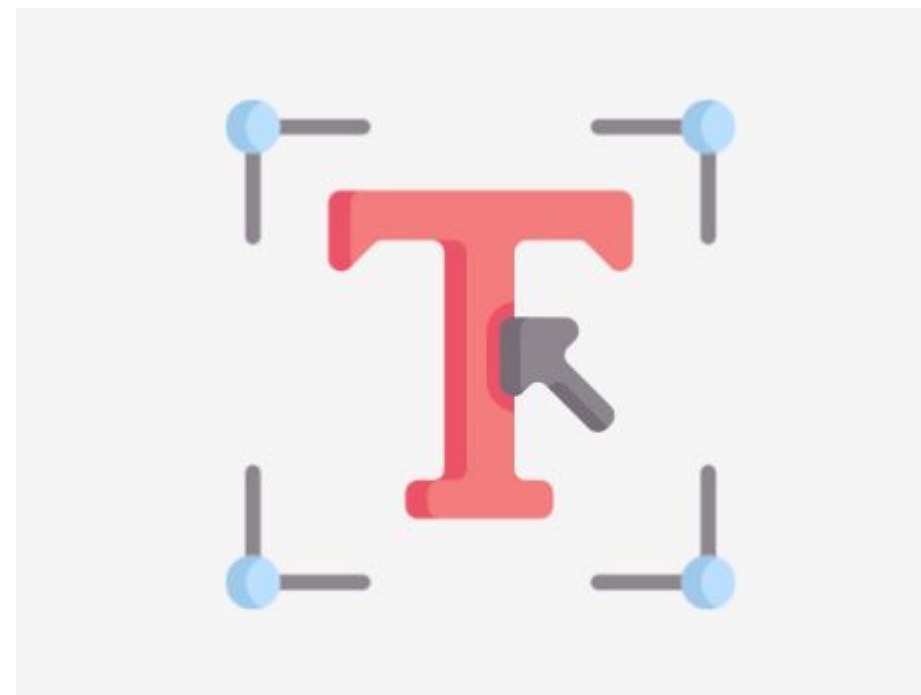
Marcos. Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

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 los textos responsivos que se adaptan al tamaño de pantalla se les conoce como **tipografía fluida** o **tipografía *responsive***. En el diseño web, la **tipografía fluida** nos ayuda a comunicarnos de forma efectiva con los usuarios. Garantizar la legibilidad del texto en diferentes dispositivos es esencial para una buena experiencia de usuario. En esta lección, aprenderemos a crear textos que se adaptan al tamaño de pantalla mediante CSS **`clamp()`**, eliminando la necesidad de múltiples *media queries*.



TIPOGRAFÍA FLUIDA VS *media queries*



Históricamente, en el diseño web hemos utilizado media queries para ajustar el tamaño de la fuente en función del ancho de la pantalla del dispositivo. Sin embargo, esto puede resultar en transiciones bruscas entre diferentes tamaños de fuente en los puntos de interrupción definidos. La tipografía fluida aborda este problema ajustando dinámicamente el tamaño de la fuente según el ancho de la pantalla, lo que crea una experiencia de lectura más uniforme y agradable para los usuarios.

UNIDADES DE MEDIDA RELATIVAS PARA LOGRAR TEXTOS *RESPONSIVE*



Para lograr una tipografía *responsive* es conveniente utilizar unidades de medida relativas, como **em** o **vh** y **vw**, en lugar de unidades absolutas, como píxeles (**px**). Las unidades relativas se ajustan dinámicamente según el tamaño de la pantalla del dispositivo, lo que garantiza que el texto se vea bien en cualquier contexto.

Por ejemplo, podemos usar la unidad relativa **vw** (viewport width) junto con CSS Clamp. De este modo, si definimos el tamaño de fuente de un elemento como **font-size: 5vw;**, el tamaño de la fuente será el 5% del ancho total de la pantalla.

USO DE CSS *Clamp* PARA CONSEGUIR TEXTOS QUE SE ADAPTEN AL TAMAÑO DE PANTALLA



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.

USO DE CSS *Clamp* PARA CONSEGUIR TEXTOS QUE SE ADAPTEN AL TAMAÑO DE PANTALLA



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.

USO DE CSS *Clamp* PARA CONSEGUIR TEXTOS QUE SE ADAPTEN AL TAMAÑO DE PANTALLA



EJEMPLO

En el siguiente ejemplo, el tamaño de la fuente se ajusta entre 1em y 2.5em, con un tamaño preferido que varía en función del ancho de la pantalla. En el tamaño preferido se usa la unidad de medida relativa vw o viewport-width, que no es más que el ancho de la pantalla o viewport.

El valor 100vw es el 100% del ancho de la pantalla actual. Por lo que el tamaño de nuestra fuente preferida va a ser un porcentaje del ancho en pixeles de la pantalla. Por ejemplo, si decimos que nuestro tamaño preferido es de 3vw, cuando el ancho de la pantalla o viewport sea de 1000px, la fuente tendrá un tamaño de 30px. Si el ancho de la pantalla es de 700px, el tamaño de la fuente será de 21px.

```
h2 {  
    font-size: clamp(1em, 3vw, 2.5em);  
}
```

USO DE CSS *Clamp* PARA CONSEGUIR TEXTOS QUE SE ADAPTEN AL TAMAÑO DE PANTALLA



Código HTML:

```
<h2>CSS clamp() es la clave para  
implementar la tipografía fluida</h2>
```

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/bNbYx>

Lp

CSS clamp() es la clave para implementar la tipografía fluida

Código CSS:

```
h2 {  
  font-size: clamp(1em, 3vw, 2.5em);  
}  
  
@supports not (font-size: clamp(1em, 3vw,  
2.5em)) {  
  /* Alternativa para navegadores que no  
  admiten clamp() */  
  font-size: min(max(1rem, 3vw), 2.5em);  
}
```

CSS *calc()* PARA MEJORAR LA TIPOGRAFÍA FLUIDA



Otra técnica para mejorar la tipografía responsive es el uso de la función `calc()` en CSS, que permite realizar cálculos matemáticos para determinar el tamaño de la fuente. Por ejemplo, podemos combinar `calc()` con unidades relativas.

En el siguiente ejemplo, el tamaño de la fuente será de al menos 16 píxeles, pero también aumentará en un 0.5% del ancho de la pantalla.

```
font-size: calc(16px + 0.5vw);
```

CSS clamp() JUNTO CON calc() PARA CÁLCULOS MATEMÁTICOS Y ESTABLECER LÍMITES



Es común mezclar el uso de `calc()` y `clamp()` en CSS para crear estilos más flexibles y dinámicos. Por ejemplo, podrías utilizar `calc()` para realizar cálculos matemáticos en propiedades como el ancho, el margen o el padding de un elemento, y luego usar `clamp()` para establecer límites para esas propiedades.

```
/* Usando calc() y clamp() juntos para el tamaño de fuente del texto */  
.texto {  
  font-size: clamp(12px, calc(1em + 3vw), 36px);  
  /* En este ejemplo, el tamaño de la fuente del 'texto' se ajustará  
  dinámicamente entre 12px y 36px, pero nunca será menor que el tamaño base de  
  1em más el 3% del ancho de la ventana (vw). */  
}
```

Esto significa que a medida que el ancho de la ventana aumenta, el tamaño de la fuente del texto también aumentará, proporcionando así un diseño adaptativo.

COMPATIBILIDAD DE CSS *clamp()* EN LOS NAVEGADORES



Como es común en CSS, hay que verificar la compatibilidad de las nuevas propiedades y funciones con los diferentes navegadores para asegurar una experiencia uniforme para todos los usuarios. Al verificar la compatibilidad de la función `clamp()` en sitios como [Can I Use](#), podemos ver que muchos navegadores modernos la admiten. Sin embargo, aún puede haber versiones anteriores de navegadores que no la soporten completamente.

Una forma de abordar la compatibilidad es mediante el uso de feature queries, una técnica que nos permite verificar si un navegador es compatible con una determinada propiedad o función antes de aplicarla. Veamos cómo podríamos implementar esto con la función `clamp()`:

COMPATIBILIDAD DE CSS *clamp()* EN LOS NAVEGADORES



```
h2 {  
  font-size: clamp(1em, 3vw, 2.5em);  
}  
  
@supports not (font-size: clamp(1em, 3vw, 2.5em)) {  
  /* Alternativa para navegadores que no admiten clamp() */  
  font-size: min(max(1rem, 3vw), 2.5em);  
}
```

En este ejemplo, estamos utilizando una *feature query* para verificar si el navegador admite la función `clamp()`. Si no es compatible, aplicamos una alternativa utilizando las funciones `min()` y `max()`. Esta alternativa garantiza que el tamaño de la fuente se ajuste dinámicamente dentro de un rango predefinido, incluso en navegadores que no admiten `clamp()`.

HERRAMIENTAS ONLINE PARA EL USO DE TIPOGRAFÍAS FLUIDAS



Para facilitar el proceso de creación de tipografías fluidas, existen varias herramientas online que permiten a los diseñadores experimentar y ajustar fácilmente el comportamiento de los textos. Veamos un par de herramientas destacadas:

- Fluidity typography editor by Adrian Bece: Esta herramienta te permite insertar los valores máximos y mínimos del tamaño de fuente, así como el tamaño preferido (o *fluid size*) y un tamaño relativo. Lo destacado de esta herramienta es su capacidad para visualizar los cambios mediante gráficos y tablas para comprender rápidamente cómo se ajustará el tamaño de la fuente en función del ancho de la pantalla. Además, proporciona código CSS listo para copiar y pegar, facilitando la implementación de los ajustes realizados.

HERRAMIENTAS ONLINE PARA EL USO DE TIPOGRAFÍAS FLUIDAS



- Min-Max-Value Interpolation by James Gilyead y Trys Mudford: Esta herramienta nos permite introducir valores máximos y mínimos del tamaño de fuente, así como valores máximos y mínimos del *viewport*. Al ajustar estos parámetros, podemos ver cómo se comportará la tipografía en función del tamaño de la pantalla. La herramienta proporciona ejemplos visuales en forma de texto y nos permite jugar con el tamaño del navegador para experimentar con diferentes escenarios de diseño responsivo.

Como has podido ver, la tipografía fluida es una parte fundamental del diseño web moderno. Al utilizar unidades relativas de medida como `vw` y `vh`, junto con técnicas avanzadas como `calc()` y `clamp()`, podemos crear experiencias de lectura agradables y accesibles en cualquier dispositivo sin necesidad de usar *media queries*.