



Unidades vh y vw en CSS

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

Carmelo José Jaén Díaz

———— 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



Tal y como vimos en el artículo sobre las unidades de medida en CSS, las unidades **vh** y **vw** son unidades relativas, por lo que se **ajustan al tamaño del viewport** (área visible del navegador o tamaño de pantalla). En CSS, representan el **porcentaje del área visible del usuario**. En este artículo, veremos qué son las unidades **vh** y **vw**, cómo se utilizan en CSS y ejemplos de aplicación prácticos.



UNIDADES VH Y VW EN CSS ¿QUÉ SON?



Como decíamos anteriormente, las unidades `vh` y `vw` son unidades de medida relativas en CSS que se utilizan para dimensionar elementos en relación con el tamaño del viewport del navegador. Veamos sus características de forma detallada:

- **`vh` (Viewport Height):** Una unidad `vh` representa el 1% del alto del viewport. Por lo tanto, `1vh` es igual al 1% del alto total del viewport del usuario. Esta unidad es útil para dimensionar elementos en función de la altura de la ventana del navegador.
- **`vw` (Viewport Width):** Por otro lado, una unidad `vw` representa el 1% del ancho del viewport. Así, `1vw` es igual al 1% del ancho total del viewport del usuario. Esta unidad es especialmente útil para dimensionar elementos en función del ancho de la ventana del navegador.

CÓMO SE USAN LAS UNIDADES VH Y VW EN CSS



Las unidades vh y vw se suelen usar para dimensionar alturas, anchuras o márgenes. Veamos algunos ejemplos:

- **Dimensionamiento de altura y anchura:** Se pueden especificar valores de altura y anchura utilizando vh y vw, lo que permite crear diseños que se escalan automáticamente según el tamaño de la ventana del navegador.

```
.elemento {  
  height: 50vh; /* La altura del elemento será el 50% del alto del viewport */  
  width: 75vw; /* El ancho del elemento será el 75% del ancho del viewport */  
}
```


CÓMO SE USAN LAS UNIDADES VH Y VW EN CSS



- Espaciado vertical y horizontal: Se pueden definir márgenes y espaciados utilizando vh y vw, lo que garantiza que los elementos mantengan proporciones adecuadas en diferentes tamaños de pantalla.

```
.contenedor {  
    margin-top: 10vh; /* Margen superior del 10% del alto del viewport */  
    margin-right: 5vw; /* Margen derecho del 5% del ancho del viewport */  
}
```

Diferencias entre usar las unidades vh, vw y %



La diferencia principal entre usar vh, vw y % radica en la referencia de dimensionamiento. Mientras que % se basa en el tamaño del elemento contenedor, vh y vw se refieren al tamaño del viewport (área visible del navegador).

Por ejemplo, si un elemento tiene una altura del 50% en relación con su contenedor, esta altura cambiará si el contenedor cambia de tamaño. En cambio, si se establece una altura del 50vh para un elemento, esta altura será siempre el 50% del alto del viewport, independientemente del tamaño del contenedor. Lo mismo se aplica al ancho con vw.

EJEMPLOS PRÁCTICOS DE USO



Las unidades vh y vw son útiles en situaciones donde se requiere diseño responsivo. Algunos ejemplos prácticos de su uso son los siguientes:

CENTRAR UN TEXTO SEGÚN EL TAMAÑO DEL VIEWPORT (CON LINE-HEIGHT)

Uno de los diseños que podemos necesitar es un texto centrado según el tamaño del viewport. Lo ideal para centrar textos en relación con el tamaño del viewport es utilizar la propiedad line-height para establecer la altura de línea del texto en 100vh, lo que hace que el texto se extienda verticalmente para ocupar toda la altura del viewport. Además, se debe definir un contenedor padre con una altura del 100% del viewport (100vh). Ver ejemplo a continuación.

EJEMPLOS PRÁCTICOS DE USO



CENTRAR UN TEXTO SEGÚN EL TAMAÑO DEL VIEWPORT (CON LINE-HEIGHT)

Código HTML:

```
<div class="container">
  <h1>Contenedor de Altura
  100vh</h1>
  <p>Este contenedor ocupa el 100%
  de la altura del viewport.</p>
</div>
```

Código CSS:

```
*{
  margin: 0;
  padding: 0;
}
.container {
  height: 100vh;
  background-color: #f0f0f0;
}
h1, p{
  text-align: center;
  line-height: 100vh;
}
```

EJEMPLOS PRÁCTICOS DE USO



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

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

Contenedor de Altura 100vh

EJEMPLOS PRÁCTICOS DE USO



VH Y VW: CENTRAR CUALQUIER ELEMENTO SEGÚN EL ANCHO DEL VIEWPORT (CON PROPIEDAD TRANSFORM)

El siguiente código lo puedes utilizar para centrar cualquier elemento vertical y horizontalmente en relación con el tamaño del viewport utilizando las unidades vh y vw, junto con la propiedad transform de CSS. Este método es útil para centrar imágenes o contenedores dentro de un div de manera rápida y sencilla.

EJEMPLOS PRÁCTICOS DE USO



VH Y VW: CENTRAR CUALQUIER ELEMENTO SEGÚN EL ANCHO DEL VIEWPORT
(CON PROPIEDAD TRANSFORM)

Código HTML:

```
<div class="contenedor">  
  <div class="centrado">  
    <i class="fas fa-star"></i>  
    <p>Texto Centrado</p>  
  </div>  
</div>
```

EJEMPLOS PRÁCTICOS DE USO



VH Y VW: CENTRAR CUALQUIER ELEMENTO SEGÚN EL ANCHO DEL VIEWPORT (CON PROPIEDAD TRANSFORM)

Código CSS:

```
html, body {  
  margin: 0;  
  padding: 0;  
  height: 100%; /* Asegura que el cuerpo y el  
html ocupen todo el alto del viewport */  
}
```

```
.contenedor {  
  position: relative; /* Establece una  
posición relativa */  
  height: 100vh; /* Utiliza el 100% del alto  
del viewport */  
}
```

```
.centrado {  
  position: absolute; /* Establece una  
posición absoluta */  
  top: 50%; /* Sitúa el elemento en el 50% de  
la altura del contenedor */  
  left: 50%; /* Sitúa el elemento en el 50%  
del ancho del contenedor */  
  transform: translate(-50%, -50%); /* Mueve  
el elemento hacia arriba y hacia la izquierda  
en un 50% de su propio tamaño */  
  text-align: center; /* Alineación del  
texto */  
}
```


EJEMPLOS PRÁCTICOS DE USO



VH Y VW: CENTRAR CUALQUIER ELEMENTO SEGÚN EL ANCHO DEL VIEWPORT
(CON PROPIEDAD TRANSFORM)

Código CSS:

```
.fa-star {  
  font-size: 48px; /* Tamaño del icono */  
  color: gold; /* Color del icono (cambia  
según tus preferencias) */  
}
```



Texto Centrado

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

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

EJEMPLOS PRÁCTICOS DE USO



DISEÑO DE FOOTER FIJO EN LA PARTE INFERIOR DEL VIEWPORT SOLO SI HAY POCO CONTENIDO

El siguiente ejemplo muestra cómo diseñar un footer que acompaña al contenido principal, adaptándose dinámicamente a su longitud. Utilizando flexbox y unidades `vh`, el footer se posiciona al final del viewport si el contenido es corto. Sin embargo, si el contenido en el área principal es más extenso, el footer se desplaza hacia abajo, siguiendo el flujo del contenido.

EJEMPLOS PRÁCTICOS DE USO



DISEÑO DE FOOTER FIJO EN LA PARTE INFERIOR DEL VIEWPORT SOLO SI HAY POCO CONTENIDO

Código HTML:

```
<div class="container">
  <header>
    <h1>Encabezado</h1>
  </header>
  <main>
    <h2>Contenido Principal</h2>
    <p>Este es el contenido principal de la página. Utilizaremos flexbox para organizarlo de manera fluida.</p>
  </main>
  <footer>
    <p>Pie de Página</p>
  </footer>
</div>
```

EJEMPLOS PRÁCTICOS DE USO



DISEÑO DE FOOTER FIJO EN LA PARTE INFERIOR DEL VIEWPORT SOLO SI HAY POCO CONTENIDO

Código CSS:

```
body, html {  
  margin: 0;  
  padding: 0;  
  height: 100%;  
}  
  
.container {  
  display: flex;  
  flex-direction: column;  
  height: 100vh;  
}
```

```
header {  
  background-color: #333;  
  color: white;  
  padding: 20px;  
}  
  
main {  
  flex-grow: 1;  
  padding: 20px;  
}  
  
footer {  
  background-color: #333;  
  color: white;  
  padding: 10px;  
  text-align: center;  
}
```

EJEMPLOS PRÁCTICOS DE USO



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

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

Encabezado

Contenido Principal

Este es el contenido principal de la página. Utilizaremos flexbox para organizarlo de manera fluida.