Medidas relativas

DigitalHouse >
Coding School

Las **medidas relativas** son aquellas que tienen en cuenta el contexto donde se encuentran.
Si el contexto cambia, estas medidas cambiarán con él.

El contexto

Según de qué medida estemos hablando, el número que pongamos será relativo a:

- El contenedor padre.
- El tamaño de la fuente del sitio.
- El tamaño de la fuente del contenedor padre.
- El tamaño del viewport.

Si tomamos los porcentajes como ejemplo, podemos decir que son una **unidad relativa**, puesto que **30%** de ancho no será lo mismo para un elemento situado dentro de un contenedor de 2000px de ancho que para un contenedor de 1000px de ancho.

Índice

1 | Porcentajes

Los porcentajes

Cualquier medida expresada en porcentaje **siempre** estará **relacionada** con la medida (en ese mismo eje) del **elemento padre** que la contiene.

Si el contenedor **padre** mide **300px** de ancho y le asignamos un ancho del **50%** al elemento interior, este medirá **150px** (el 50% del ancho padre).

Ojo: no se recomienda usar porcentajes para el alto de un elemento.

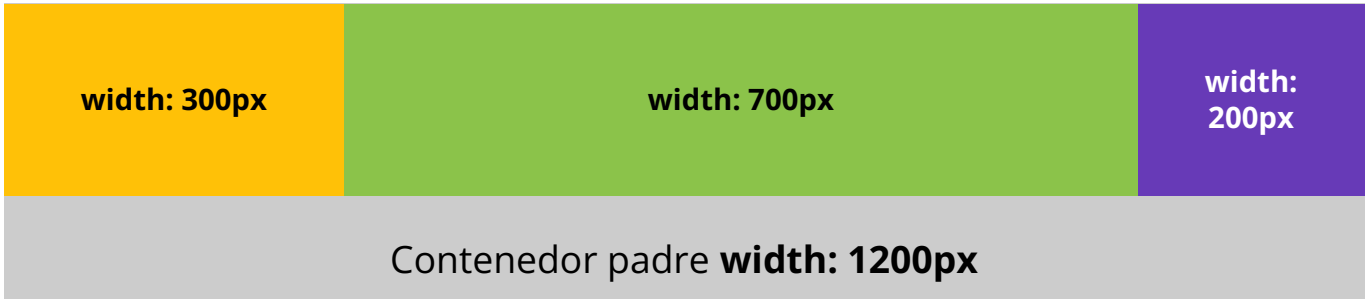
```css
.elementoContenedor { width: 300px }
.elementoInterior { width: 50% } // Será 150px
```

Calculando el porcentaje

Una buena herramienta para calcular o hacer el traslado de píxeles a porcentajes, es la regla de 3 simple.

Para cada elemento del contenedor padre, el cálculo sería:

elemento hijo * 100 / elemento padre = porcentaje



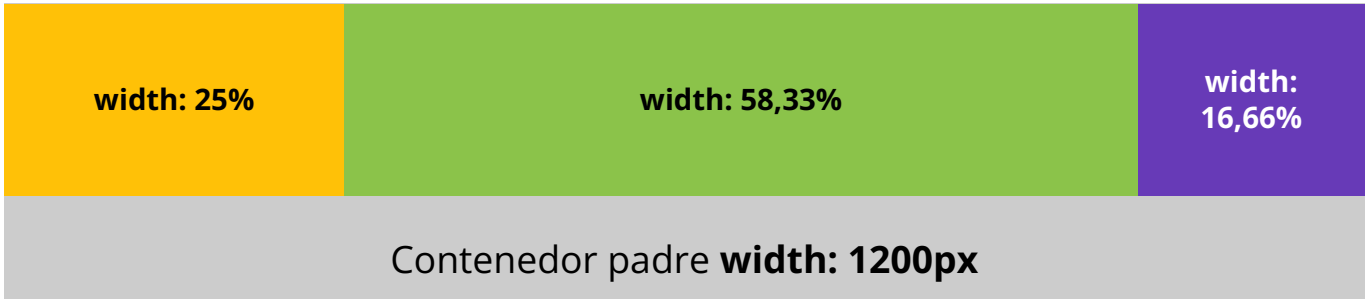
Porcentajes

Entonces, si calculamos cada elemento, nos quedaría así:

300 * 100 / 1200 = **25%**

700 * 100 / 1200 = **58.33%**

200 * 100 / 1200 = **16.66%**



2 | em y rem

Cómo funcionan los ems

Los **ems** son siempre **relativos al elemento padre**. Tomarán como valor de referencia la propiedad **font-size**.

El punto de referencia inicial es el valor de **font-size** del elemento <html>, que por defecto es **16px**. El resto de los elementos tendrán **1em** de **font-size** que equivale al mismo valor que tenga el padre.

Si le asignamos **1.5em** al font-size de cualquier elemento, el tamaño resultante será el valor del padre multiplicado por el valor en ems → **16px * 1.5 = 24px**.

```css
p { font-size: 1.5em } // 16px * 1.5 = 24px
```

Cómo funcionan los ems

Si utilizamos ems en una propiedad que no sea **font-size**, se tomará para el cálculo el font-size que tenga el elemento que estemos modificando.

Por lo general, se utilizan este tipo de unidades para todo lo que sea relacionado a la tipografía, pero también podemos aplicarlo a otras propiedades como margin y padding para que varíen en función de los tamaños de las fuentes.

```css
p {
    font-size: 20px;
    line-height: 2em; // 20px * 2 = 40px
    padding: 1.5em; // 20px * 1.5 = 30px
}
```

Calculando los ems de los elementos

Por defecto `<html>` tendrá **16px** en font-size y los demás elementos **1em**.

Por lo tanto, todos los elementos del sitio tendrán **16px** de font-size.

```
<html>  → font-size: 16px
    <body>  → font-size: 1em → 16px
        <div>  → font-size: 1em → 16px
            <p>  → font-size: 1em → 16px
                <strong>Hola!</strong>  → font-size: 1em → 16px
            </p>
        </div>
    </body>
</html>
```

Calculando los ems de los elementos

Si cambiamos el font-size del `<div>` a **2em**, este tendrá 2 veces el tamaño del font-size de su padre.

Por lo tanto el `<div>` y todos sus hijos tendrán **32px** de font-size.

```
<html>  → font-size: 16px
    <body>  → font-size: 1em → 16px
        <div>  → font-size: 2em → 16px * 2 → 32px
            <p>  → font-size: 1em → 32px
                <strong>Hola!</strong>  → font-size: 1em → 32px
            </p>
        </div>
    </body>
</html>
```

Calculando los ems de los elementos

Teniendo en cuenta lo anterior, si a `<p>` le asignamos un **font-size** de **1.5em**, el cálculo se hará en base al tamaño de su padre, el `<div>`: 32px * 1.5 = 48px.

Nuevamente, todos los hijos de `<p>` heredarán ese tamaño de fuente.

```
<html>  → font-size: 16px
    <body>  → font-size: 1em
        <div>  → font-size: 2em → 32px
            <p>  → font-size: 1.5em → 32px * 1.5 → 48px
                <strong>Hola!</strong>  → font-size: 1em → 48px
            </p>
        </div>
    </body>
</html>
```

Calculando los ems de los elementos

Si ahora cambiamos el font-size del elemento `<html>` a **10px**, todos los elementos cambiarán de tamaño sin haber modificado directamente su font-size.

```
<html>  → font-size: 10px
    <body>  → font-size: 1em → 10px
        <div>  → font-size: 2em → 20px
            <p>  → font-size: 1.5em → 30px
                <strong>Hola!</strong>  → font-size: 1em → 30px
            </p>
        </div>
    </body>
</html>
```

Cómo funcionan los rems

RECOMENDADO

Como te habrás podido dar cuenta usar ems puede ser muy complicado.

Los **rems** funcionan muy parecido a los ems, con la diferencia de que siempre tomarán de base el tamaño de font size del elemento `<html>`.

Eso quiere decir que el tamaño expresado en rems no modificará el de los elementos hijos y tampoco se verá afectado por el del elemento padre.

Por lo general es mejor usar rems en lugar de ems ya que conservamos las ventajas de una unidad relativa, pero nos evitamos hacer cálculos complejos y estar pendientes de cómo se afectan los elementos entre sí.

```css
p { font-size: 1.5rem }
```

Calculando los rems de los elementos

Teniendo en cuenta lo anterior, podemos asignar distintos valores a los elementos sin que estos afecten a los otros. Todos los elementos tomarán de base el tamaño de font-size del elemento `<html>`.

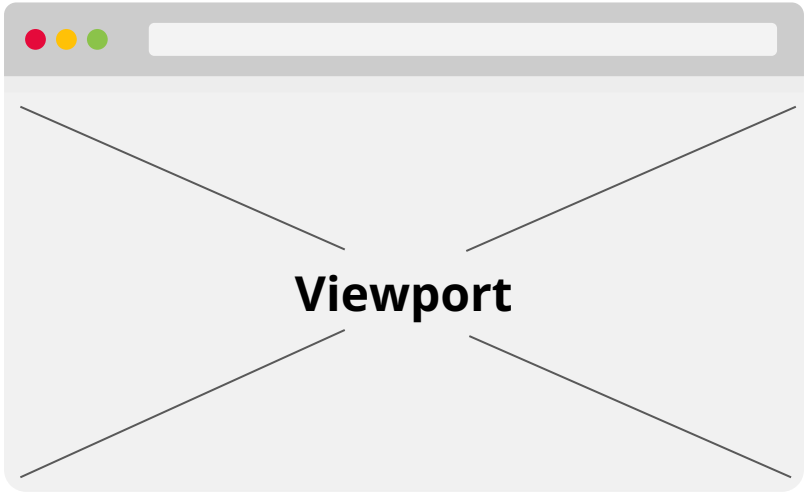
```
<html>  → font-size: 16px
    <body>  → font-size: 2rem → 16px * 2 → 32px
        <div>  → font-size: 3rem → 16px * 3 → 48px
            <p>  → font-size: 1rem → 16px
                <strong>Hola!</strong>  → font-size: 0.5rem → 16px * 0.5 → 8px
            </p>
        </div>
    </body>
</html>
```

3 | vw y vh

Medidas de viewport

El viewport es el espacio visible que tiene el navegador para mostrar el sitio.

Eso quiere decir que se pueden utilizar medidas relativas a este espacio para poder determinar el tamaño de ciertos elementos.



Viewport width y viewport height

vw o viewport width es relativo al ancho total del viewport.

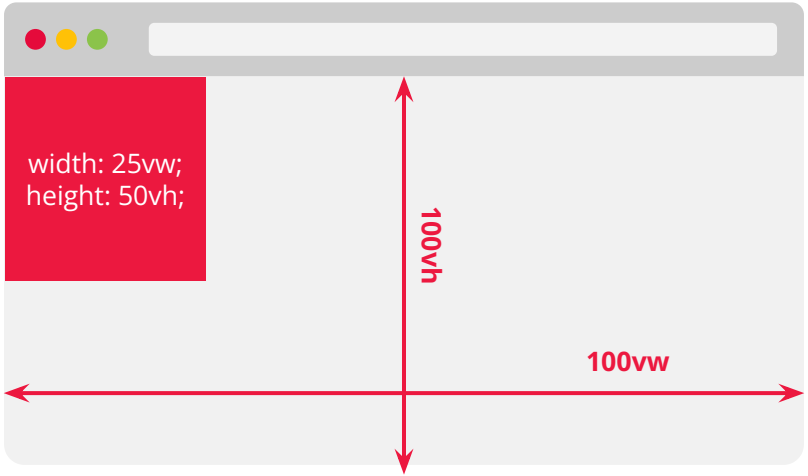
vh o viewport height es relativo al alto total del viewport.

Ambas medidas están expresadas como porcentajes del total, eso quiere decir que **50vh** será equivalente al 50% del alto disponible en el viewport.

```css
div {
    width: 25vw; // 25% del ancho disponible
    height: 50vh; // 50% del alto disponible
}
```

Visualizando vh y vw

Cualquier medida expresada en viewport width (vw) o viewport height (vh) tomará **siempre** como referencia al viewport del documento.




DigitalHouse>
Coding School