



# Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web  
IES Playamar

## 04-03b.- Unidades Css absolutas, Relativas y por ViewPort

|           |  |       |  |
|-----------|--|-------|--|
| Nombre    |  | Curso |  |
| Apellidos |  | Fecha |  |

## Unidades Css absolutas, Relativas y por ViewPort

### Unidades absolutas

¿Qué son las unidades absolutas?

Las unidades absolutas son un tipo de medida fija que no cambia, que no depende de ningún otro factor. Son ideales en contextos donde las medidas no varían como pueden ser en medios impresos (documentos, impresiones, etc...), pero son unidades poco adecuadas para la web moderna, donde necesitamos adaptarnos a diferentes resoluciones o pantallas.

No obstante, para comenzar, existe una unidad absoluta recomendable, que nos puede servir para dar nuestros primeros pasos en el Desarrollo web y ayudarnos a afianzar conocimientos, y así en el futuro, progresar y aprender de forma más rápida otras unidades CSS.

### La unidad **px**

La unidad **px** (**píxel**) es probablemente la más comunmente utilizada. Su uso está muy extendido en el desarrollo web, y es prácticamente una de las primeras unidades de CSS que se recomienda aprender, ya que su valor es siempre el mismo y es fácil de aprender.

Observa el siguiente fragmento de código:

```
.container {  
width: 300px;
```

```
height: 200px;  
background: indigo;  
}
```

En este fragmento utilizamos las propiedades **width (tamaño de ancho)** y **height (tamaño de alto)** para dar un tamaño concreto al elemento con clase **.container**. De esta forma, en todos los navegadores que veamos la página, ese elemento siempre tendrá un tamaño de **300x200** píxels.

Densidad de píxeles

## Otras unidades absolutas

A parte del píxel, es conveniente saber que no es la única unidad absoluta que existe, y que probablemente más adelante nos interese aprender otros tipos de unidades que pueden ser mucho más versátiles y útiles en situaciones concretas. A continuación, podemos ver las diferentes unidades absolutas que pueden utilizarse en CSS son las siguientes (**de mayor a menor tamaño**):

| Unidad         | Significado  | Medida aproximada                           |
|----------------|--------------|---|
| px             | Píxels       | -   |
| pt             | Puntos       | $96\text{px} / 72 \approx 1.33\text{px}$    |
| Otras unidades |              |   |
| in             | Pulgadas     | 96px  |
| cm             | Centímetros  | $96\text{px} / 2.54 \approx 37.79\text{px}$ |
| pc             | Picas        | $96\text{px} / 6 \approx 16\text{px}$       |
| mm             | Milímetros   | $96\text{px} / 25.4 \approx 3.78\text{px}$  |
| Q              | Cuarto de mm | $96\text{px} / 6.35 \approx 15.12\text{px}$ |

La mayoría de estas unidades, a excepción de **px**, no suelen utilizarse en el día a día del desarrollo web, sino que se utilizan exclusivamente en casos concretos de medios impresos o similares.

## Unidades relativas

Las unidades relativas son un tipo de medida mucho más potente y habitual en el CSS que creamos generalmente. Al contrario que las unidades absolutas, las unidades relativas dependen de algún otro factor (**elemento padre, tamaño de letra, interlineado, etc...**).

Tienen una curva de aprendizaje más compleja, pero son ideales para trabajar en dispositivos con diferentes tamaños, ya que son muy flexibles y versátiles y bien utilizadas pueden ahorrar trabajo de desarrollo.

Las unidades relativas más conocidas son las siguientes:

| Unidad   | Medida aproximada                                      | Ejemplo  |
|----------|--|--|
| %        | Relativa a herencia (concretamente, al elemento padre) | 50% = mitad del padre                            |
| em / rem | Relativo al <b>font-size</b> en ese elemento           | $1.5\text{em} = 1.5 * 16\text{px} = 24\text{px}$ |
| lh / rlh | Relativo al <b>line-height</b> en ese elemento         |  |

| Unidad            | Medida aproximada  | Ejemplo                 |
|-------------------|--|-------------------------|
| <b>ex / rex</b>   | Relativo a la altura del carácter <b>x minúscula</b>             | <b>1ex ~ 0.5em</b>      |
| <b>ch / rch</b>   | Relativo al ancho del carácter <b>0</b>                          | <b>1ch ~ 1 carácter</b> |
| <b>cap / rcap</b> | Relativo a la altura del primer carácter en <b>mayúscula</b>     |                         |
| <b>ic / ric</b>   | Relativo al ancho del glifo 水 <b>CJK</b> (Chino-Japones-Koreano) |                         |

Probablemente, las más conocidas son **%**, **em** y **rem**, por lo que te recomiendo empezar con ellas.

Unidades relativas

## La unidad % (porcentaje)

Los porcentajes llevan mucho tiempo en el universo de CSS, sin embargo requieren cierta destreza y deben usarse con cuidado en nuestras primeras fases de aprendizaje, ya que el valor de porcentaje establecido depende del tamaño del elemento padre inmediato.

```
<div class="parent">
  <div class="child"></div>
</div>
```

En este caso anterior, el elemento padre .parent tendrá un tamaño de 200px. Su hijo, el elemento .child tiene un tamaño de ancho de 50%, o lo que es lo mismo, la mitad del tamaño de su padre: 100px. Por otro lado, si el elemento padre .parent no tuviera una propiedad width definida, su elemento hijo tendría valor 0. Entenderemos mejor este último caso, cuando veamos la propiedad display.

### % (porcentaje)

El **porcentaje depende del tamaño del texto del elemento padre inmediato**.

### Cómo funciona

Cuando usas % en font-size, el navegador hace esto:

tamaño final = tamaño del padre × porcentaje

### Ejemplo

```
body {
  font-size: 16px;
}
```

```
p {  
  font-size: 150%;  
}
```

El párrafo tendrá:

**16px × 150% = 24px**

Si el padre cambia, el tamaño del hijo también cambia.

## Detalles importantes

- % **solo mira al padre directo**
- Se comporta parecido a em, pero es más explícito
- Poco usado hoy para tipografía moderna

## em

em es una **unidad relativa al tamaño de fuente del elemento padre...** pero con un detalle clave.



### Cómo funciona

- 1em = tamaño del texto del padre
- Se **acumula** cuando hay anidamiento



### Ejemplo clásico

```
body {  
  font-size: 16px;  
}  
  
.caja {  
  font-size: 1.5em; /* 24px */  
}  
  
.caja p {  
  font-size: 1.5em;  
}
```



Cálculo:

- .caja →  $16 \times 1.5 = \mathbf{24px}$
- p →  $24 \times 1.5 = \mathbf{36px}$

Aquí está el “problema” de em: **escala en cascada.**

## Cuándo usar em

- Padding, margin relacionados con texto
- Componentes que deben escalar internamente

## rem (root em)

rem significa **root em**, y **siempre depende del tamaño del html**, no del padre.



### Cómo funciona

```
html {  
  font-size: 16px;  
}
```

- 1rem = 16px
- 2rem = 32px

Siempre igual, sin importar dónde esté el elemento.

## Ejemplo

```
.caja { font-size: 2rem; }
```

```
.caja p { font-size: 1rem; }
```

El párrafo será **16px**, aunque esté dentro de una caja gigante.

## Comparación directa

| Unidad | Depende de    | Se acumula | Uso típico        |
|--------|---------------|------------|-------------------|
| %      | Padre directo | ✗          | Ajustes simples   |
| em     | Padre         | ✓          | Componentes       |
| rem    | html          | ✗          | Tipografía global |

## Recomendación práctica (real)

En proyectos modernos:

```
html { font-size: 16px; }  
body { font-size: 1rem; }
```

```
p { font-size: 1rem; }  
h1 { font-size: 2.5rem; }
```

Y usa em para:

```
button {  
  padding: 0.5em 1em;  
}
```

## Unidades de viewport

Existen unas unidades de nueva generación que resultan muy útiles, porque dependen del viewport (**región visible de la página web en el navegador**). Con estas unidades podemos hacer referencia a un porcentaje concreto del tamaño específico que tengamos en la ventana del navegador.

Es decir, cada vez que hagamos referencia a una unidad precedida por **v** estamos haciendo referencia a un **%** del tamaño total de la ventana. De esta forma, podemos hacer que elementos concretos tengan valores proporcionales al tamaño de la ventana del navegador.

Las unidades de esta categoría son las siguientes:

| Unidad      | Significado      | Medida aproximada                                   |                         | Más info |
|-------------|------------------|---|-------------------------|----------|
| <b>vw</b>   | viewport width   | <b>1vw = 1%</b> del ancho del navegador             |                         |          |
| <b>vh</b>   | viewport height  | <b>1vh = 1%</b> del alto del navegador              |                         |          |
| <b>vmin</b> | viewport minimum | <b>1vmin = 1%</b> del alto o ancho (el mínimo)      |                         |          |
| <b>vmax</b> | viewport maximum | <b>1vmax = 1%</b> del alto o ancho (el máximo)      |                         |          |
| <b>vi</b>   | viewport inline  | Versión lógica inline de <b>vw</b> y/o <b>vh</b>    | Ver propiedades lógicas |          |
| <b>vb</b>   | viewport block   | Versión lógica en bloque de <b>vw</b> y/o <b>vh</b> | Ver propiedades lógicas |          |

## La unidad vw (viewport width)

La unidad vw es un tipo de unidad que hace referencia al **ancho del viewport** (*región visible de la página en el navegador*). Por ejemplo, si definimos 50vw, estamos indicando un 50% del ancho actual del navegador. En la siguiente demo, el viewport es la región en blanco que aparece:

```
body {  
  margin: 0; /* Reset */  
}  
  
.container {  
  width: 50vw; height: 50px;  
  background: indigo;  
}
```

En el caso anterior, si la ventana del navegador ocupa 800px de ancho, la propiedad width tendrá un valor de 400px. Esto nos permite utilizar tamaños que dependan de las dimensiones de la ventana del navegador.

## La unidad vh (viewport height)

Por su parte, la unidad vh hace referencia de la misma forma al **alto del viewport**. Si con la unidad vw podemos hacer referencia al ancho, con vh hacemos referencia al alto.



```
body {  
  margin: 0; /* Reset */  
}  
.container {  
  width: 50vw;  
  height: 100vh;  
  background: indigo;  
}
```

En este ejemplo, mientras 50vw referencia al 50% del **ancho del navegador**, con la propiedad height a 75vh estaremos haciendo referencia al 75% del **alto del navegador**. Si nuestro navegador tiene un tamaño de 1080px de alto, con 75vh estaríamos indicando unos 810px.

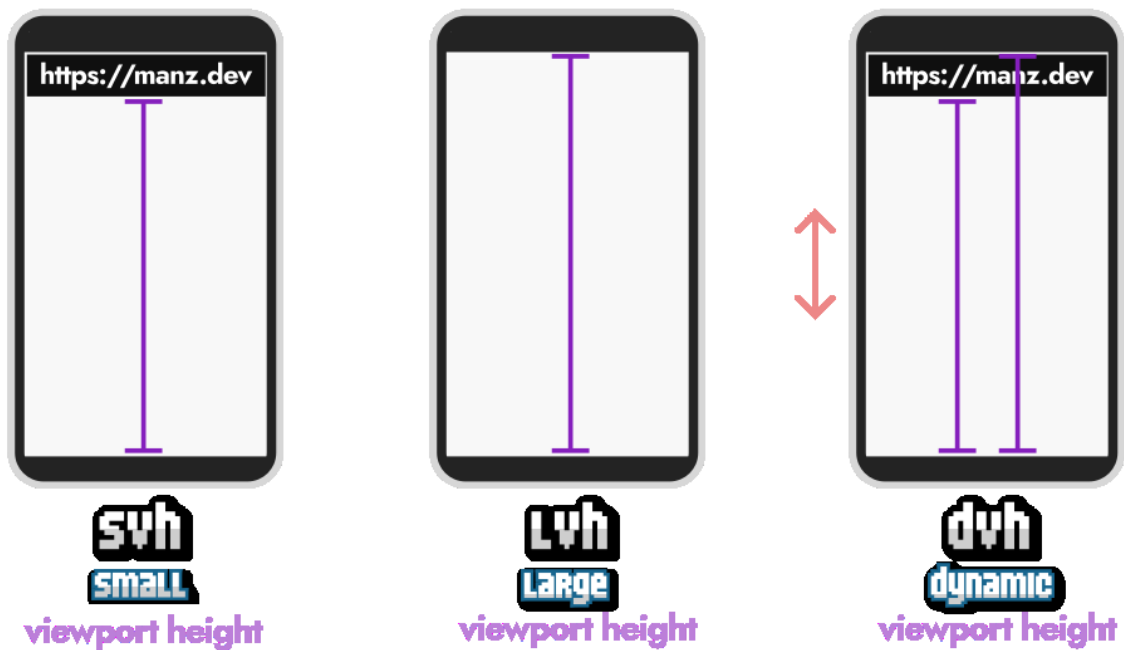
Recuerda que por defecto, el navegador añade unos márgenes en elementos como <body>, por lo que si no los reseteas a cero, no te va a encajar exactamente y aparecerán las scrollbars.

## Unidades de viewport dinámicas

Si has leído las unidades anteriores, debes saber que las unidades que veremos a continuación son variaciones de las unidades de viewport. Existen unas unidades prefijadas por las letras **s**, **l** y **d**, pero que son exactamente las mismas que las anteriores:

| Unidad                       | Significado      | Medida aproximada                                       | Más info |
|------------------------------|------------------|---|----------|
| <b>svw / lvw / dvw</b>       | viewport width   | Relativo al ancho del navegador                         |          |
| <b>svh / lvh / dvh</b>       | viewport height  | Relativo al alto del navegador                          |          |
| <b>svmin / lvmin / dvmin</b> | viewport minimum | Relativo al valor más pequeño entre el alto y el ancho. |          |
| <b>svmax / lvmax / dvmax</b> | viewport maximum | Relativo al valor más grande entre el alto y el ancho.  |          |
| <b>svi / lvi / dvi</b>       | viewport inline  | Versión lógica inline de <b>*vw</b> y/o <b>*vh</b>      |          |
| <b>svb / lvb / dvb</b>       | viewport block   | Versión lógica en bloque de <b>*vw</b> y/o <b>*vh</b>   |          |

Para entender estas unidades, el siguiente gráfico resulta bastante representativo:



#### □ Small Viewport (Unidades: **svw**, **svh**, **svmin**, **svmax**, **svi**, **svb**)

Observa que la letra de prefijo denota que tipo de unidad estamos utilizando. Por ejemplo, si utilizamos las unidades prefijadas por **s**, estamos haciendo referencia a unidades que excluirán la parte de la interfaz del navegador donde escribes la URL del navegador, excluyéndolo.

#### □ Large viewport (Unidades: **lvw**, **lvh**, **lvmin**, **lvmax**, **lvi**, **lvb**)

Por otro lado, si utilizamos unidades prefijadas por **l**, estaremos haciendo referencia a unidades que incluyen en su tamaño el espacio de la interfaz del navegador donde escribes la URL.

#### □ Dynamic viewport (Unidades: **dvw**, **dvh**, **dvmin**, **dxmax**, **dvi**, **dvb**)

Por último, con las unidades prefijadas por **d**, es un tipo de unidad que tendrá en cuenta cuando el usuario hace scroll y los navegadores ocultan y muestran la barra del navegador.

En algunas ocasiones te interesará una de estas unidades y en otras ocasiones otras. No hay unidades mejores o peores, simplemente ocasiones en las que interesa más una u otra.

# Práctica

Haz un ejemplo con cada una de las unidades mostradas.