

Diseño de Interfaces Web

Guías de estilo y
lenguaje de marcas

Tamaños

Existen multitud unidades de medida para especificar tamaños en CSS.

- ☐ PX.
- ☐ EM.
- ☐ %.
- ☐ Etc.

La diferencia principal es que px es una medida estática mientras que em y % son relativas (al tamaño de la fuente actual o al tamaño de la fuente padre).

Unidades estáticas

Se recomiendan para medios conocidos, como por ejemplo impresión (se puede configurar para impresión en A4 que siempre tiene el mismo tamaño de ancho por alto).

Por pantalla no es conocido a priori, puesto que se puede acceder a una página Web desde multitud de dispositivos con infinidad de tamaños de pantalla diferentes.

Unidades estáticas

Ejemplos de unidades estáticas o absolutas:

- ☐ Px.
- ☐ Cm.
- ☐ Mm.
- ☐ In.
- ☐ Pt.
- ☐ Pc.

Unidades relativas

Ejemplos de unidades relativas:

- ☐ Em.
- ☐ Ex.
- ☐ Ch.
- ☐ Rem.
- ☐ Vw.
- ☐ Vh.
- ☐ Vmin.
- ☐ Vmax.
- ☐ %.

Responsive

Para tener un control absoluto del tamaño y hacer una interfaz escalable, se emplean %, em, rem y vw.

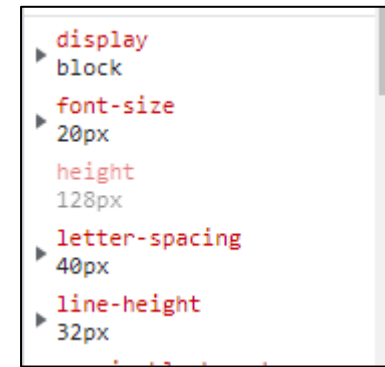
El valor por defecto de tamaño de fuente de una etiqueta es de 16px si no se aplica ninguna regla.

Em hace referencia a em veces el tamaño de fuente del elemento. Sin embargo, rem toma su valor respecto del tamaño fuente principal (aplicado a <html>).

Ejemplo

Aplicando sobre un párrafo:

```
p {  
  font-size: 20px;  
  line-height: 2rem;  
  letter-spacing: 2em;  
}
```



Media queries

Se introdujo en la versión 3 de CSS. Se emplea para imponer reglas según el tamaño del dispositivo, zona de visualización, orientación y resolución.

El formato es:

```
@media not|only mediatype and  
(expressions) { /* reglas */ }
```

Mediatype puede valer: all | print | screen | speech

Media queries

Las expressions pueden ser:

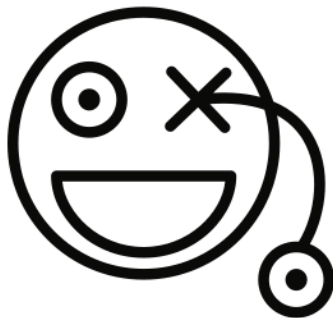
- ☐ Min-width.
- ☐ Max-width.
- ☐ (Min-width) and (max-width)
- ☐ Orientation: landscape | portrait.

Ejemplo:

```
@media screen and (min-width: 480px){  
  p {  
    font-size: 2em;  
  }  
}
```

Actividad

Entrega tu código html y css necesario para obtener esta visualización en un navegador como Google Chrome. Se emplea `rgb(0,0,0)`, Segoe UI“, 15/20px, 34/40px, #dbdbdb.



Lorem ipsum et dolor sumum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere. Sed venenatis orci eu massa facilisis mattis. Donec posuere, odio nec accumsan eleifend, elit nulla sodales leo, nec cursus mi diam sed justo. Aliquam tellus dui, blandit at tristique sed, pulvinar quis nibh. Donec convallis ultricies venenatis. Fusce eu orci non lectus congue rutrum vitae sit amet sapien. Nulla efficitur fringilla urna, ac volutpat leo mattis sed. [et dolor](#) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere. Sed venenatis orci eu massa facilisis mattis. Donec posuere, odio nec accumsan eleifend, elit nulla sodales leo, nec cursus mi diam sed justo. Aliquam tellus dui, blandit at tristique sed, pulvinar quis nibh. Donec convallis ultricies venenatis. Fusce eu orci non lectus congue rutrum vitae sit amet sapien. Nulla efficitur fringilla urna, ac volutpat leo mattis sed.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere. Sed venenatis orci eu massa facilisis mattis. Donec posuere, odio nec accumsan eleifend, elit nulla sodales leo, nec cursus mi diam sed justo. Aliquam tellus dui, blandit at tristique sed, pulvinar quis nibh. Donec convallis ultricies venenatis. Fusce eu orci non lectus congue rutrum vitae sit amet sapien. Nulla efficitur fringilla urna, ac volutpat leo mattis sed

Lorem

dolor sumum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere.

Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere.

Et

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere.

Actividad

La vista responsive para máximo ancho de 375 px es la siguiente:

Lorem	dolor sumum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere.
Ipsum	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere.
Et	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum ac laoreet metus. Vestibulum lacinia vulputate posuere.

Flexbox

Proporciona un mecanismo sencillo y flexible para disponer elementos en un contenedor, sobre todo si se desconocen a priori los tamaños de los elementos que contiene.

Por tanto, es un mecanismo efectivo para diseñar elementos en nuestra interfaz que sean fácilmente responsive (disposición adecuada en diferentes anchos de dispositivos).

El funcionamiento de este módulo es permitir al contenedor modificar los anchos y altos de los elementos que contiene, de forma que se expandan/contraigan de forma eficiente en el espacio que tienen.

Flexbox

Se añaden además numerosas opciones en cuanto a flexibilidad en el orden, alineación, espacio ocupado, etc. de los elementos del contenedor.

Partes en que consta:

- ☐ Contenedor: es el elemento padre.
- ☐ Ítems: son los elementos hermanos que se encuentran como hijos del contenedor.

Los ítems se distribuyen siguiendo el eje principal o el eje transversal, y lo hacen desde el inicio hasta el final del eje.

El eje principal de un contenedor es aquel donde se disponen los ítems. Depende de la propiedad flex-direction para que este eje sea horizontal o vertical.

Flexbox

El tamaño será el ancho o alto de los ítems.

El eje transversal es aquel perpendicular al eje principal.

Propiedades del contenedor:

- ☐ Display: flex;
- ☐ Flex-direction: row | row-reverse | column | column-reverse;
- ☐ Flex-wrap: nowrap | wrap | wrap-reverse;
- ☐ Flex-flow: column wrap;
- ☐ Justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly | ...;
- ☐ Align-items: stretch | flex-start | flex-end | center | baseline | ...;

Flexbox

Propiedades del contenedor:

- ❑ `align-content: flex-start | flex-end | center | space-between | space-around | space-evenly | stretch | ...;`

Propiedades del ítem:

- ❑ `order: 2; /* por defecto vale 0 */`
- ❑ `flex-grow: 3; /* por defecto vale 0 */`
- ❑ `flex-shrink: 4; /* por defecto vale 1 */`
- ❑ `flex-basis` (define el tamaño de un elemento antes de que se distribuya por el espacio que quede libre): `| auto; /* por defecto vale auto */`
- ❑ Flex (resume las tres anteriores en esta notación única):
`none | [<'flex-grow'> <'flex-shrink'>? || <'flex-basis'>]`
- ❑ `Align-self: auto | flex-start | flex-end | center | baseline | stretch;`

Actividad



Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Maecenas suscipit, dolor quis
aliquam congue, sapien felis rhoncus quam,
sed blandit ligula augue ac sapien.



Lorem ipsum dolor sit
amet, consectetur
adipiscing elit.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Maecenas suscipit, dolor quis
aliquam congue, sapien felis
rhoncus quam, sed blandit
ligula augue ac sapien.

Lorem ipsum dolor sit
amet, consectetur
adipiscing elit. Maecenas
suscipit.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Maecenas suscipit, dolor quis
aliquam congue, sapien felis

Lorem ipsum dolor sit
amet.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Maecenas suscipit, dolor quis
aliquam congue, sapien felis
rhoncus quam, sed blandit
ligula augue ac sapien.



Grid

Es un sistema en CSS basado en el diseño mediante una cuadrícula.

Al igual que Flexbox, simplifica enormemente la maquetación de una página Web, permitiendo diseños complicados y permitiéndonos dejar a Grid que se encargue del ajuste de tamaños y alturas de elementos según el espacio que se disponga en la pantalla.

Además de esta flexibilidad en el control de elementos en dos dimensiones, se emplea menos código, optimizan recursos y se facilita el responsive.

Grid

¿Cómo se implementa?

- ❑ Se crea el bloque contenedor y se le aplica la propiedad `display: grid`.
- ❑ Se definen las filas (`grid-template-rows`), columnas (`grid-template-columns`) y a continuación se colocan los elementos hijos en la rejilla empleando `grid-column` y `grid-row`.

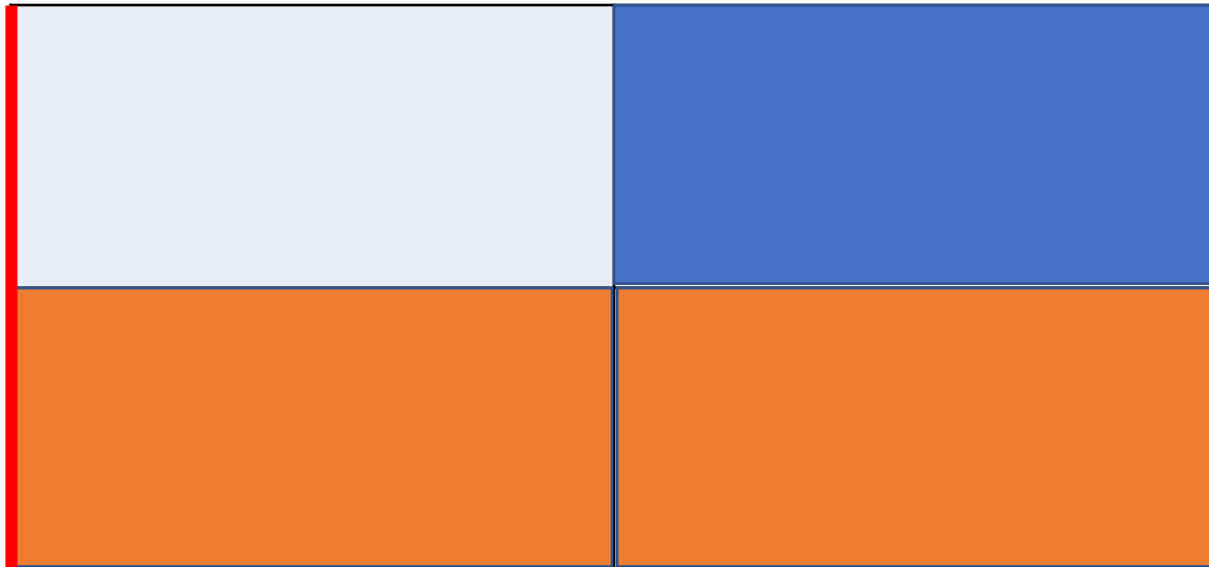
Al igual que Flex, el orden de los elementos no importa puesto que puede ser modificado con Grid.

Se puede usar cualquier unidad vista hasta ahora, o bien `fr` (fracción del espacio restante) de Grid.

Grid

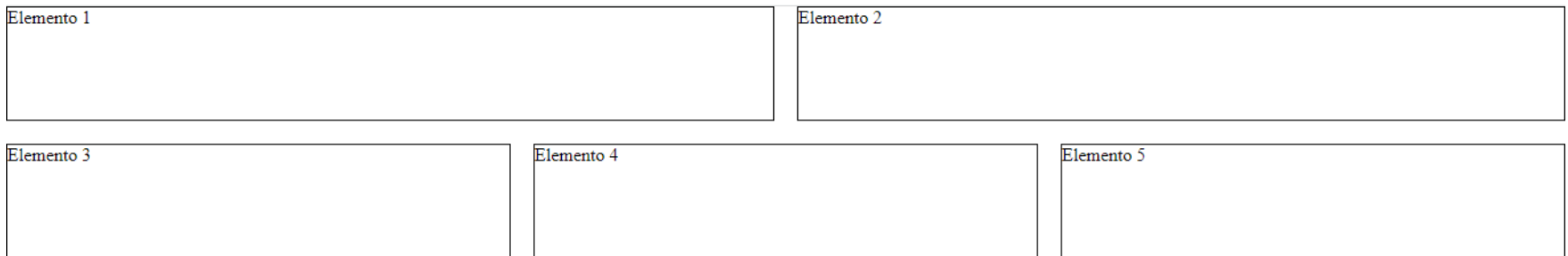
Otros términos importantes:

- ❑ Grid Line.
- ❑ Grid Cell.
- ❑ Grid Track.
- ❑ Grid Area.



Grid

Ejemplo



```
<div class="grid_contenedor">
  <div class="grid_item_1">Elemento 1</div>
  <div class="grid_item_2">Elemento 2</div>
  <div class="grid_item_3">Elemento 3</div>
  <div class="grid_item_4">Elemento 4</div>
  <div class="grid_item_5">Elemento 5</div>
</div>
```

```
.grid_contenedor {
  display: grid;
  grid-template-columns: repeat(6, 1fr);
  grid-template-rows: 100px 100px;
  column-gap: 20px;
  row-gap: 20px;
  justify-content: center;
}
.grid_item_1 {
  grid-row: 1;
  grid-column: 1 / 4;
  border: 1px solid #000;
}
.grid_item_2 {
  grid-row: 1;
  grid-column: 4 / 7;
  border: 1px solid #000;
}
```

```
.grid_item_3 {
  grid-row: 2;
  grid-column: 1 / 3;
  border: 1px solid #000;
}
.grid_item_4 {
  grid-row: 2;
  grid-column: 3 / 5;
  border: 1px solid #000;
}
.grid_item_5 {
  grid-row: 2;
  grid-column: 5 / 7;
  border: 1px solid #000;
}
```

Grid

Propiedades

Contenedor	Elementos
<ul style="list-style-type: none"><input type="checkbox"/> Display<input type="checkbox"/> Grid-template-columns<input type="checkbox"/> Grid-template-rows<input type="checkbox"/> Grid-template-areas<input type="checkbox"/> Column-gap<input type="checkbox"/> Row-gap<input type="checkbox"/> Grid-column-gap<input type="checkbox"/> Grid-row-gap<input type="checkbox"/> Justify-items<input type="checkbox"/> Align-items<input type="checkbox"/> Justify-content<input type="checkbox"/> Align-content	<ul style="list-style-type: none"><input type="checkbox"/> Grid-column-start<input type="checkbox"/> Grid-column-end<input type="checkbox"/> Grid-row-start<input type="checkbox"/> Grid-row-end<input type="checkbox"/> Grid-column<input type="checkbox"/> Grid-row<input type="checkbox"/> Grid-area<input type="checkbox"/> Justify-self<input type="checkbox"/> Align-self

Bootstrap

Es una librería o conjunto de herramientas útiles para el diseño en nuestro front-end.

Contiene multitud de elementos, iconos, componentes y plugins para aplicar estilos CSS, animaciones y efectos a nuestras páginas Web.

Bootstrap

El primer paso es implementar la librería, bien descargándola, bien haciendo link a ella.

<https://github.com/twbs/bootstrap/releases/download/v5.1.3/bootstrap-5.1.3-dist.zip>

Bootstrap

Breakpoints

Se corresponde con los anchos personalizados que determinan cómo se comportará tu diseño responsive.

Empleándolos estarás haciendo tu sitio responsive desde el primer momento.

Bootstrap

Breakpoints

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px

Bootstrap

Contenedores

Elemento básico de Bootstrap necesario si se emplea el Grid propio de la librería.

Tipos de contenedores:

- ☐ `.container` (ancho máximo al del breakpoint).
- ☐ `.container-fluid` (ancho 100% en todos los breakpoints).
- ☐ `.container-{breakpoint}` (ancho 100% en el breakpoint especificado).

Bootstrap

Grid

El sistema empleado por Bootstrap para el diseño de su rejilla es mediante flexbox, y para ello se emplean las siguientes clases:

- ☐ Row.
- ☐ Col.

Hay 12 columnas disponibles por cada fila. Si no se especifican las columnas, tomará el ancho necesario.

Bootstrap

Grid

Para tomar como ancho de la columna según el contenido se empleará la etiqueta auto:

`Col-{breakpoint}-auto`

Bootstrap

Columnas

Como se comentaba anteriormente, Bootstrap emplea flexbox por lo que las propiedades de flex pueden emplearse en las columnas en las clases empleadas:

- ☐ Align-items-start (align-items-center, etc.).
- ☐ Align-items-self-start (align-items-self-center, etc.).
- ☐ Justify-content-center..
- ☐ Etc.



CEU

*Centro de Estudios
Profesionales*

Fundación San Pablo Andalucía