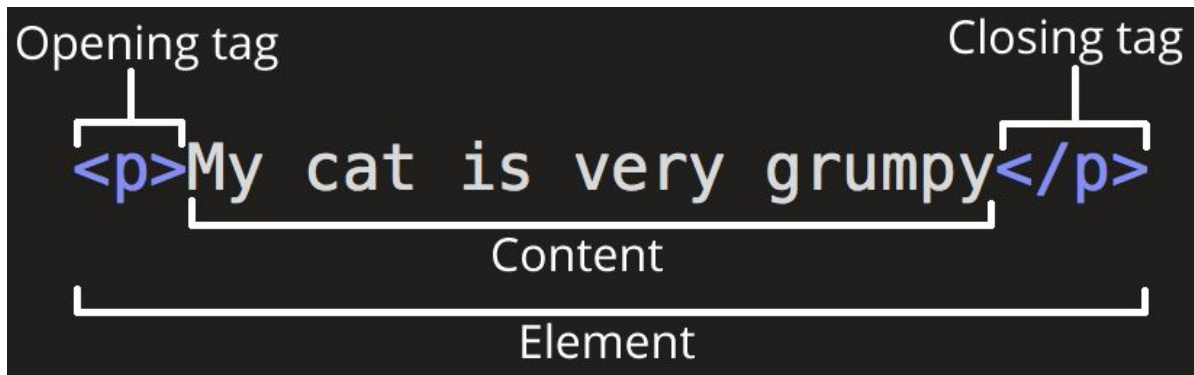


# Webapp 6

# HTML

**H**ypertext **M**arkup **L**anguage define la estructura del contenido de una web mediante **<etiquetas></etiquetas>**. Está basado en XML.

Entre las etiquetas se puede incluir contenido o más etiquetas:



[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction to HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML)

# HTML Etiquetas importantes

- div
- section
- p
- span
- h1, h2, h3, h4, h5 y h6
- a
- button
- header
- footer
- article
- aside
- ul
- ol
- li
- nav
- main
- details
- strong
- em
- img
- input

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

# HTML Bloques

Hay elementos de HTML que se ponen en línea y otros que no. La propiedad que cambia si se pone en línea es **display: block;** o **display: inline-block;**

```
<p>H</p><p>o</p><p>l</p><p>a</p>
```

H

o

l

a

```
<span>H</span><span>o</span><span>l</span><span>a</span>
```

Hola

# HTML Etiquetas Autocerradas

Hay algunas **<etiquetas/>** autocerradas como: **<img />**, **<input />**, **<link />**, etc.

[http://xahlee.info/js/html5\\_non-closing\\_tag.html](http://xahlee.info/js/html5_non-closing_tag.html)

# HTML Atributos

Una etiqueta puede tener N atributos:

```
<span class="subtitle" title="Hello">Hello world</span>
```

Hay atributos que admiten varios valores:

```
<span class="subtitle important primary">Hello world</span>
```

# HTML Atributos importantes

- class
- id
- data-
- style
- disabled
- readonly
- required
- min, max
- src
- tabindex

<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>

# HTML Semántico

Blog de programación

[Contacto](#)

[Artículos](#)

[Acerca de](#)

[Artículos recientes](#)

[Artículos más votados](#)

Artículos

[- 2020](#)

[- 2019](#)

[- 2018](#)

## ¡Hola mundo!

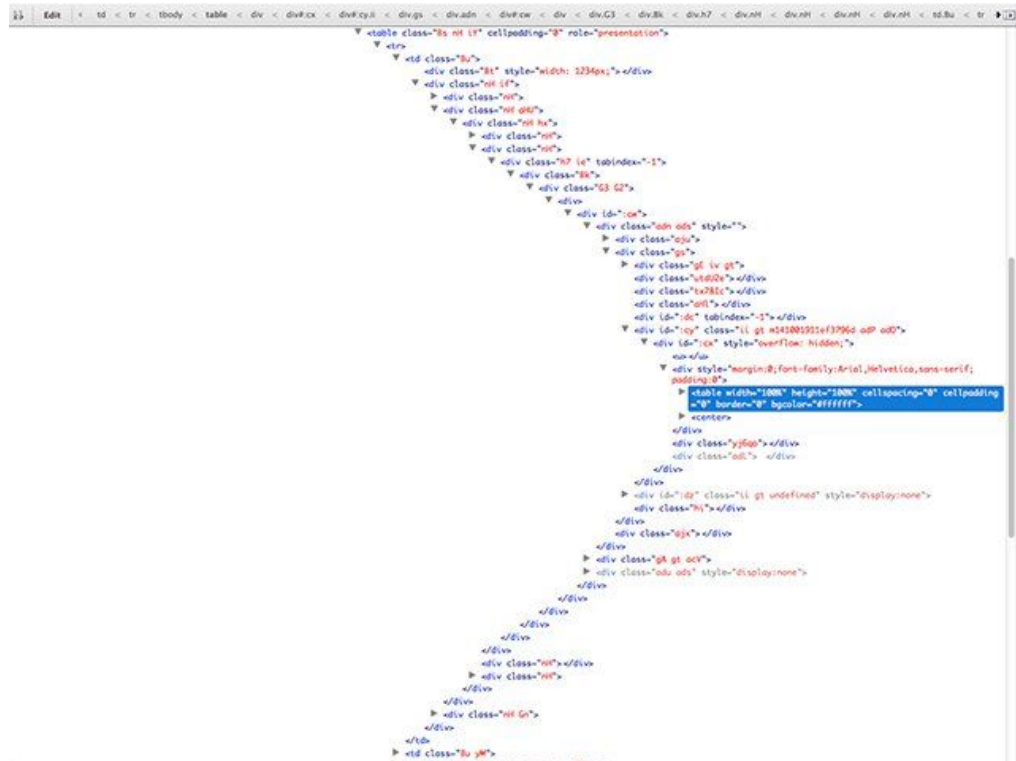
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ab accusantium animi assumenda consequuntur dicta dignissimos dolor est expedita iste iusto magni numquam provident ratione reiciendis saepe unde velit voluptas, voluptatibus?



# Maquetación

- Tablas
- Floats
- Flexbox
- Grid

# Maquetación II - Tablas



A short sentence telling readers how you got their email address and where to unsubscribe.

# 1-2-3

HOME | LINK 1 | LINK 2 | LINK 3

## Fusce Interdum

Donec in lacus in ante facilis dignissim quis auctor elit.  
Praesent hendrerit ligula sit amet tortor luctus, ut  
hendrerit turpis fringilla. Maecenas nec sapien lacinia,  
mollis nisi ut, euismod turpis.

Curabitur cursus metus hendrerit iaculis laoreet.  
Vivamus risus felis, sollicitudin at ipsum eu, placerat  
laoreet massa. Phasellus eget justo luctus, facilisis urna  
lobortis, consectetur tellus. Nunc quis euismod purus,  
dapibus condimentum enim.

## Side Content Heading

Link 1  
Link 2  
Link 3

Footer text and links go here.

UNSUBSCRIBE INSTRUCTIONS/LINK

**CONTACT US:**  
FRED FLINTSTONE  
BEDROCK GRAVEL, INC.  
123 BENCH RD  
BEDROCK, NY 11111

# Maquetación III - Floats



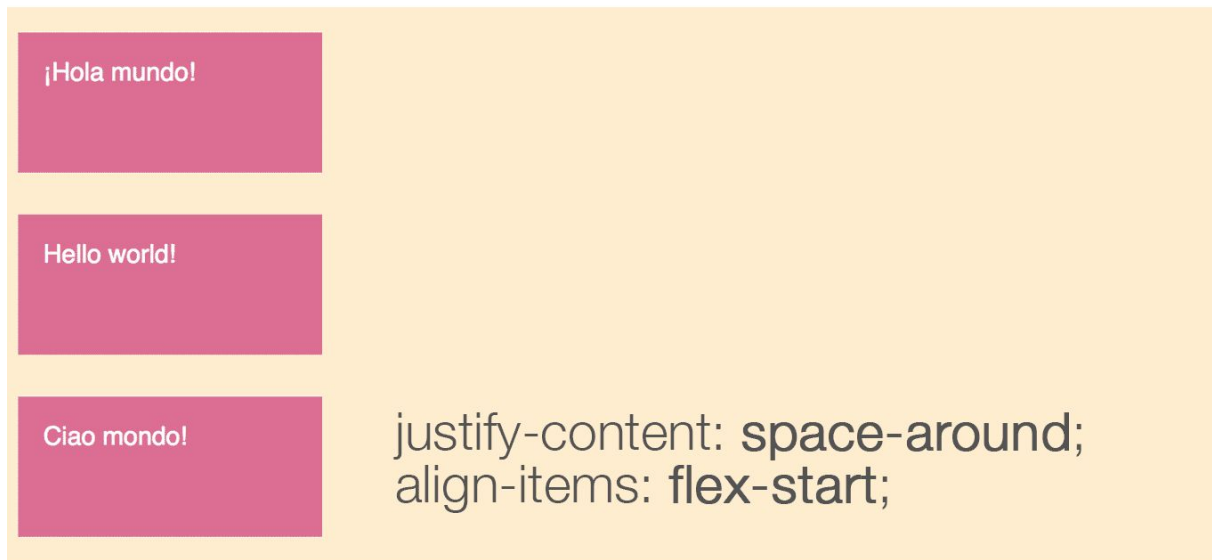
The images are contained within the paragraph tag. The image has floated to the left, and also to the right because we have used both types of image floating in this example. Notice how the text wraps around the images quite nicely. The images are contained within the paragraph tag. The image has floated to the left, and also to the right because we have used both types of image floating in this example. Notice how the text wraps around the images quite nicely.

This second paragraph has an image that is floated to the right. It should be noted that a margin should be added to images so that the text does not get too close to the image. There should always be a few pixels between words and borders, images, and other content. This second paragraph has an image that is floated to the right. It should be noted that a margin should be added to images so that the text does not get too close to the image. There should always be a few pixels between words and borders, images, and other content.



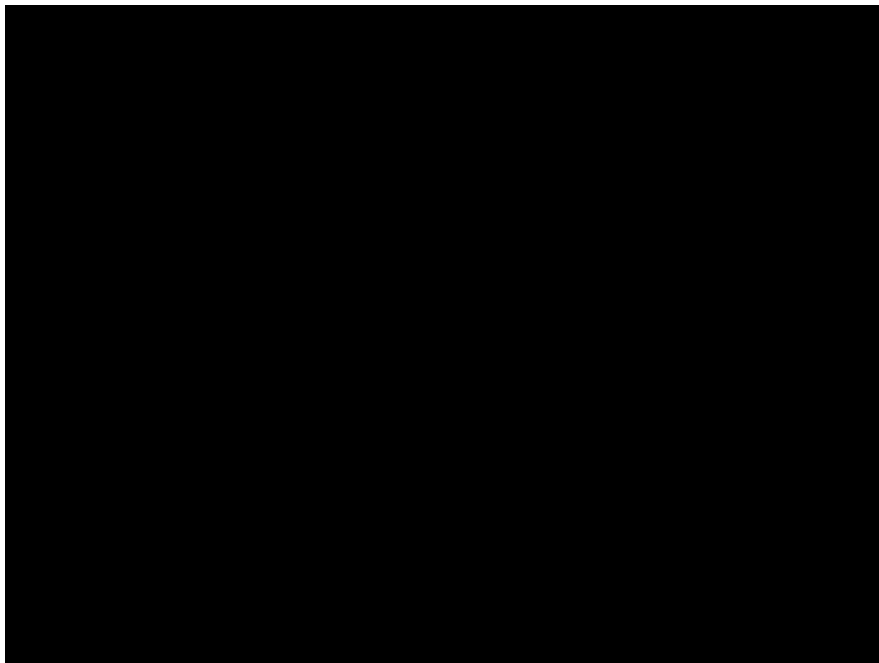
<https://internetingishard.com/html-and-css/floats/>

# Maquetación IV - Flexbox



<https://www.adictosaltrabajo.com/2018/02/14/maquetacion-con-flexbox/>

# Maquetación V - Grid



<https://www.adictosaltrabajo.com/2018/01/30/maquetacion-con-css-grid/>

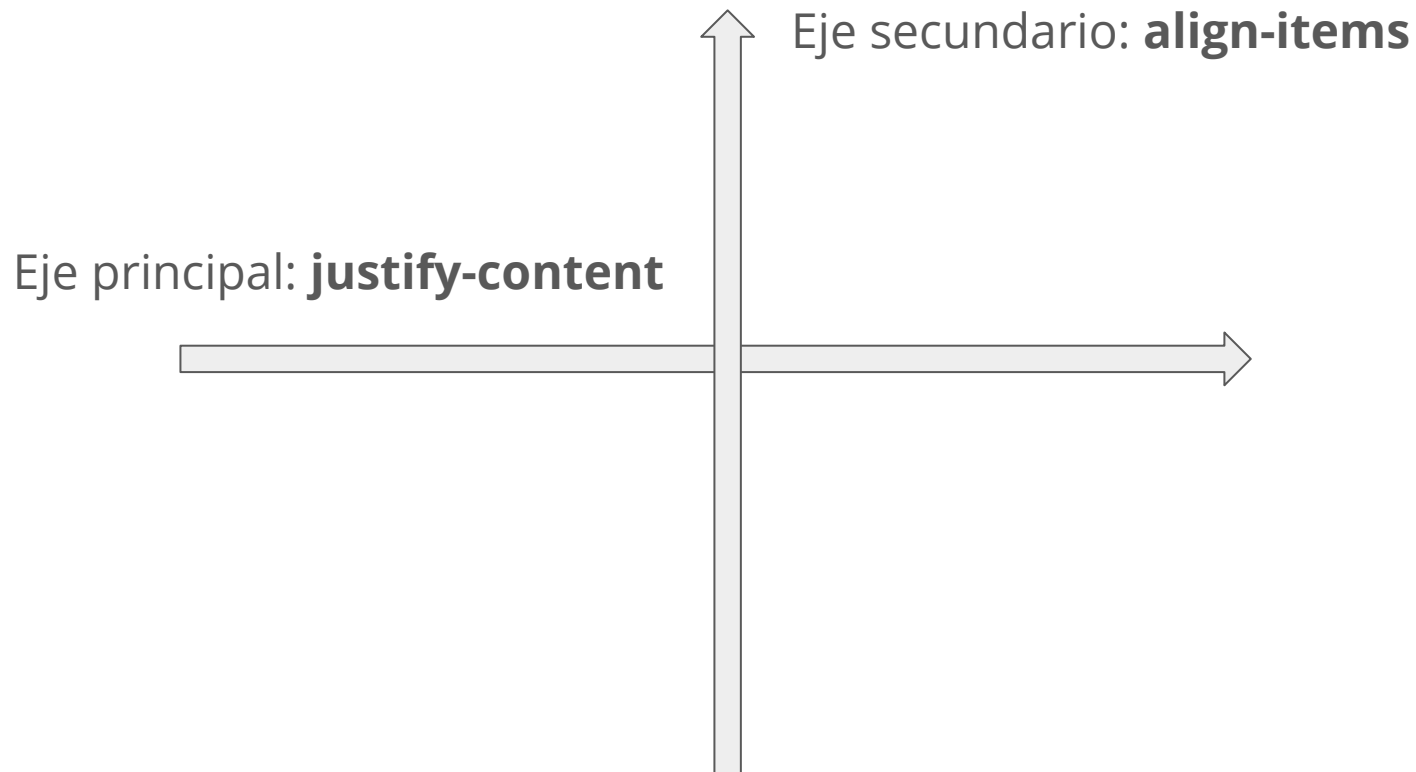
# Flexbox

Se aplica **display: flex;** en el contenedor. Todos los elementos **inmediatos** se pondrán en modo “flex”.



<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

# Flexbox II



# Flexbox III - Justify Content

Contenedor

Define la alineación en el **eje principal**.

**justify-content:**

- flex-start
- flex-end
- center
- space-between
- space-around
- space-evenly



# Flexbox IV - Align items

Contenedor

Define la alineación en el **eje secundario**.

## **align-items:**

- stretch
- flex-start
- flex-end
- center
- baseline

# Flexbox V - Flex direction

Contenedor

Define el **eje principal**.

**flex-direction:**

- row
- column
- row-reverse
- column-reverse

# Flexbox VI - Flex grow

Contenido

Define la proporción a la que **agrandarse** de ser necesario.

**flex-grow:**

- <number>

# Flexbox VII - Flex shrink

[Contenido](#)

Define la proporción a la que **encogerse** de ser necesario.

**flex-shrink:**

- <number>

# Flexbox VIII - Flex basis

Contenido

Define el tamaño por defecto antes que se distribuya el espacio restante.

**flex-basis:**

- <length> (rem, em, px, %, etc)

<https://www.paradigmadigital.com/dev/diferencia-flex-basis-width/>

# Flexbox IX - Flex

## Contenido

Shorthand property para flex-grow, flex-shrink y flex-basis. Recomendado usar esta en propiedad en vez de las otras propiedades.

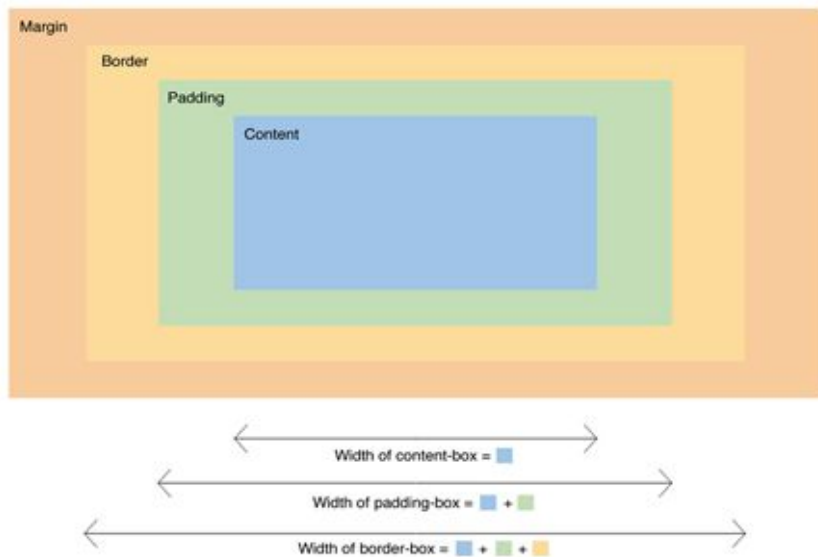
### **flex:**

- none | [ <'flex-grow'> <'flex-shrink'>? | | <'flex-basis'> ]

# Flexbox VI

Práctica: <https://flexboxfroggy.com/>

# Box sizing



<https://css-tricks.com/box-sizing/>



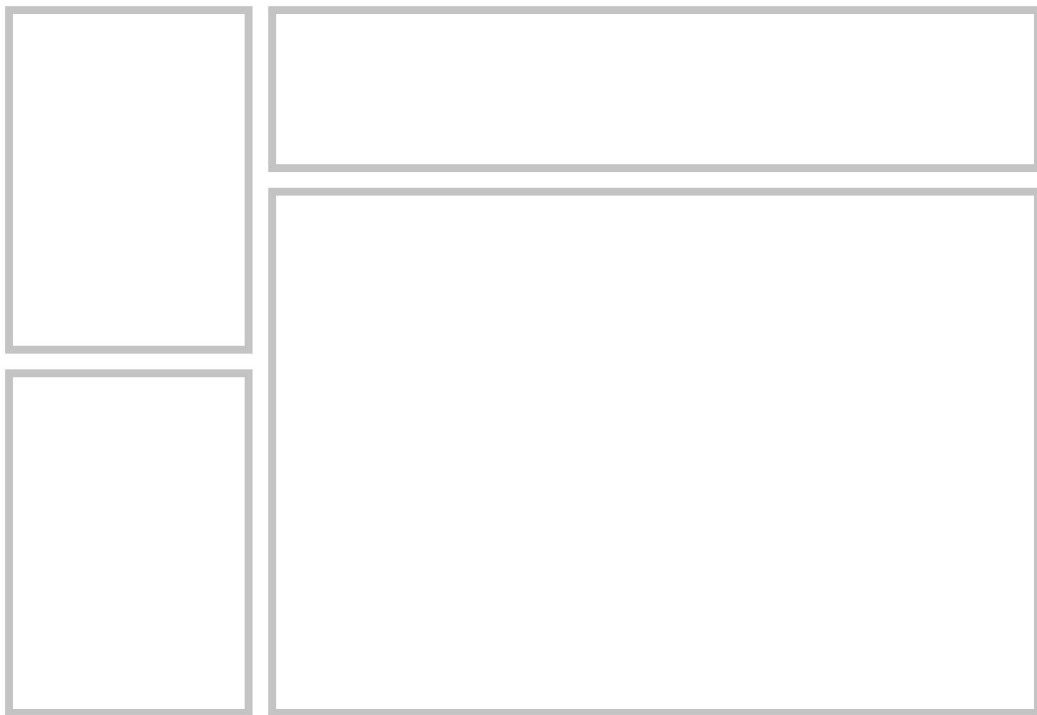
# Flexbox Ejercicios I

An empty rectangular box with a thin gray border, intended for a code snippet or diagram.An empty rectangular box with a thin gray border, intended for a code snippet or diagram.An empty rectangular box with a thin gray border, intended for a code snippet or diagram.

# Flexbox Ejercicios II



# Flexbox Ejercicios II



# Flexbox Ejercicios IV



# Flexbox Ejercicios V

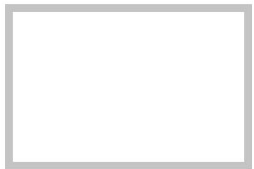


# Flexbox Ejercicios VI


# Flexbox Ejercicios VII

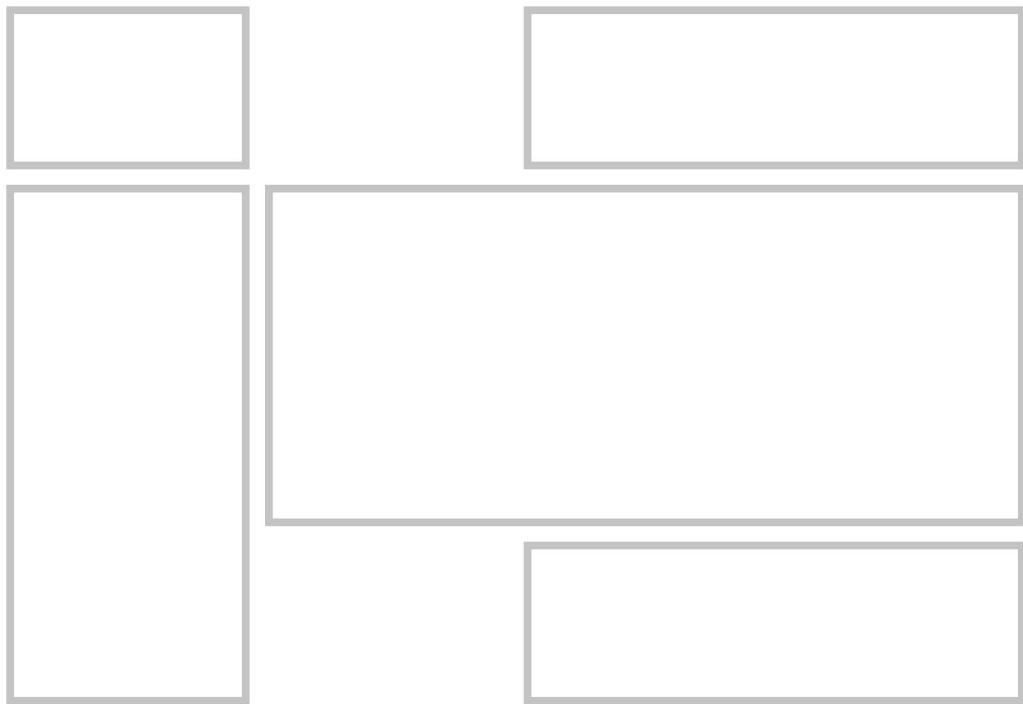


# Flexbox Ejercicios VIII





# Flexbox Ejercicios IX



# Grid

Grid es un sistema de grilla en dos dimensiones, donde se definen **filas** y **columnas**.



<https://css-tricks.com/snippets/css/complete-guide-grid/>

# Grid - Contenedor

CSS grid es un sistema de maquetación basado en grillas y se caracteriza por ser **bidimensional, independiente del orden** del markup y **flexible**.

# Grid Contenedor - Columnas

Contenedor

Con **grid-template-columns** conseguimos definir el número de columnas y el ancho de las mismas.

**grid-template-columns:**

- `<track-size> ... | <line-name> <track-size> ...;`

# Grid Contenedor - Filas

Contenedor

Con **grid-template-rows** conseguimos definir el número de filas y el ancho de las mismas.

**grid-template-rows:**

- `<track-size> ... | <line-name> <track-size> ...;`

# Grid Contenedor - repeat

Contenedor

Las partes que se repiten puedes simplificarlas con **repeat**.

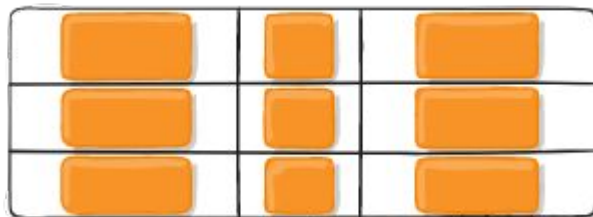
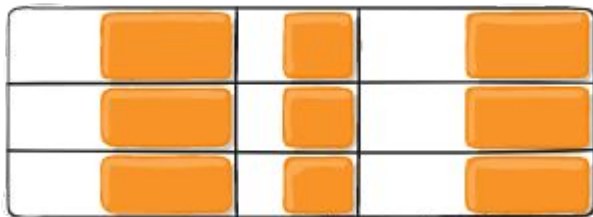
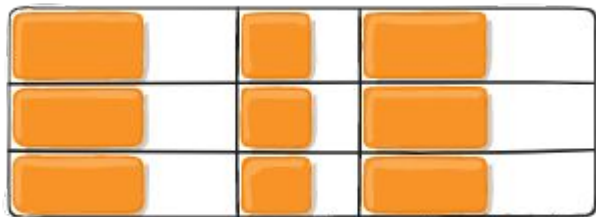
- **repeat**(3, 20px)
- **repeat**(5, 1fr 2fr)

# Grid Contenedor - Justify items

Contenedor

Con **justify-items** alineamos en el eje de las filas:

`justify-items: start | end | center | stretch;`

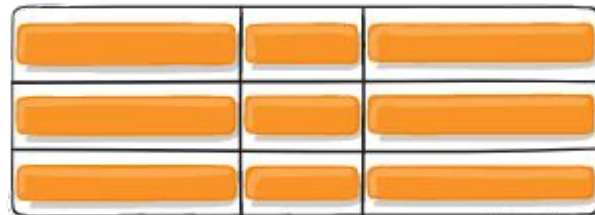
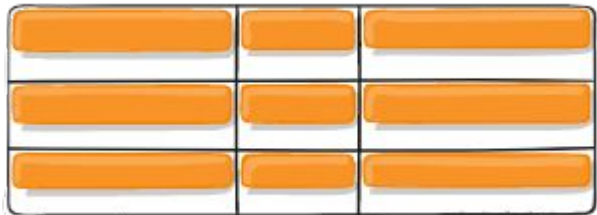


# Grid Contenedor - Align items

Contenedor

Con **align-items** alineamos en el eje de las columnas:

`justify-items: start | end | center | stretch;`



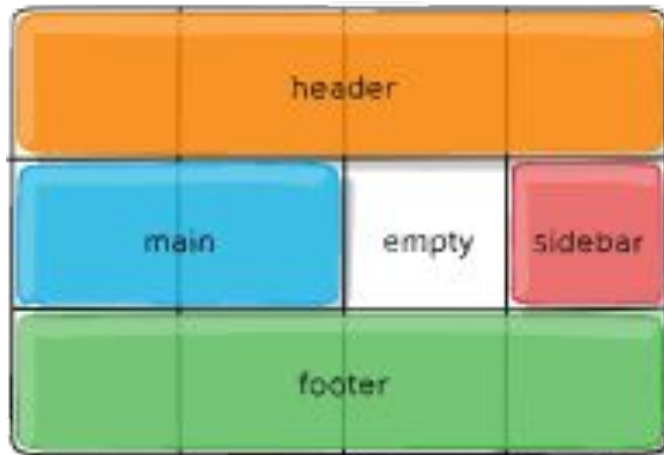


# Grid Contenedor - Areas

Contenedor

**grid-template-areas** nos permite hacer referencia a los **grid-areas** para colocar el contenido de forma “visual”

```
.container {  
  display: grid;  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```



# Grod Contenido - Grid Area

Contenido

Con **grid-area** podemos asociar un selector a un area, para posteriormente referenciarlo con **grid-template-areas**.

```
grid-area: <value>;
```

# Grid Contenido - Gap

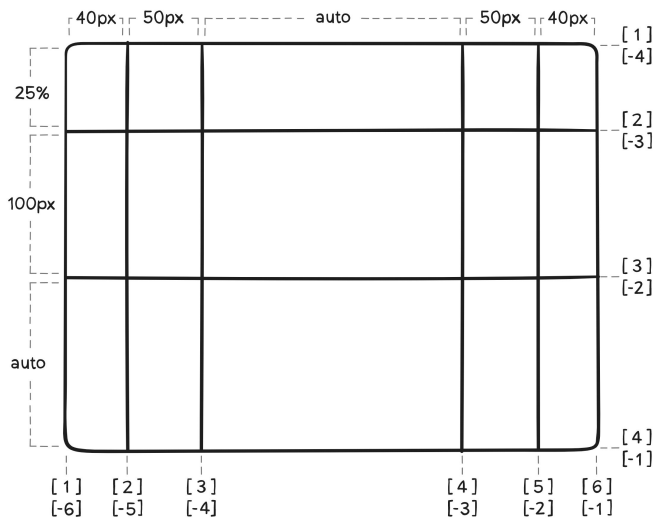
Contenido

Con la propiedad **grid-gap** podemos definir el espaciado entre las columnas o filas:

```
grid-gap: <grid-row-gap> <grid-column-gap>;
```

# Grid Contenido - Column y row

**grid-column-start**, **grid-column-end**, **grid-row-start** y **grid-row-end** determina el posicionamiento de un item de la grilla respecto al número de línea.



# **Grid contenido - justify-self**

# Grid - Funciones

- `minmax(100px, 1fr)`
- `repeat(auto-fill, minmax(200px, 1fr))`

# Grid Ejercicios I

--

--

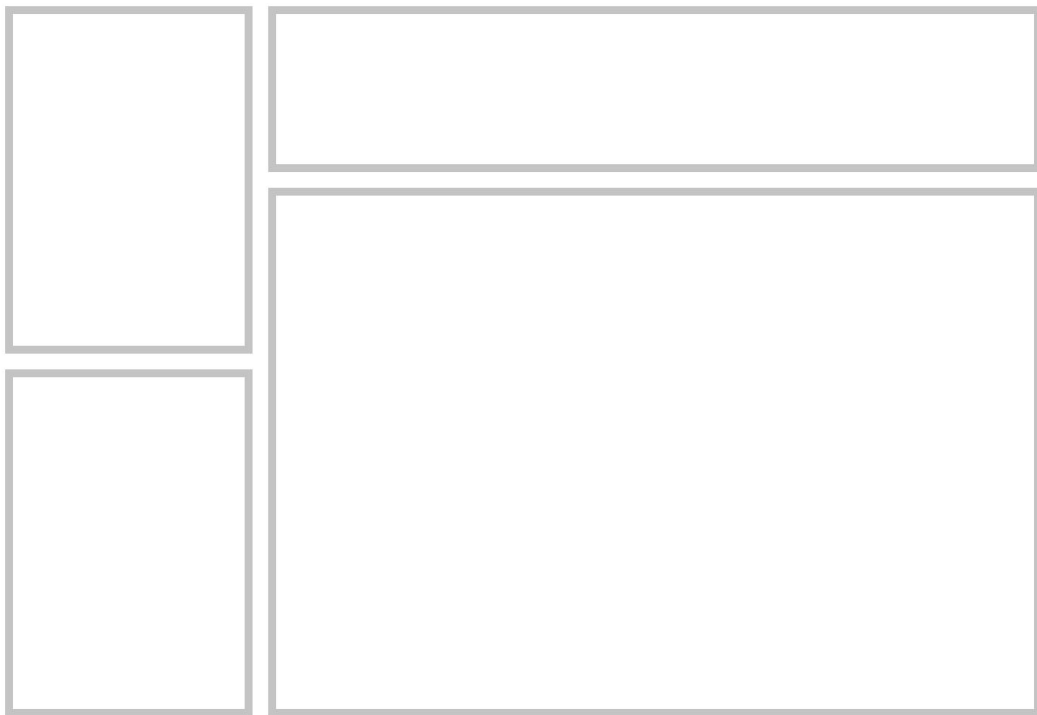
--

# Grid Ejercicios II





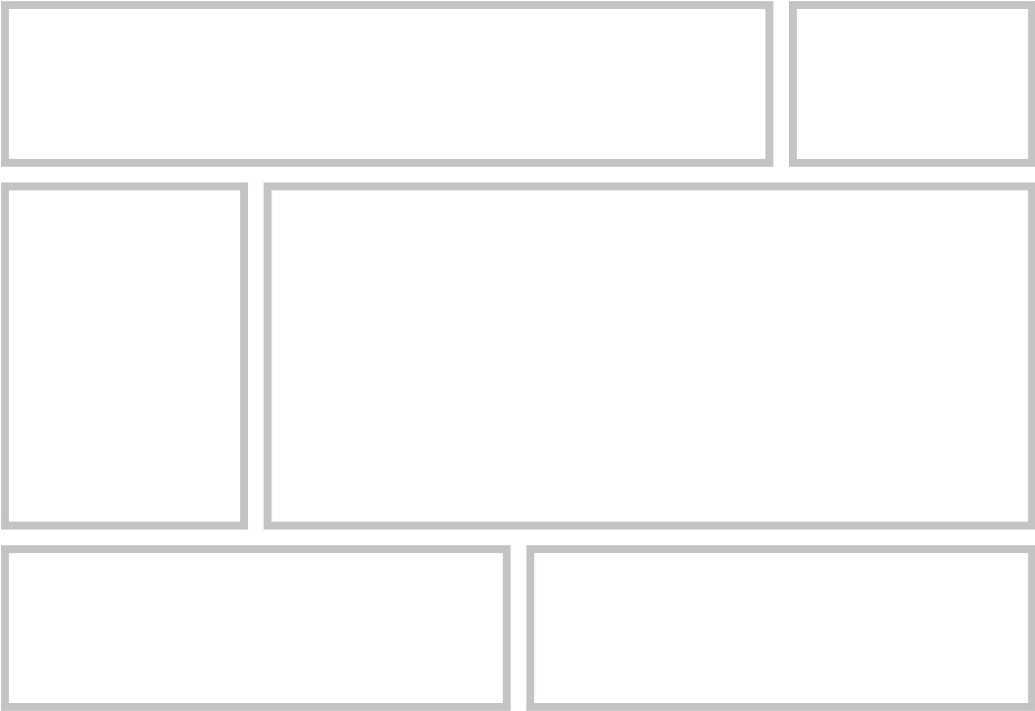
# Grid Ejercicios II



# Grid Ejercicios IV



# Grid Ejercicios V

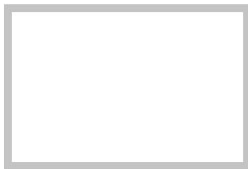


# Grid Ejercicios VI


# Grid Ejercicios VII



# Grid Ejercicios VIII



# Grid Ejercicios IX

