

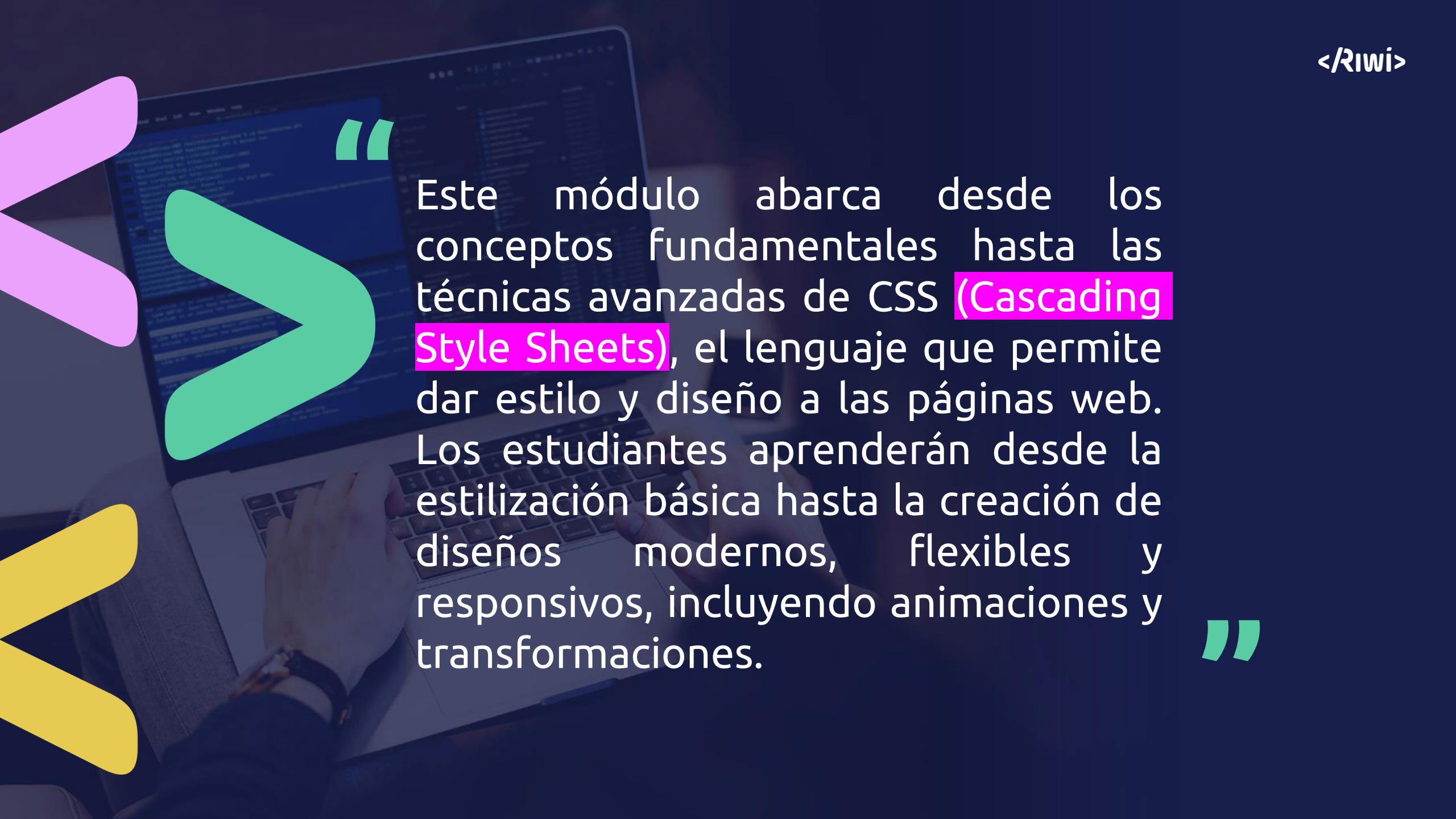


Somos un **ecosistema** de desarrolladores de software

Introducción a CSS (Cascading Style Sheets)



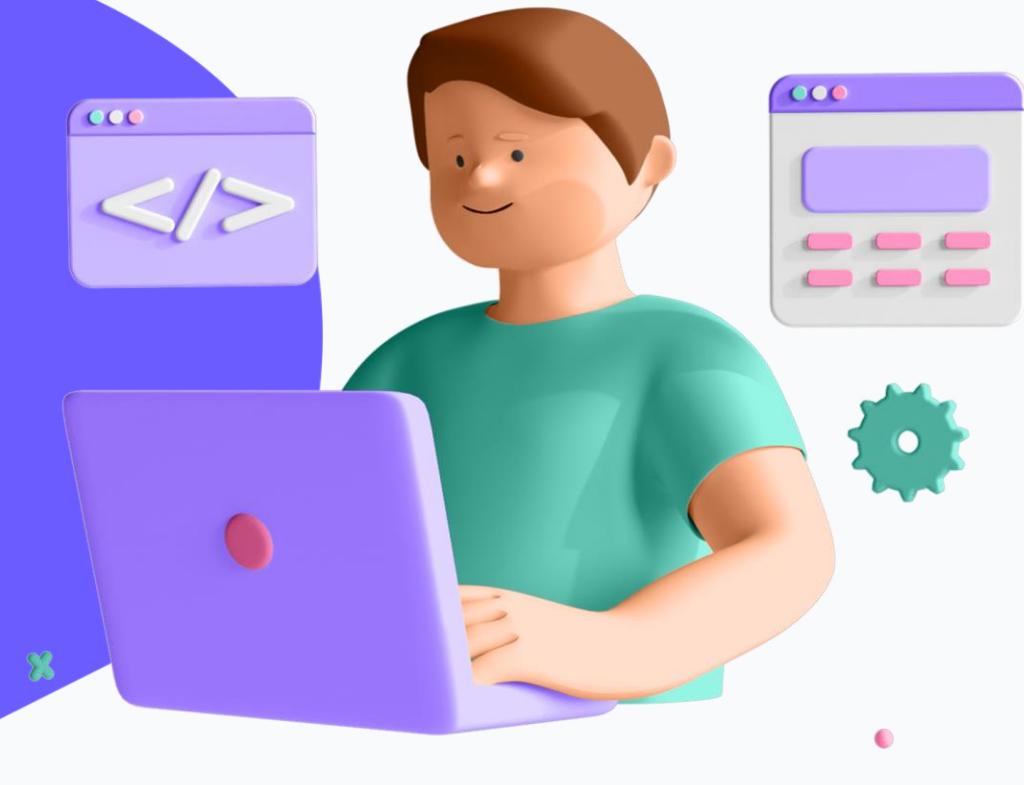
```
8"> <!--
ation">
.html">Home</a></li>
events.html">Home Events</a></li>
col-menu.html">Multiple Column Men
ren"> <a href="#" class="current">
all-button-header.html">Tall But
image-logo.html">Image Logo</a></li>
ve <a href="tall-logo.html">Ta
f="#">Carousels</a>
le-width-slider.html">Variat
le-slider.html">Testimon
```



“Este módulo abarca desde los conceptos fundamentales hasta las técnicas avanzadas de CSS (**Cascading Style Sheets**), el lenguaje que permite dar estilo y diseño a las páginas web. Los estudiantes aprenderán desde la estilización básica hasta la creación de diseños modernos, flexibles y responsivos, incluyendo animaciones y transformaciones.”

¿Qué es CSS?

Hojas de estilo en cascada, que permitirán darle a tu código HTML un diseño único y especial en pocas líneas.



CSS

¿Porque se usa CSS?

- Los **documentos HTML (contenido)** incluirán sólo información y datos, todo lo relativo a la información a transmitir.
- Los **documentos CSS (presentación)** incluirán sólo los aspectos relacionados con el estilo (diseño, colores, formas, etc...).



CSS

¿Porque se usa CSS?

</Puni>
</Riwi>

HTML



HTML the Skeleton



CSS



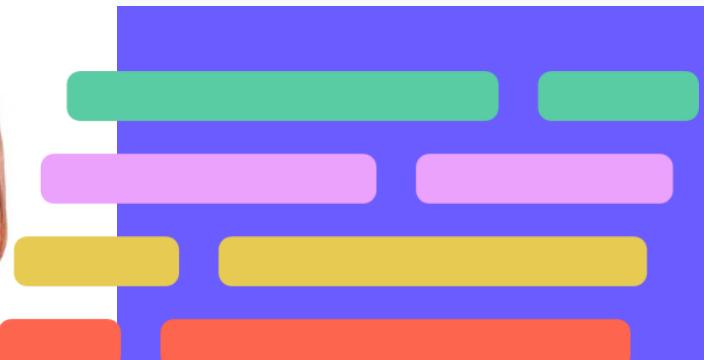
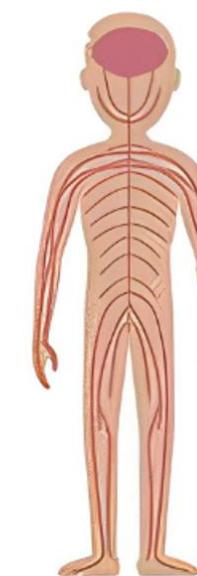
CSS the Skin

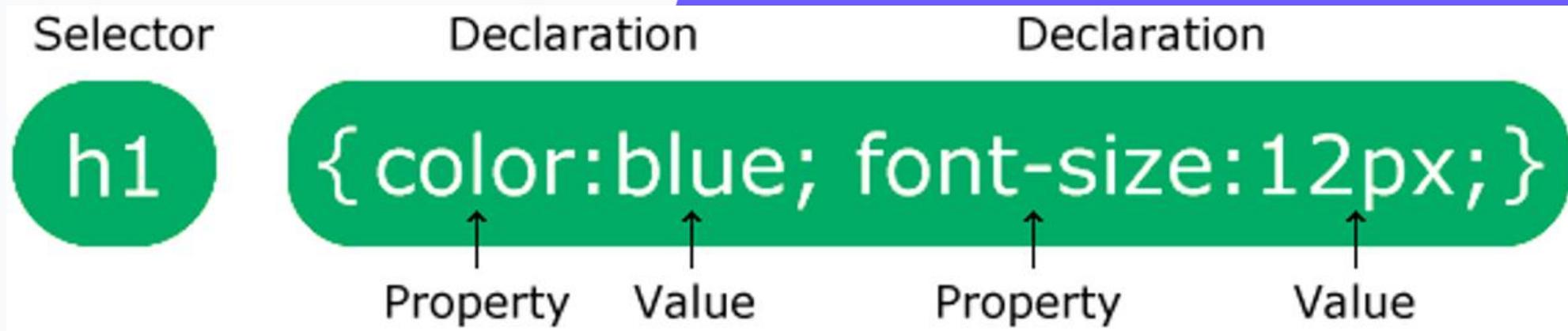


JS



Javascript the Brain





SELECTOR DE ELEMENTO



- Seleccionará un elemento en base a su etiqueta:

```
p {  
    color:blue;  
}
```

SECTOR DE CLASE

Una clase se puede crear múltiples veces (es reutilizable) e inicia con un punto.

```
.cliente {  
    color: blue;  
}
```

SELECTOR ID

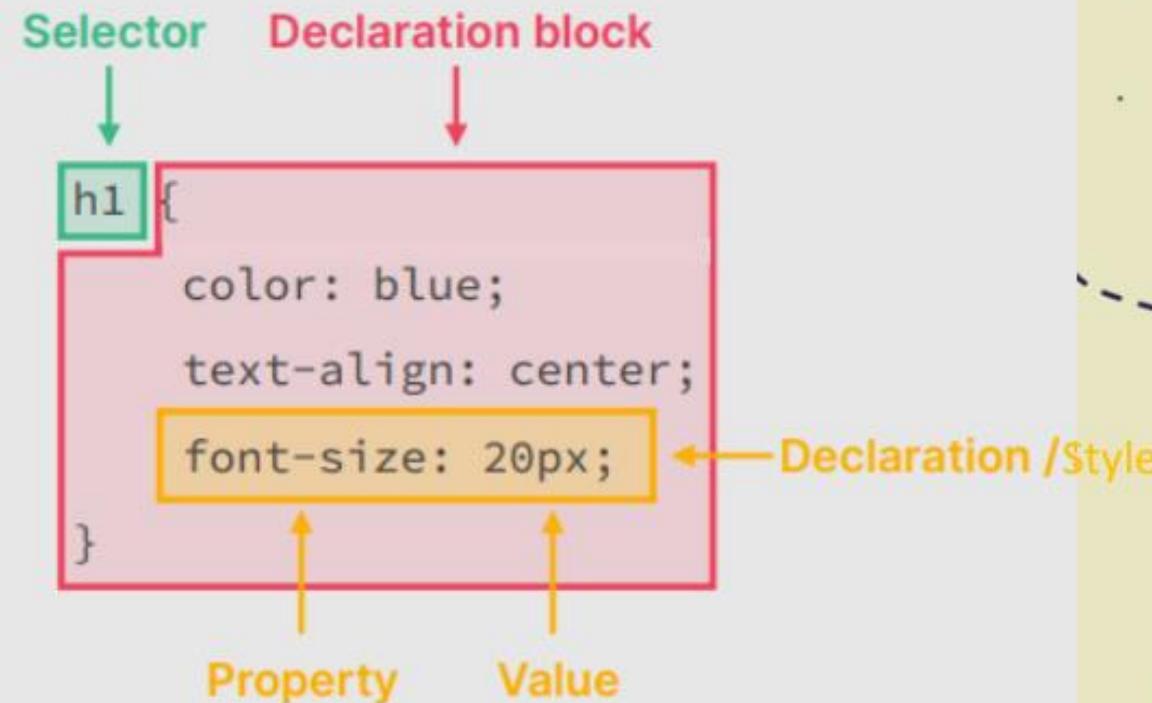
Puedes tener múltiples ID`s por página pero sus.
nombres no pueden repetirse.

```
#cliente {  
    color: blue;  
}
```

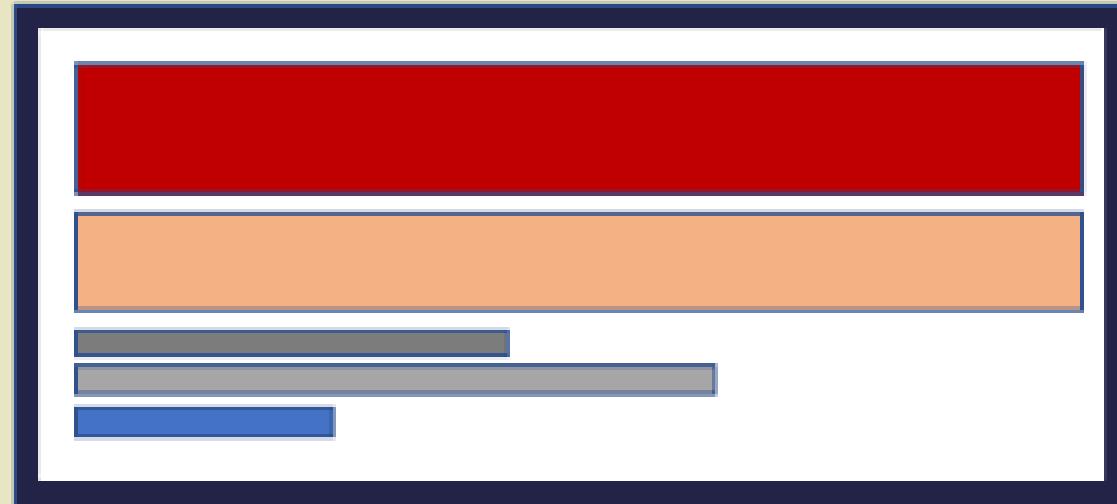
Nombre	Mediante...	Descripción
Archivo CSS externo	Etiqueta <code><link rel="stylesheet"></code>	El código se escribe en un archivo <code>.css</code> a parte. Recomendado.
Bloque de estilos	Etiqueta <code><style></code>	El código se escribe en una etiqueta <code><style></code> en el documento HTML.
Estilos en línea	Atributo HTML <code>style="..."</code>	El código se escribe en un atributo HTML <code>style</code> en una etiqueta.

¿Cómo seleccionamos y estilizamos los elementos?

- A CSS RULE



Significa que el elemento se colocará uno debajo del otro sin importar su tamaño o contenido...



Significa que el elemento se posicionará a la derecha una vez que haya tomado el espacio que requiere...

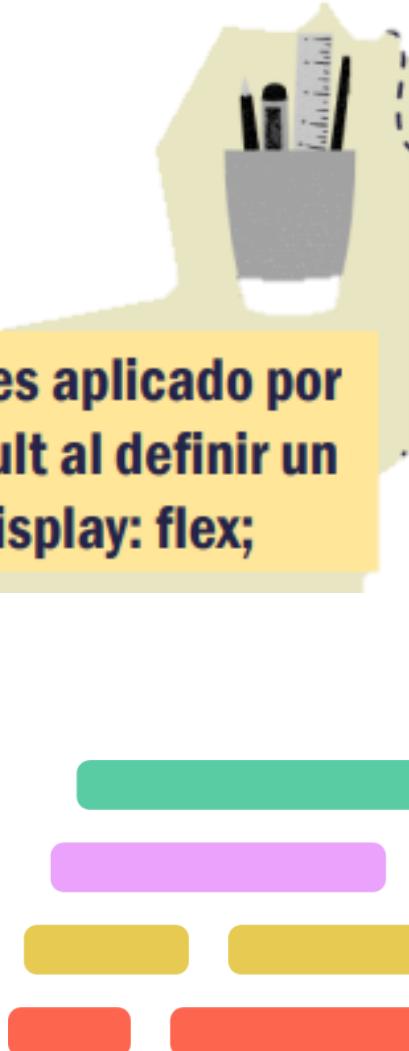
ENLACE 1 ENLACE 2 ENLACE 3 ENLACE 4

Ningún elemento tiene un display: flex, o display: grid; por default

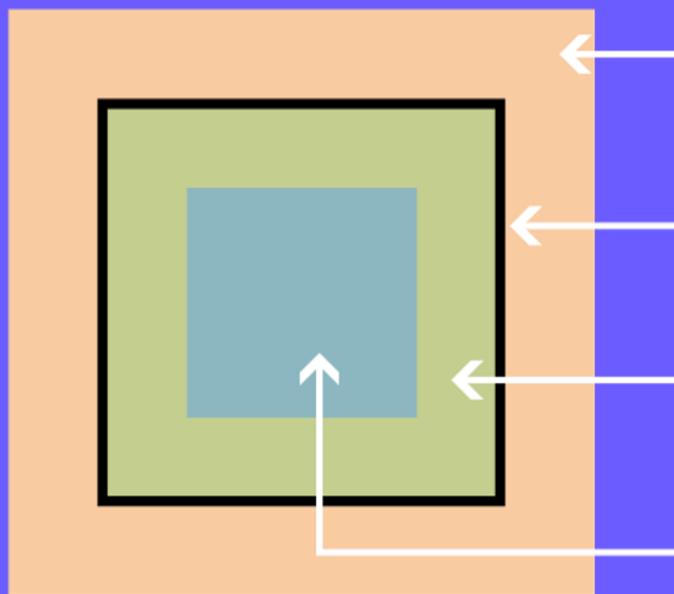
En Flexbox solo puedes colocar y distribuir tus elementos en una dirección fila (row) o columna (column)

Row es aplicado por default al definir un display: flex;

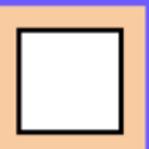
Flexbox fue diseñado como un modelo unidimensional para crear layouts.



Modelo de cajas



margin



border



padding



content



- El **márgen (margin)**, en naranja, es la parte exterior del elemento, por fuera del borde.

- El **borde (border)**, en negro, es el límite que separa el interior del exterior del elemento.

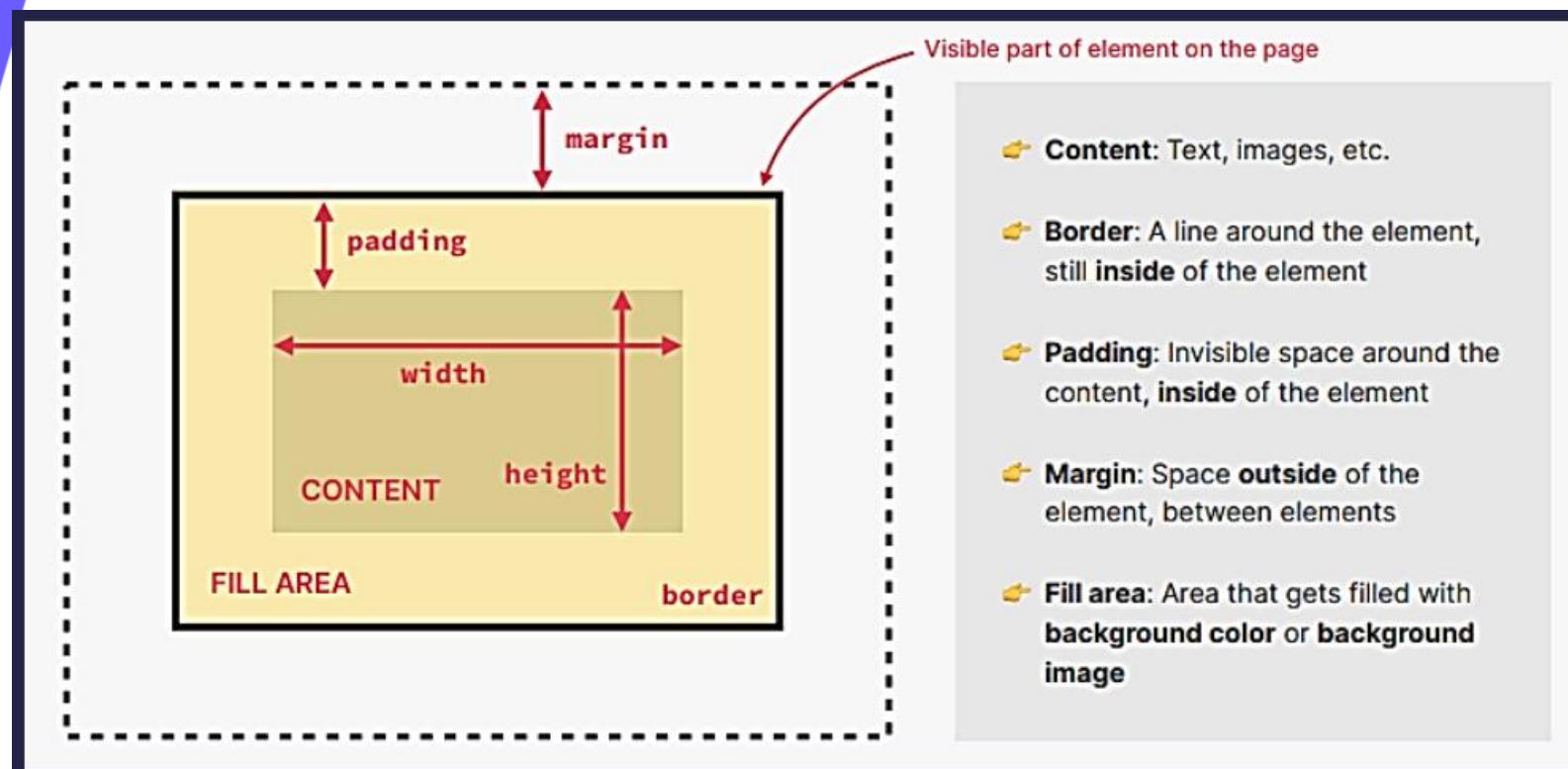
- El **relleno (padding)**, en verde, es la parte interior del elemento, entre el contenido y el borde.

- El **contenido**, en azul, es la parte interior del elemento, excluyendo el relleno.

Modelo de cajas

Dimensiones

Propiedad	Valor	Significado
width	Auto	Tamaño de ancho de un elemento
height	Auto	Tamaño de alto de un elemento



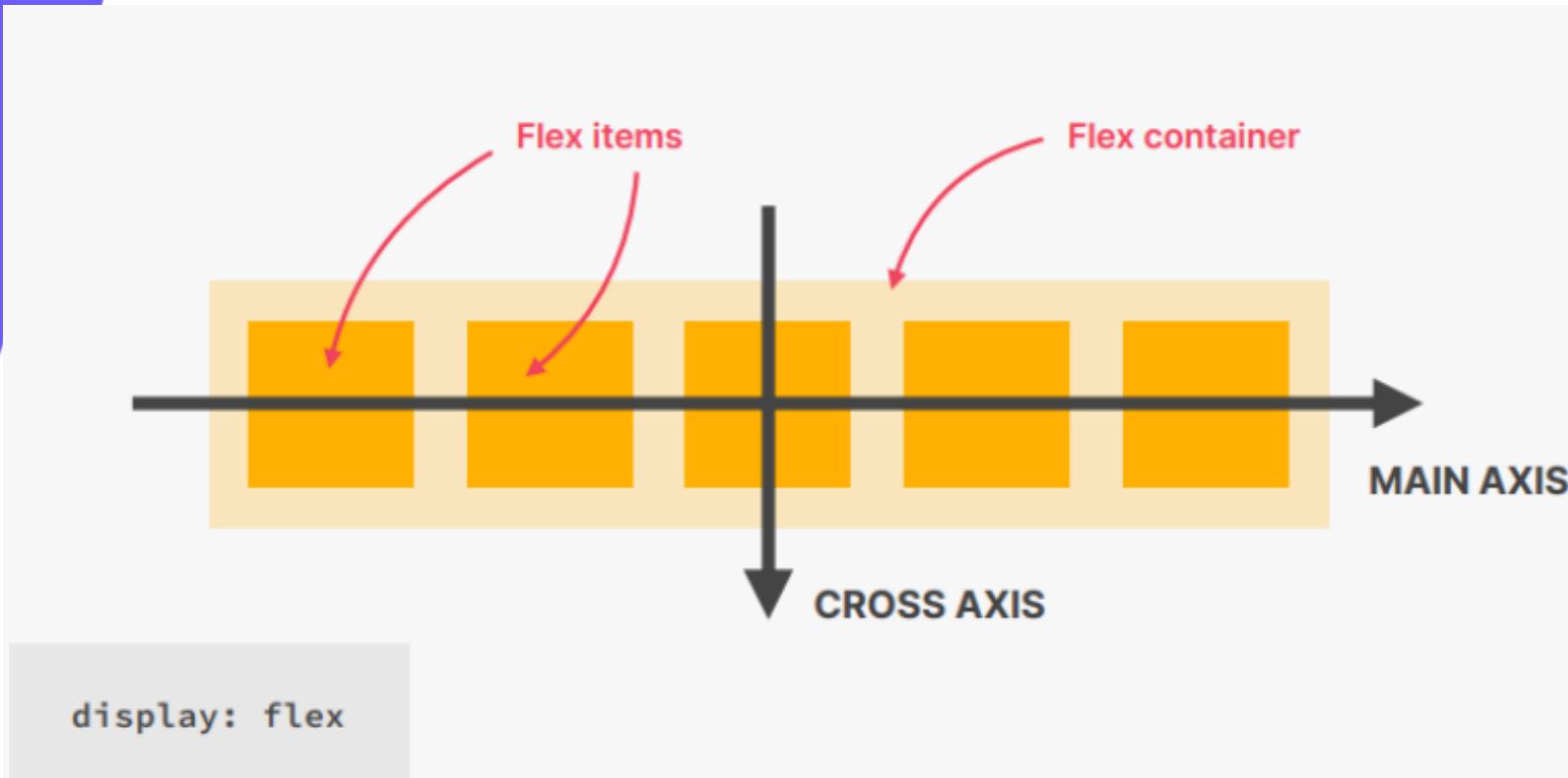
Posición absoluta

The diagram illustrates the concept of absolute positioning. A yellow rectangular box labeled ".el" is positioned absolutely within a larger orange rectangular container labeled ".container". The ".el" box has a width of 200px and a height of 100px, indicated by double-headed arrows. It is positioned 100px from the top edge and 200px from the left edge of its container. The ".el" box contains the following CSS code:

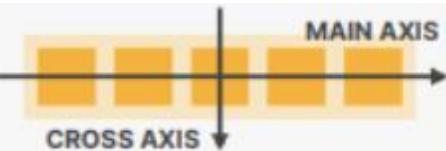
```
.el {  
    position: absolute;  
    top: 100px;  
    left: 200px;  
    background-color: #f4b33f;  
}
```

```
.container {  
    position: relative;  
    background-color: #f7e6c1;  
}
```

- 👉 Flexbox is a set of related **CSS properties** for building 1-dimensional layouts
- 👉 The main idea behind flexbox is that empty space inside a container element can be **automatically divided** by its child elements
- 👉 Flexbox makes it easy to automatically **align items to one another** inside a parent container, both horizontally and vertically
- 👉 Flexbox solves common problems such as **vertical centering** and creating **equal-height columns**
- 👉 Flexbox is perfect for **replacing floats**, allowing us to write fewer and cleaner HTML and CSS code



FLEX CONTAINER



FLEX ITEMS

1 `gap: 0 | <length>`

👉 To create **space between items**, without using margin

2 `justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly`

👉 To align items along main axis (**horizontally**, by default)

3 `align-items: stretch | flex-start | flex-end | center | baseline`

👉 To align items along cross axis (**vertically**, by default)

4 `flex-direction: row | row-reverse | column | column-reverse`

👉 To define which is the **main axis**

5 `flex-wrap: nowrap | wrap | wrap-reverse`

👉 To allow items to **wrap into a new line** if they are too large

6 `align-content: stretch | flex-start | flex-end | center | space-between | space-around`

👉 Only applies when there are **multiple lines** (`flex-wrap: wrap`)

1 `align-self: auto | stretch | flex-start | flex-end | center | baseline`

👉 To **overwrite align-items** for individual flex items

2 `flex-grow: 0 | <integer>`

👉 To allow an element **to grow** (0 means no, 1+ means yes)

3 `flex-shrink: 1 | <integer>`

👉 To allow an element **to shrink** (0 means no, 1+ means yes)

4 `flex-basis: auto | <length>`

👉 To define an item's width, **instead of the width property**

5 `flex: 0 1 auto | <int> <int> <len>`

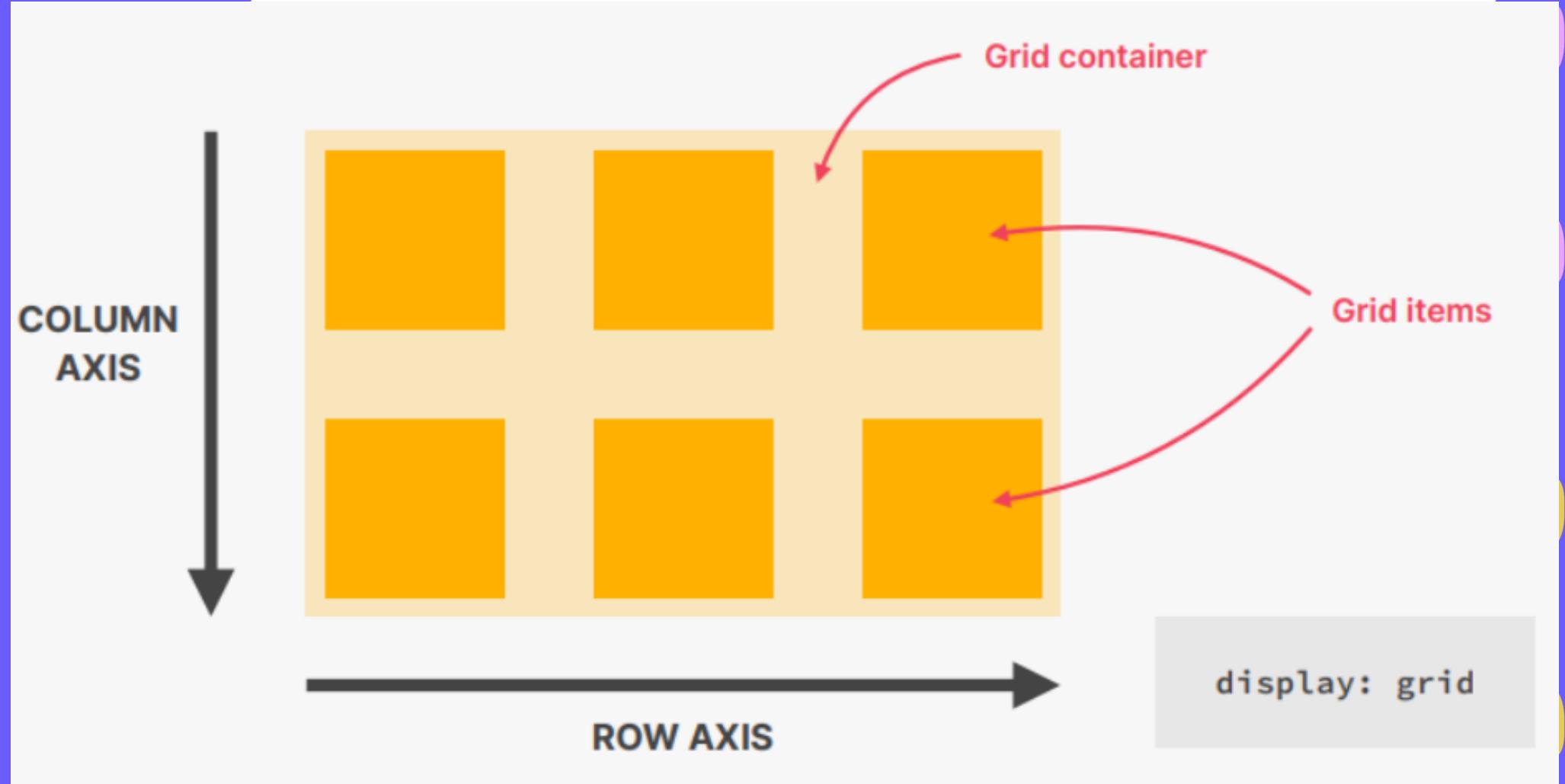
👉 Recommended shorthand for `flex-grow`, `-shrink`, `-basis`.

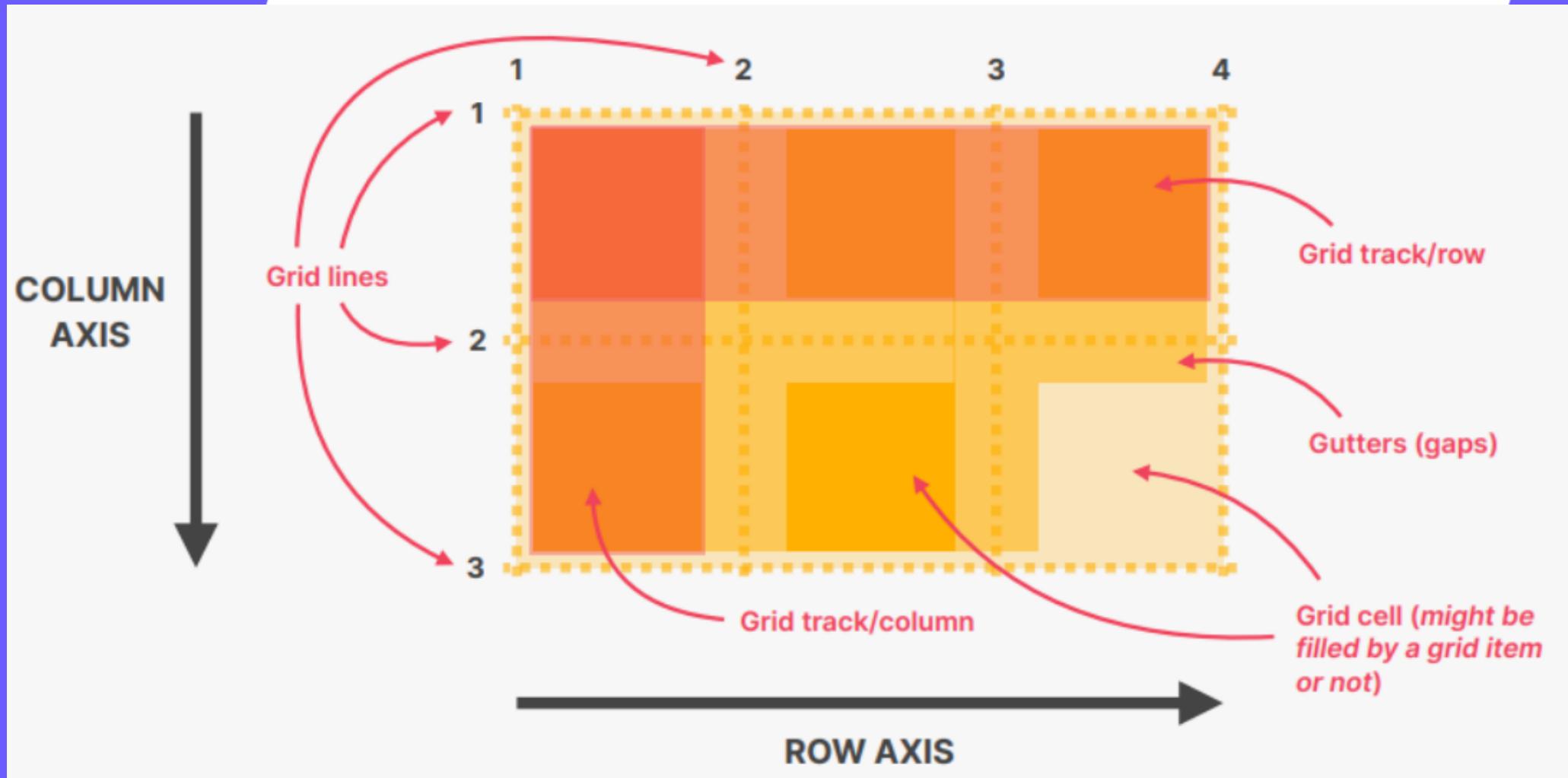
6 `order: 0 | <integer>`

👉 Controls order of items. -1 makes item **first**, 1 makes it **last**

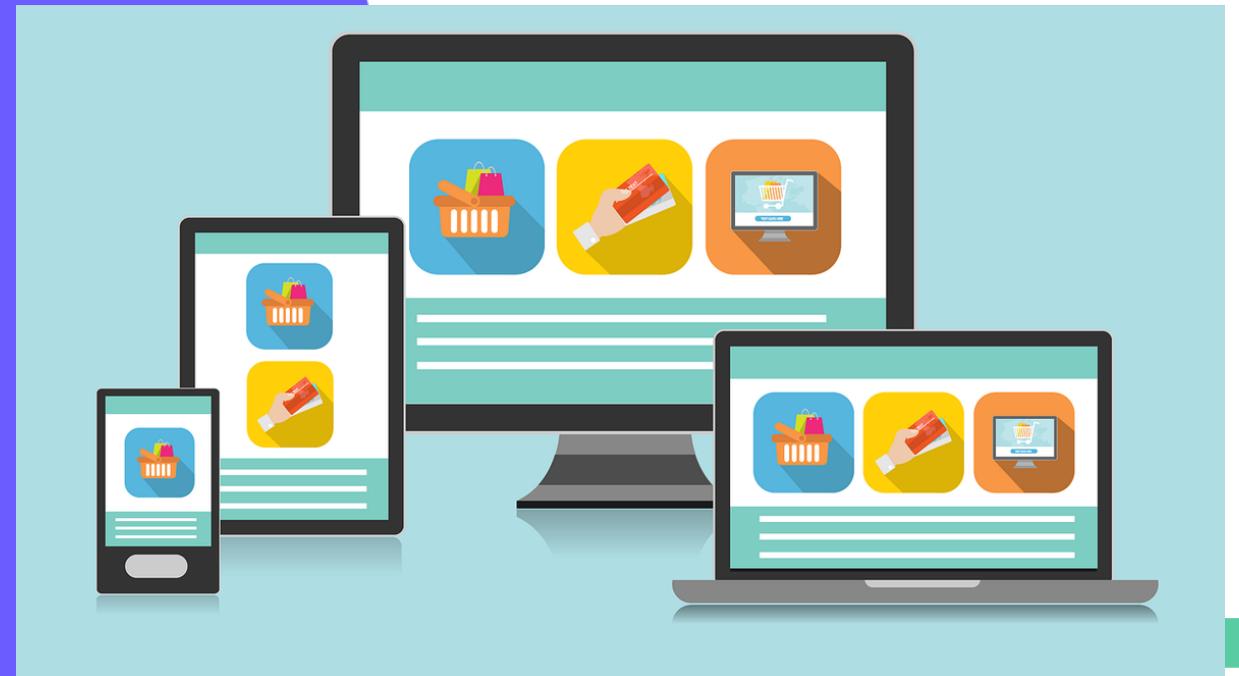
- 👉 CSS Grid is a set of **CSS properties** for **building 2-dimensional layouts**
- 👉 The main idea behind CSS Grid is that we **divide a container element into rows and columns** that can be filled with its child elements
- 👉 In two-dimensional contexts, CSS Grid allows us to write **less nested HTML** and **easier-to-read CSS**
- 👉 CSS Grid is **not meant to replace flexbox!** Instead, they work perfectly together. Need a **1D** layout? Use flexbox. Need a **2D** layout? Use CSS Grid.







Es un enfoque que nos dice que nuestros diseños deberían adaptarse a las interacciones del usuario y la resolución que utilizan. Los sitios web deberían adaptarse a celulares, tablets, laptops, computadores de escritorio, televisores.



CSS

¿Cómo logro que mi sitio sea responsive?

Media Queries

```
@media (min-width: 768px){  
}  
  
@media (min-width: 992px){  
}
```

Se debe ubicar en el bloque `<head></head>`

Se incluye la etiqueta `<link>`, con la que establecemos relación entre el **HTML** y el archivo **.css** como **index.css**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE-edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hipervinculos</title>
    <link rel="stylesheet" href="index.css" />
</head>
```

CSS

Bloque de estilos

</Riwi>

```
<!DOCTYPE html>
<html>
<head>
    <title>Título de la página</title>
    <style>
        div {
            background: hotpink;
            color: white;
        }
    </style>
</head>
...
</html>
```

CSS

estilos en linea

</Riwi>

```
<p>¡Hola <span style="color: red; padding: 8px">amigo lector</span>!</p>
```



Concepto	Descripción
Selector	Elemento o elementos del documento que vamos a seleccionar para aplicarle un estilo concreto.
Propiedad	Característica principal que vamos a definir con el valor indicado a continuación.
Valor	Cada propiedad tiene una serie de valores concretos asignables, con los que tendrá uno u otro comportamiento.
Comentario	Fragmento de texto entre /* y */ con anotaciones o aclaraciones para el desarrollador (el navegador los ignorará).
Regla	Par de propiedad y su respectivo valor asociado.

```
selector {  
    propiedad : valor ;  
    propiedad : valor ;  
    propiedad : valor ;  
}
```

UNA REGLA POR LÍNEA

terminada con punto y coma

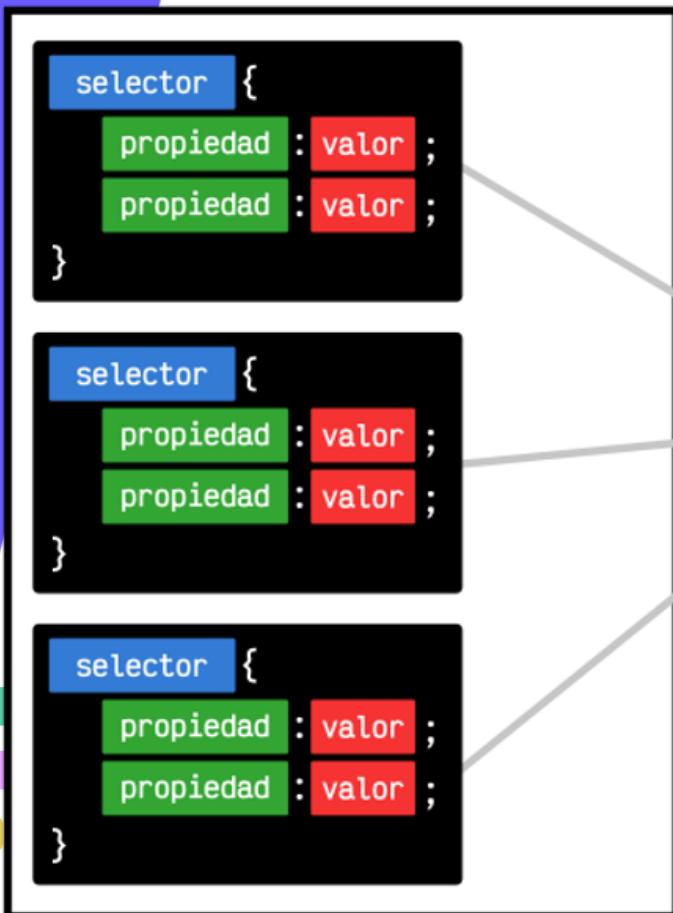
TERMINACIÓN DE REGLA

opcional (al ser la última)

CSS

Documento CSS

</Riwi>



DOCUMENTO CSS

CONJUNTO DE REGLAS CSS

CSS

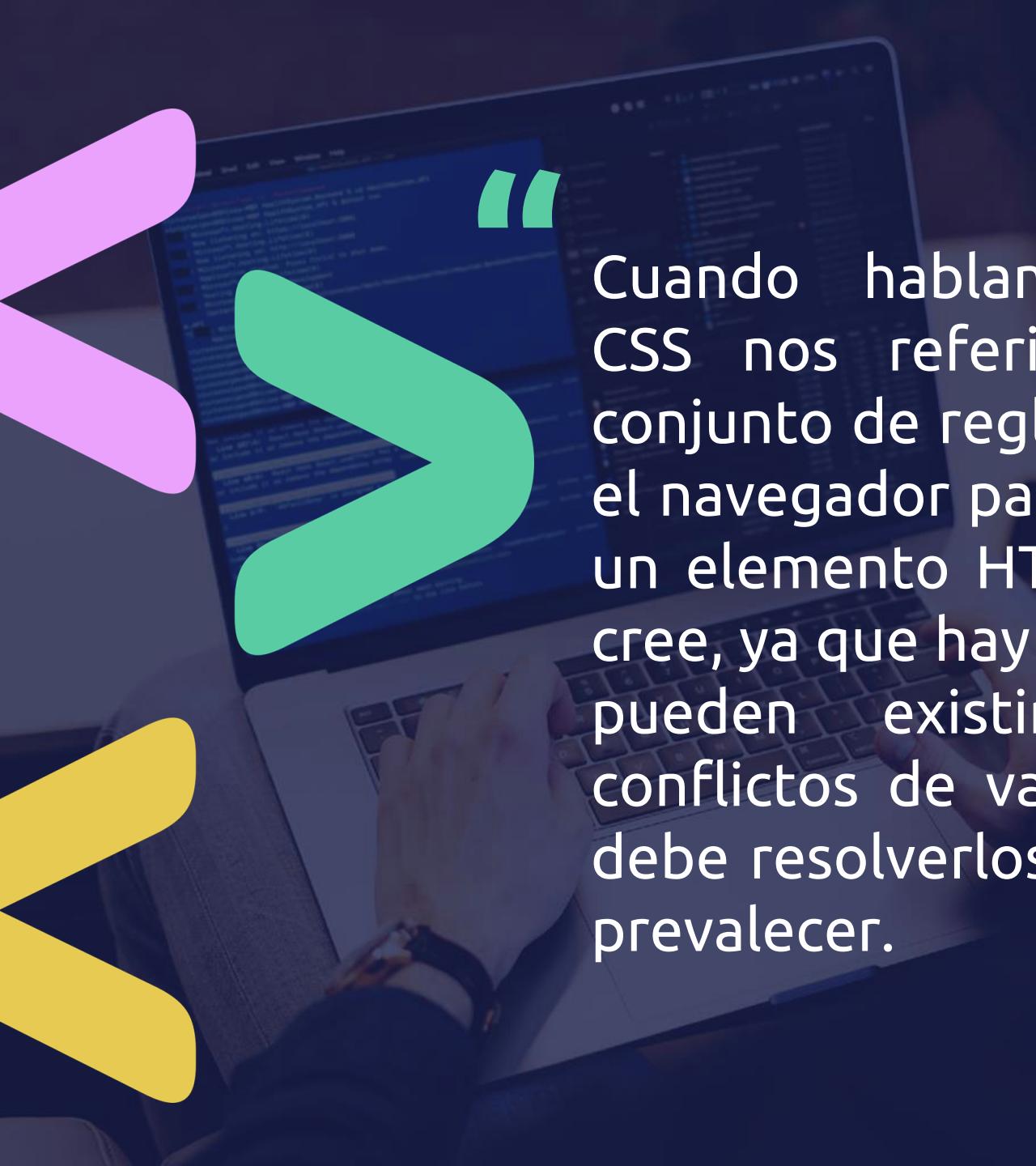
Herencia

</Riwi>

Algunas propiedades CSS, como **color** o **font-family**, se **heredan** desde los elementos HTML padres a los elementos HTML hijos, modificando el valor que tienen por defecto.

```
<div class="container">
  <p>Texto del parent</p>
  <div class="child">
    <p>Texto del hijo</p>
  </div>
</div>
<style>
  .container {
    color: green;          /* Propiedad heredable */
    border: 1px solid red; /* Propiedad no heredable */
  }
</style>
```

Texto del parent
Texto del hijo



“ Cuando hablamos de la Cascada CSS nos referimos al algoritmo o conjunto de reglas y normas que tiene el navegador para aplicar estilos CSS a un elemento HTML. Como la mayoría cree, ya que hay múltiples casos donde pueden existir ambigüedades o conflictos de valores, y el navegador debe resolverlos para saber cual debe prevalecer.

”

La **cascada** comprende varios temas importantes a la hora de resolver dichos conflictos:

Cascada	Descripción
Importancia	Los estilos pueden tener una importancia concreta, eso le da mayor o menor prioridad.
Origen	Existen varios orígenes de CSS, cada uno con una prioridad específica.
Orden de aparición	El orden en el que se procesa el CSS influye a la hora de resolver conflictos
Especificidad	Cuanto más específicos sean los estilos CSS, mayor prioridad
Capas	Los estilos CSS se pueden aislar en capas fusionables.

!important

```
<div class="text">Texto del elemento</div><style>
  .text {
    color: red !important;
  }

  .text {
    color: blue;
  }
</style>
```

- ① Atributo `style` en una etiqueta HTML.
- ② Bloque de estilos `

Variables

Para definir una **custom property** haremos uso de los dos guiones -- previos al nombre que queramos utilizar.

```
Comment Code |  
:root {  
    --background-color: black;  
}  
  
Comment Code  
.element {  
    background: var(--background-color);  
}
```

</Be a
coder>