

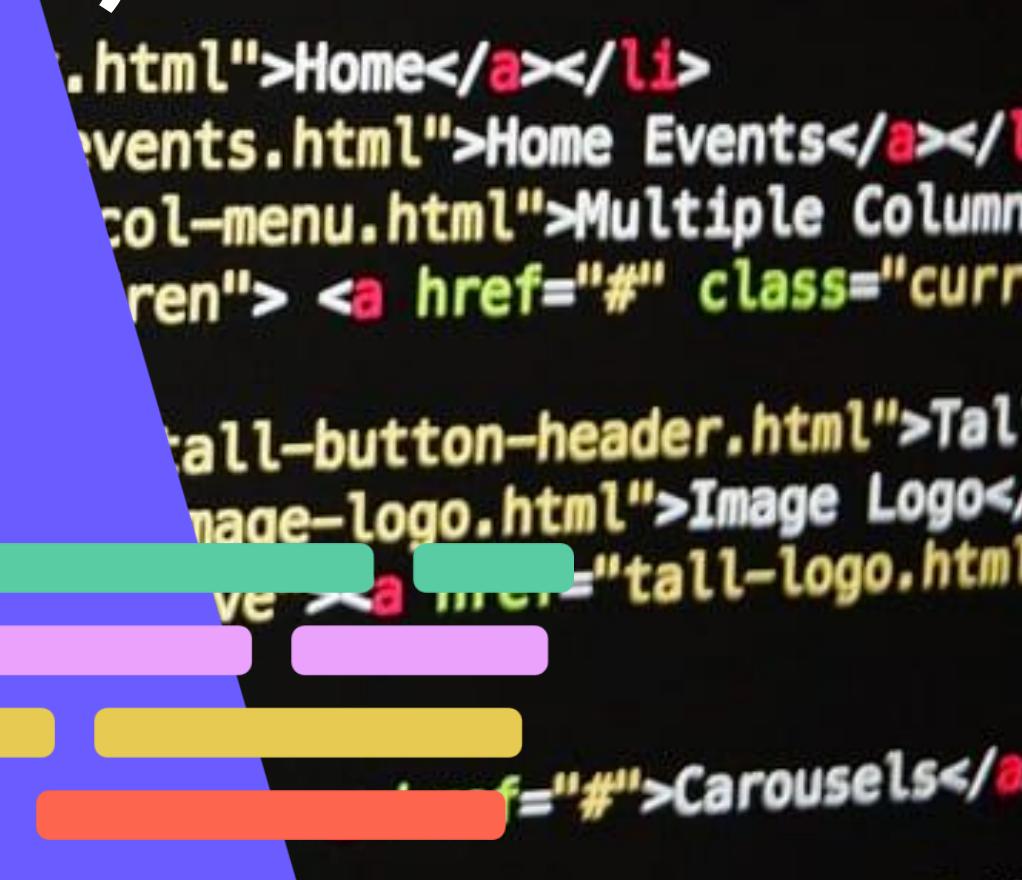


Somos un **ecosistema** de desarrolladores de software

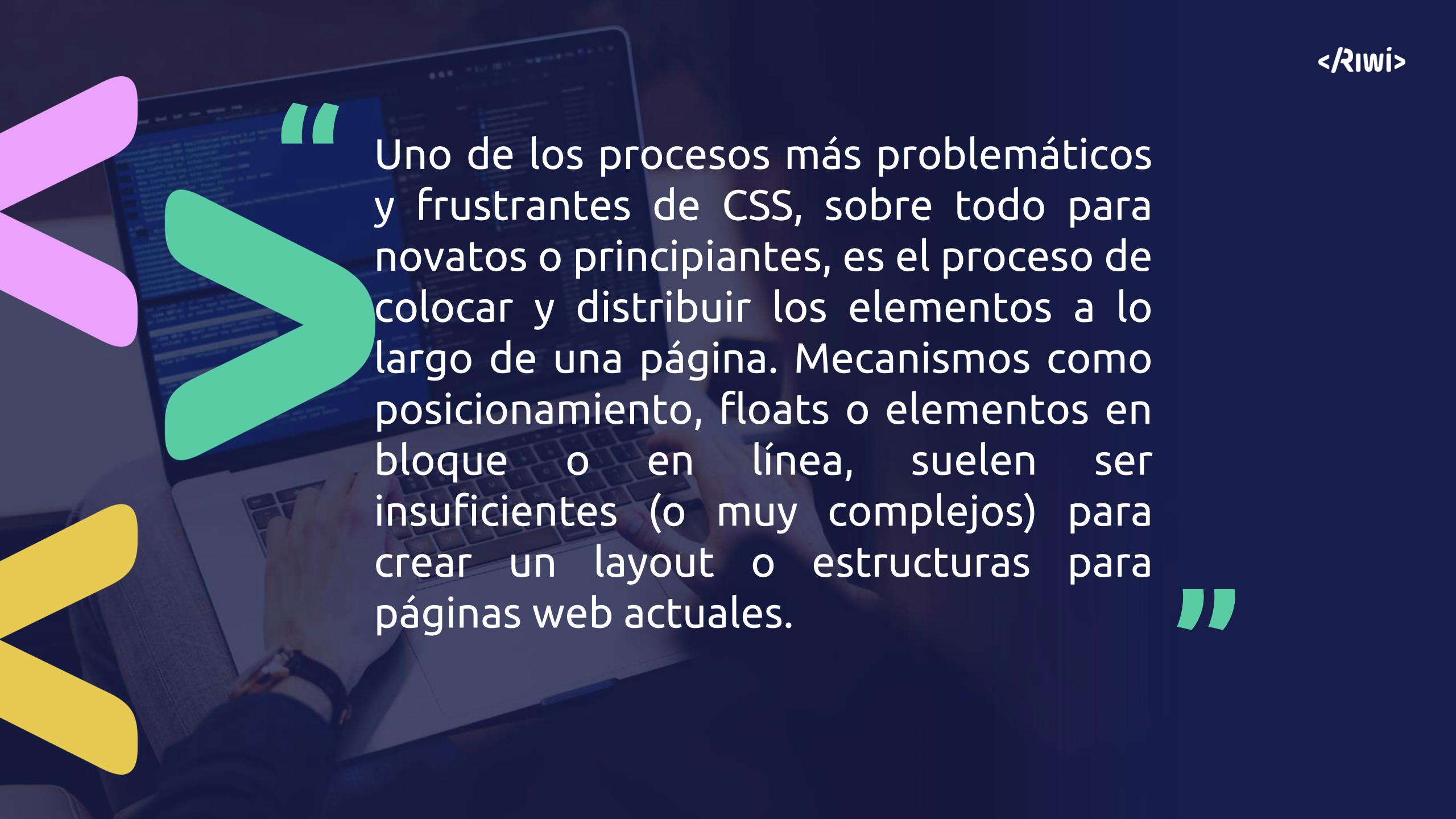
Introducción a CSS (Cascading Style Sheets)



```
<8"> <!-- BEGIN NAVIGATION -->
navigation">
eets)
    .html">Home</a></li>
    events.html">Home Events</a></li>
    col-menu.html">Multiple Column Men
    ren"> <a href="#" class="current">
        tall-button-header.html">Tall But
        mage-logo.html">Image Logo</a></li>
        ve <a href="#">Tall Logo</a>
    
```



```
<!-- END NAVIGATION -->
    f="#">Carousels</a>
    le-width-slider.html">Variat
    e-width-slider.html">Testimon
```

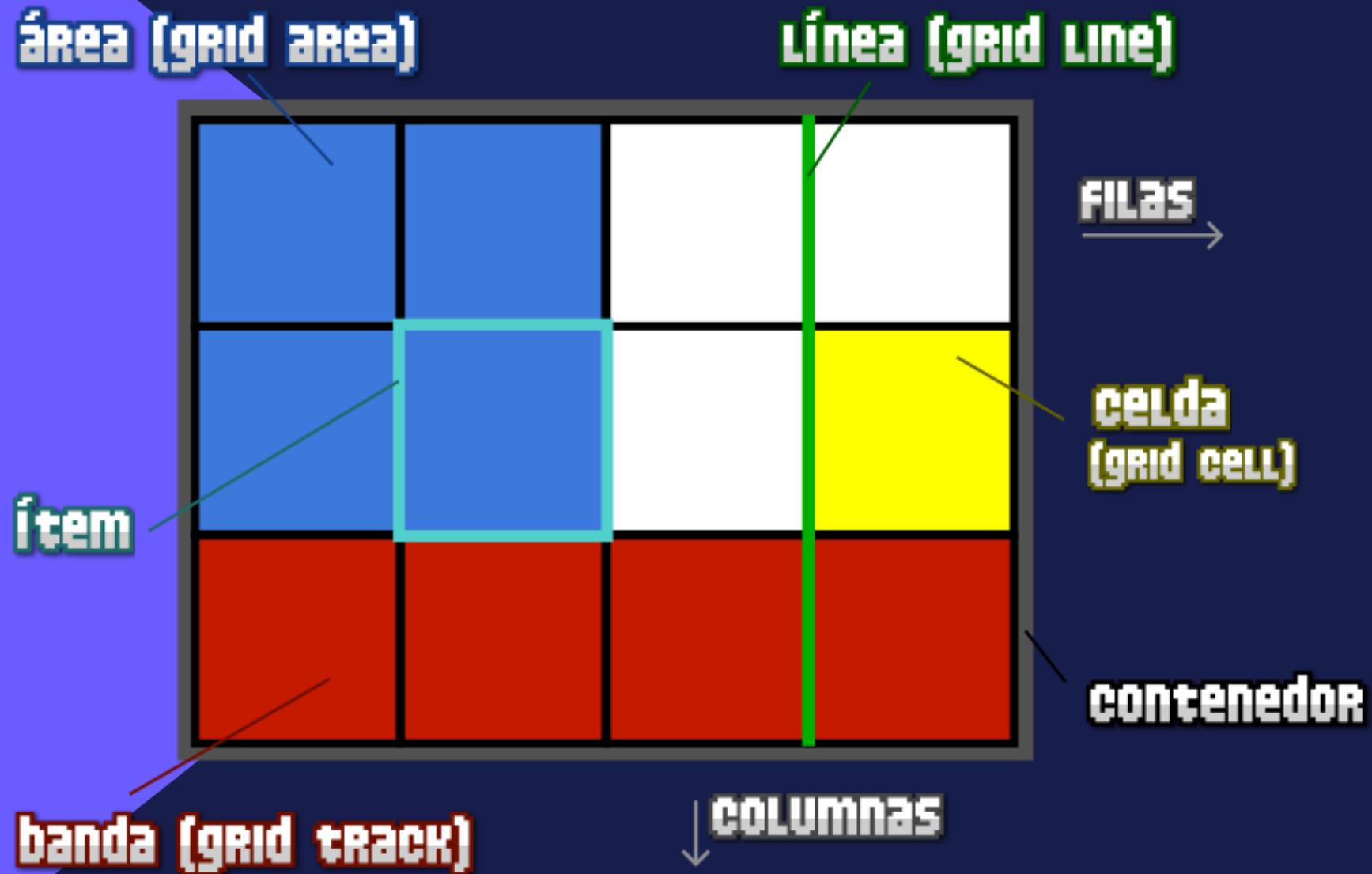


“ Uno de los procesos más problemáticos y frustrantes de CSS, sobre todo para novatos o principiantes, es el proceso de colocar y distribuir los elementos a lo largo de una página. Mecanismos como posicionamiento, floats o elementos en bloque o en línea, suelen ser insuficientes (o muy complejos) para crear un layout o estructuras para páginas web actuales.

”

Grid

Propiedades



Grid

Propiedades

- **Contenedor:** El elemento padre contenedor que definirá la cuadrícula o rejilla.
- **Ítem:** Cada uno de los hijos que contiene la cuadrícula (elemento contenedor).
- **Celda (grid cell):** Cada uno de los cuadritos (unidad mínima) de la cuadrícula.
- **Área (grid area):** Región o conjunto de celdas de la cuadrícula.
- **Banda (grid track):** Banda horizontal o vertical de celdas de la cuadrícula.
- **Línea (grid line):** Separador horizontal o vertical de las celdas de la cuadrícula.

Grid

Propiedades

Tipo de elemento	Descripción
inline-grid	Establece una cuadrícula con ítems en línea , de forma equivalente a inline-block.
grid	Establece una cuadrícula con ítems en bloque , de forma equivalente a block.

Propiedad	Valor	Descripción
grid-template-columns	[col1] [col2] ...	Establece el SIZE de cada columna (col 1, col 2...).
grid-template-rows	[fila1] [fila2] ...	Establece el SIZE de cada fila (fila 1, fila 2...).

Grid

Filas y columnas

Comment Code

```
.grid {  
  display: grid;  
  grid-template-columns: 50px 300px;  
  grid-template-rows: 200px 75px;  
}
```

grid-template-rows

grid-template-columns

50px

300px

200px

75px



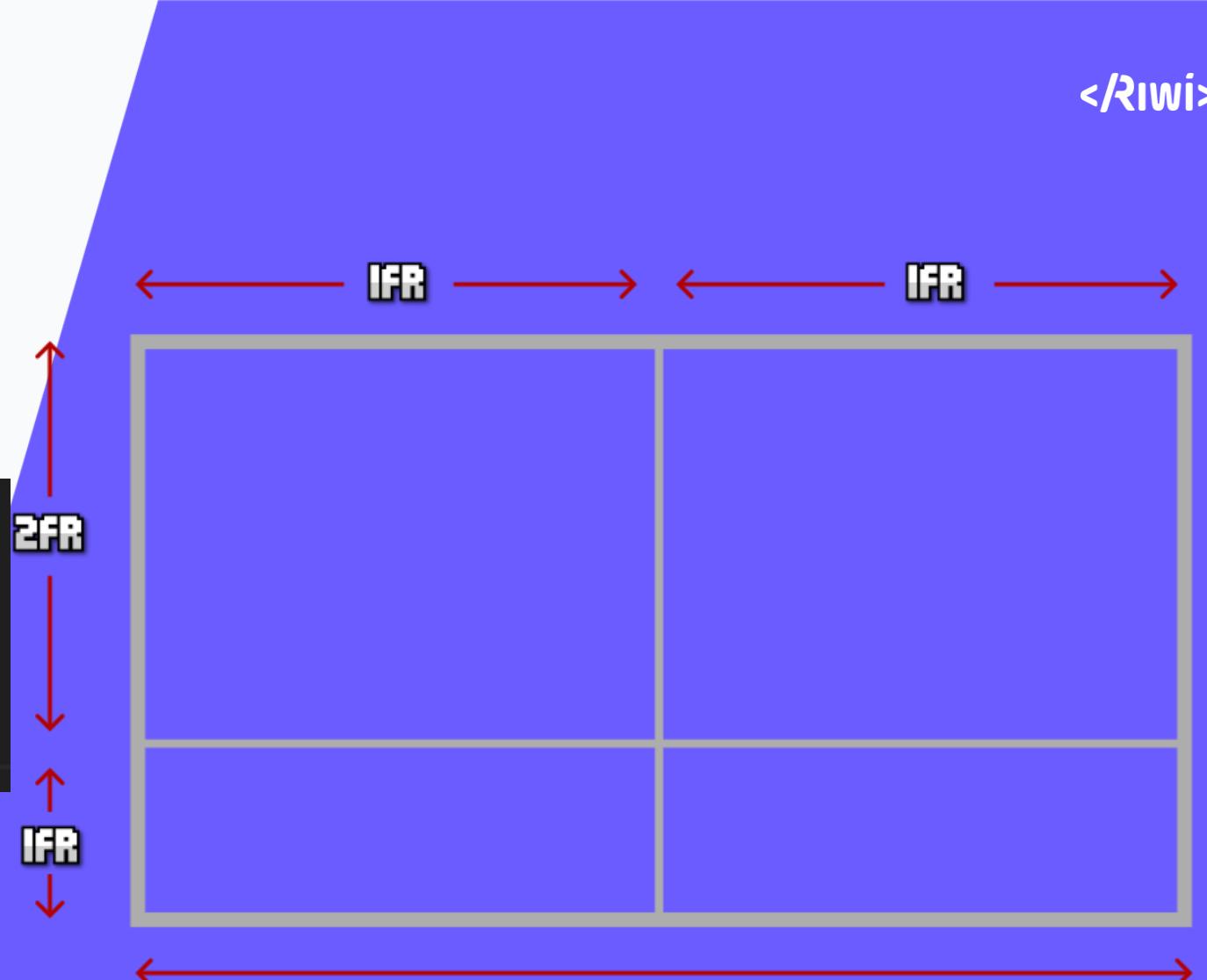
añadir ítems



Grid

Unidad fracción restante (fr)

```
Comment code  
.grid {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  grid-template-rows: 2fr 1fr;  
}
```



Profundidad (capas)

Z-index:15

Comment Code

```
.grid {  
  display: grid;  
  
  grid-template-columns: 100px repeat(4, 50px) 200px;  
  grid-template-rows: repeat(2, 1fr 2fr);  
  
  /* Equivalente a... */  
  grid-template-columns: 100px 50px 50px 50px 50px 200px;  
  grid-template-rows: 1fr 2fr 1fr 2fr;  
}
```

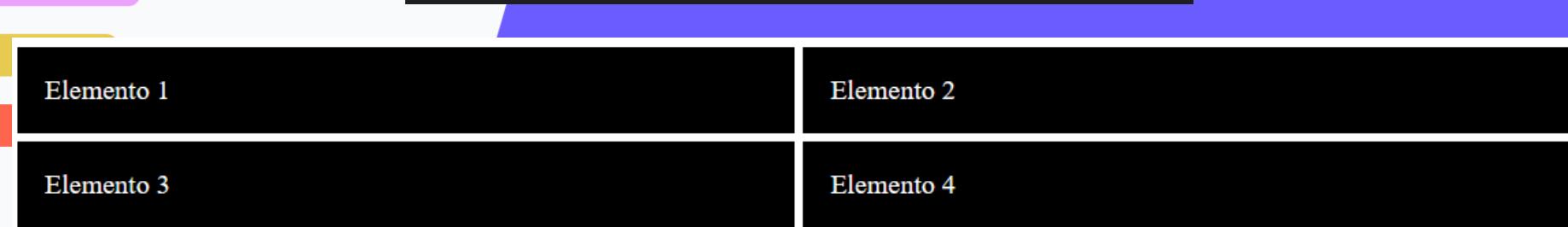


Grid

Funcion minmax()

Función	Descripción
minmax(min SIZE, max SIZE)	Define un rango entre min y max.

```
.container {  
  display: grid;  
  grid-template-columns: repeat(2, minmax(400px, 600px));  
  grid-template-rows: repeat(2, 1fr);  
  gap: 5px;  
}  
  
Comment Code  
.item {  
  background: □black;  
  color: ■white;  
  padding: 1em;  
}
```

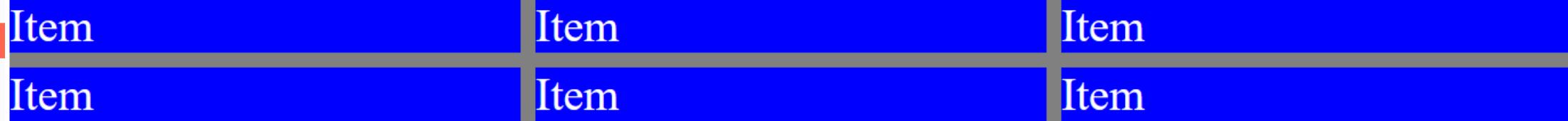
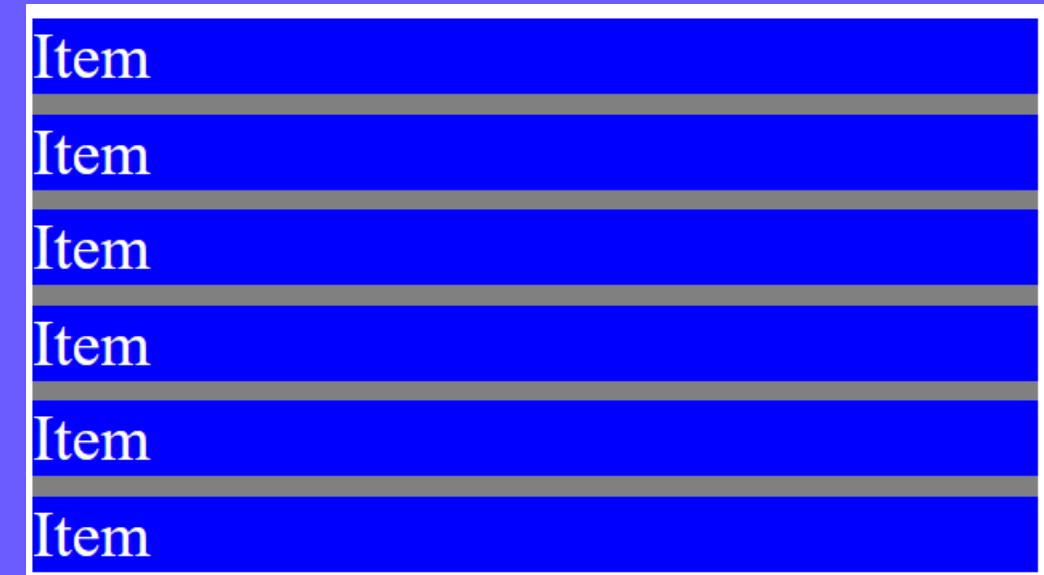
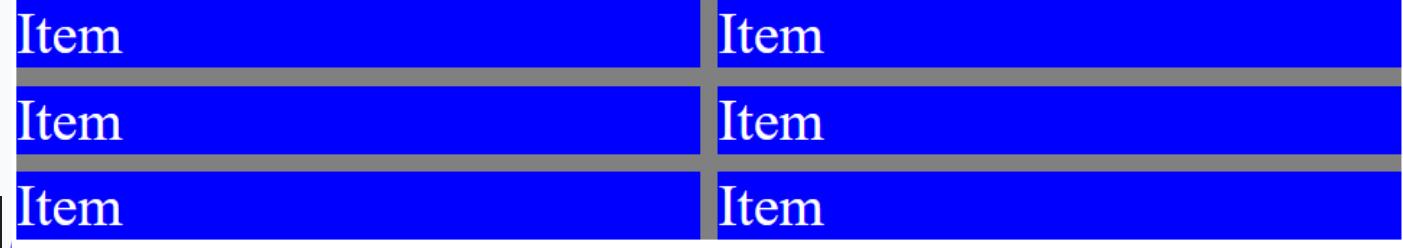


Grid

Los valores auto-fill y auto-fit

```
Comment Code
.grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  background: #grey;
  gap: 10px;
}
```

```
Comment Code
.item {
  background: #blue;
  color: #fff;
  font-size: 2rem;
}
```



Grid

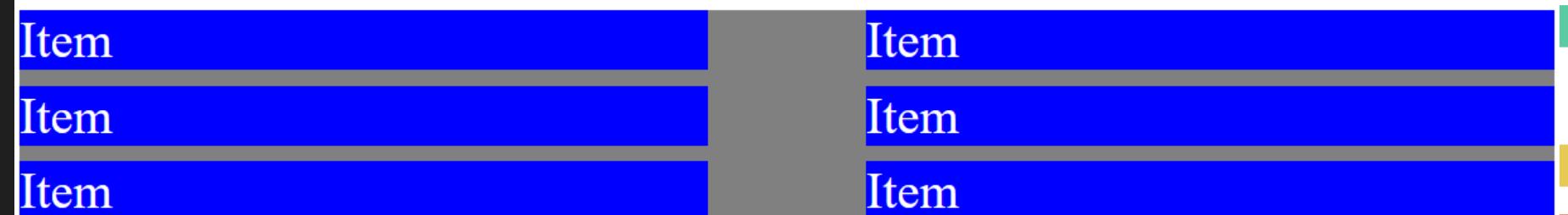
Huecos en grid

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
  background: □ grey;  
  gap: 10px;  
  column-gap: 100px;  
  row-gap: 10px;  
}
```

Comment Code

```
.item {  
  background: □ blue;  
  color: □ #fff;  
  font-size: 2rem;  
}
```

Propiedad	Descripción
column-gap	Establece el SIZE de los huecos entre columnas (líneas verticales).
row-gap	Establece el SIZE de los huecos entre filas (líneas horizontales).



Grid

Propiedades de alineación

Propiedad	Valores	Afecta a...
justify-items	start end center stretch	Elementos ①
align-items	start end center stretch	Elementos ②
justify-content	start end center stretch space-around space-between space-evenly	Contenido ①
align-content	start end center stretch space-around space-between space-evenly	Contenido ②

justify-items: Alinea los elementos (hijos) en horizontal (eje principal) dentro de cada celda.

align-items: Alinea los elementos (hijos) en vertical (eje principal) dentro de cada celda.

justify-content: Alinea el contenido (la cuadricula) en horizontal (eje secundario) en el contenedor padre

.

align-content: Alinea el contenido (la cuadricula) en vertical (eje secundario) en el contenedor padre.

Grid

Propiedad justify-items

Valor	Descripción
start	Coloca la cuadrícula en su conjunto al inicio del contenedor padre en su eje principal (horizontal).
end	Coloca la cuadrícula en su conjunto al final del contenedor padre en su eje principal (horizontal).
center	Coloca la cuadrícula en su conjunto al centro del contenedor padre en su eje principal (horizontal).
stretch	Estira la cuadrícula ocupando todo el espacio disponible del contenedor padre en su eje principal (horizontal).
space-between	Establece espacios sólo entre las celdas, en su eje principal (horizontal).
space-around	Establece espacios alrededor de las celdas, en su eje principal (horizontal).
space-evenly	Idem al anterior, pero solapando los espacios, de modo que sean todos de tamaño equivalente.

Grid

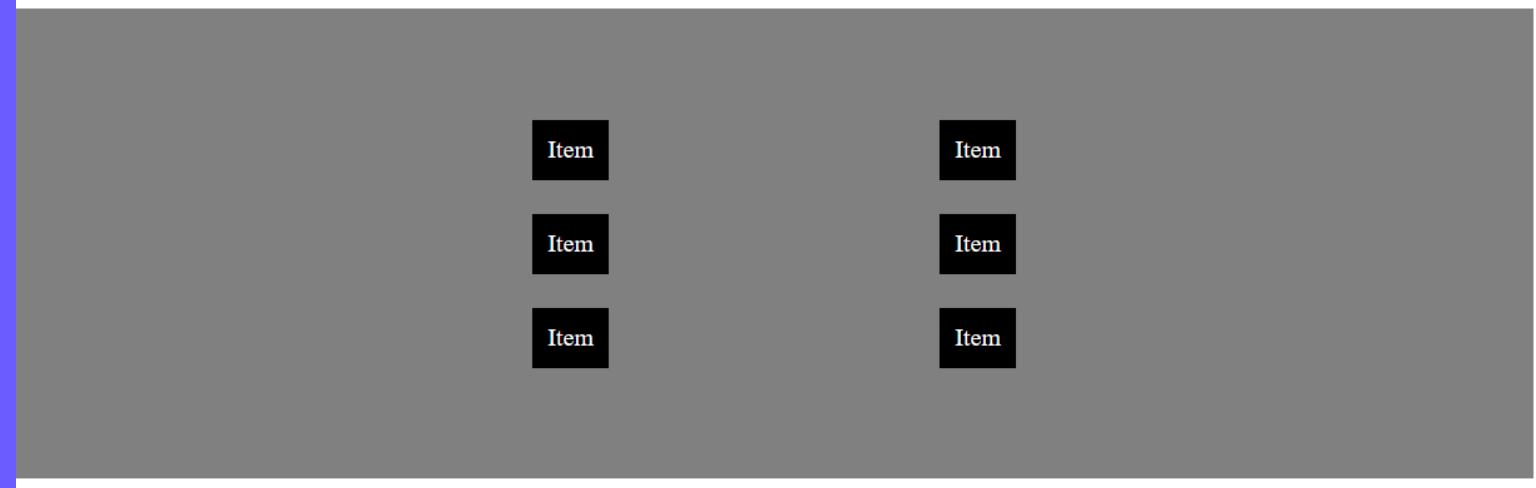
Propiedad align-content

```
.container {  
  display: grid;  
  grid-template-columns: repeat(2, 250px);  
  grid-template-rows: repeat(3, 50px);  
  gap: 10px;  
  background: #grey;  
  height: 300px;  
  
  justify-items: center;  
  align-items: center;  
  justify-content: center;  
  align-content: center;  
}
```

Comment Code

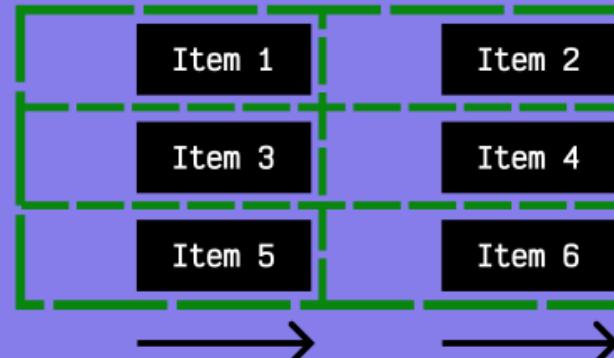
```
.item {  
  padding: 10px;  
  background: black;  
  color: white;  
}
```

```
<div class="container">  
  <div class="item">Item</div>  
  <div class="item">Item</div>  
  <div class="item">Item</div>  
  <div class="item">Item</div>  
  <div class="item">Item</div>  
  <div class="item">Item</div>  
</div>
```



Grid

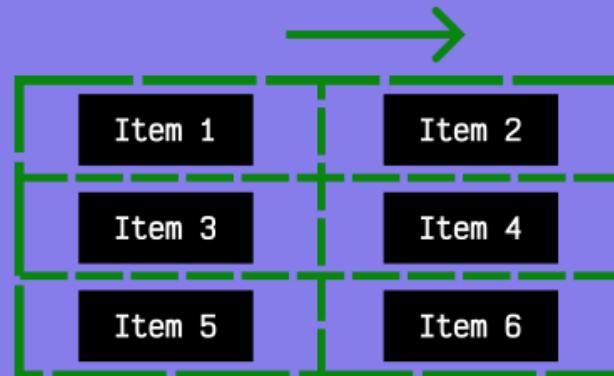
Justify-content



JUSTIFY-ITEMS: end



ALIGN-ITEMS: end



JUSTIFY-CONTENT: end



ALIGN-CONTENT: end

Grid

Por áreas

Propiedad	Descripción
grid-template-areas	Indica la disposición de las áreas en el grid. Cada texto entre comillas simboliza una fila.
grid-area	Indica el nombre del área. Se usa sobre ítems hijos del grid.

```
.container {  
  display: grid;  
  grid-template-areas: "head head"  
                      | "menu main"  
                      | "foot foot";  
  
  grid-template-columns: 1fr 1fr;  
  grid-template-rows: 100px 200px 100px;  
}
```

Comment Code

```
.item-1 { grid-area: head; background: blue; }
```

Comment Code

```
.item-2 { grid-area: menu; background: red; }
```

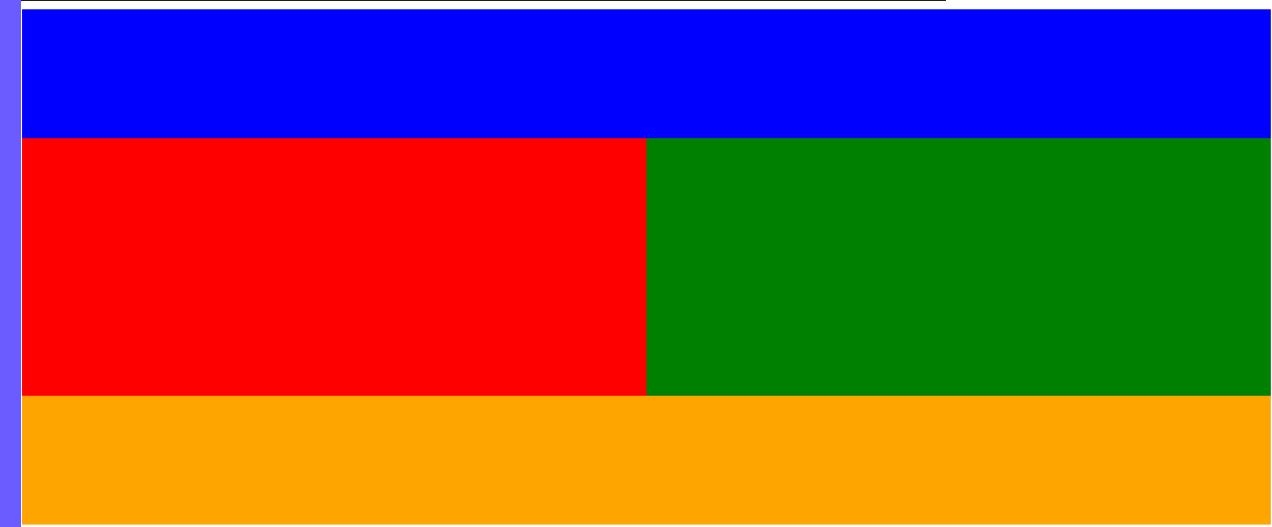
Comment Code

```
.item-3 { grid-area: main; background: green; }
```

Comment Code

```
.item-4 { grid-area: foot; background: orange; }
```

```
<div class="container">  
  <div class="item item-1"></div>  
  <div class="item item-2"></div>  
  <div class="item item-3"></div>  
  <div class="item item-4"></div>  
</div>
```

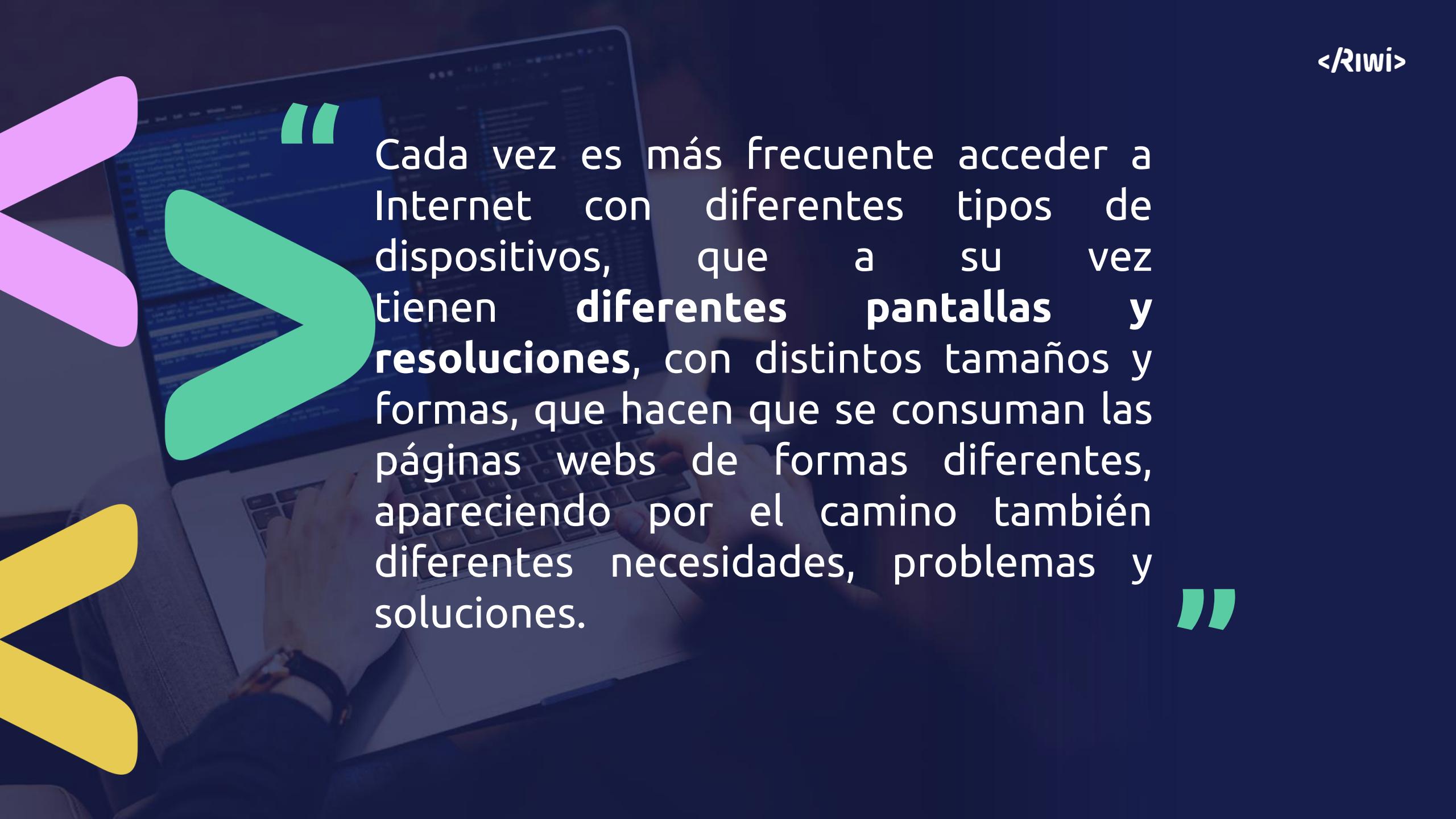


Grid

Propiedades grid-templates-areas

Propiedad	Valores	Descripción
grid-template-areas	none STRING fila1, STRING fila2, ...	Define cada fila del grid, indicando el nombre del área a colocar.

Valores	Descripción
none	Indica que no se creará ninguna plantilla de áreas.
"head"	Indica que se creará una fila de una columna con el área head.
"head menu"	Indica que se creará una fila de 2 columnas con el área head en una y el área menu en otra.
"head head"	Indica que se creará una fila de 2 columnas con el área head ocupando ambas.
". "	Indica que se colocará una celda sin nombre (nula) en esta posición.



“ Cada vez es más frecuente acceder a Internet con diferentes tipos de dispositivos, que a su vez tienen **diferentes pantallas y resoluciones**, con distintos tamaños y formas, que hacen que se consuman las páginas webs de formas diferentes, apareciendo por el camino también diferentes necesidades, problemas y soluciones.

”

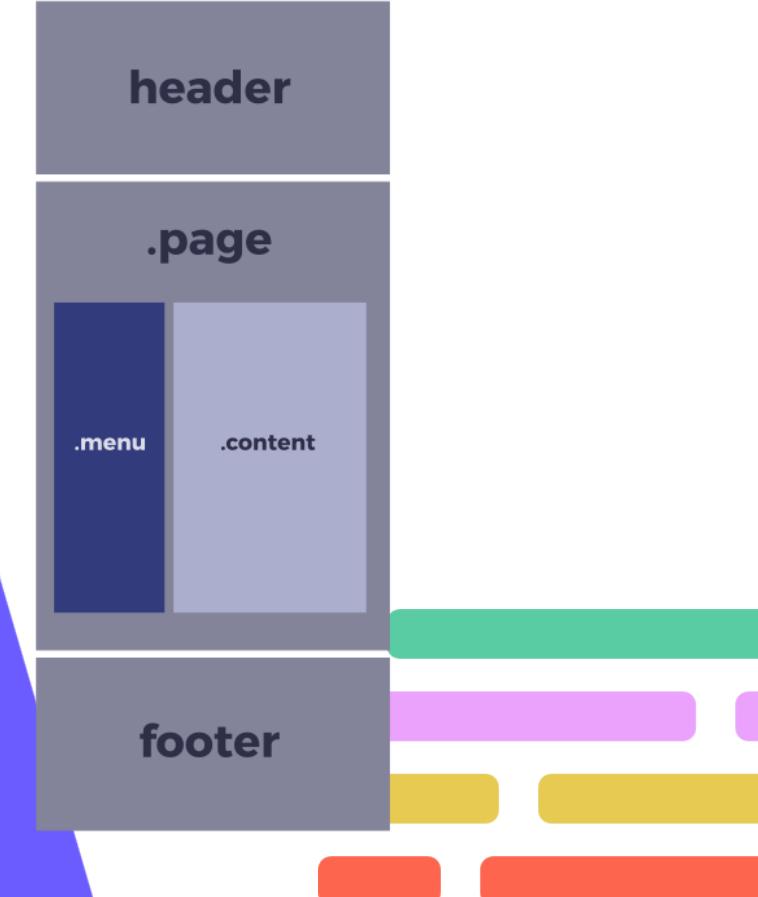
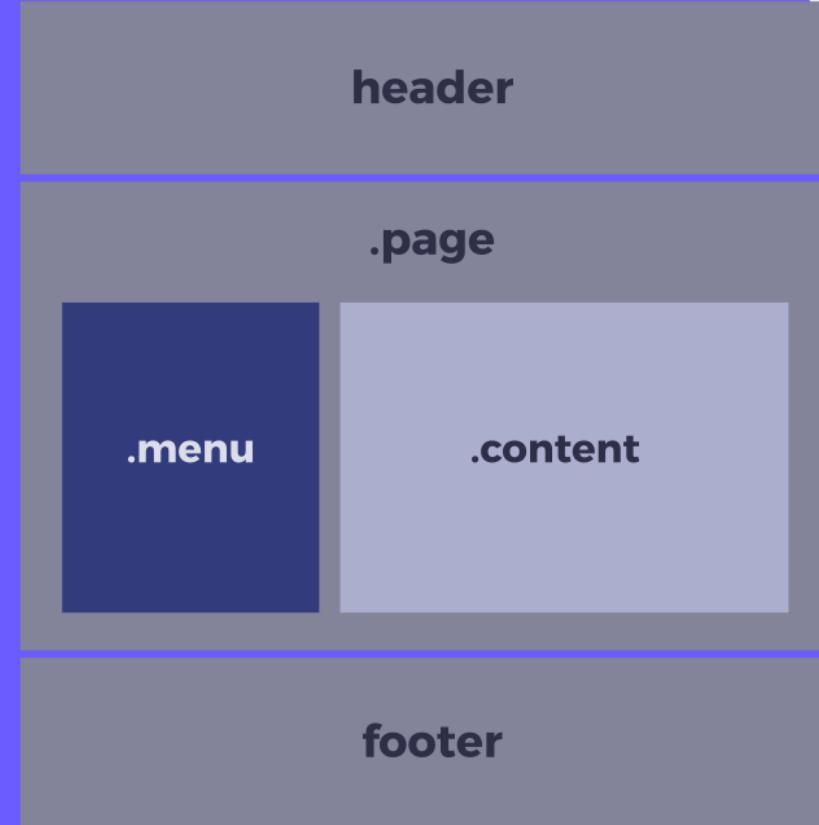
Responsive

Diseño con porcentajes (Adaptativo)

Un ancho de 100% a header, .page y footer

Un ancho de 30% a .menu

Un ancho de 70% a .content



Responsive

Solución recomendada

Como comentamos antes, una forma simple y recomendada de solucionar el problema del ejemplo anterior, es aprender a utilizar Flex o Grid.

De esta forma, establecemos que el padre .page sea un elemento flexible, por lo tanto, conseguimos que nuestro diseño se adapte de forma adecuada a la página, sin necesidad de tener que ajustar los márgenes, rellenos, bordes o tamaño de los contenidos.

```
.page{  
    display: flex;  
}
```



Responsive

Tamaños mínimo y máximos

```
Comment Code  
img,  
Comment Code  
video,  
Comment Code  
object,  
Comment Code  
embed {  
    max-width: 100%;  
    height: auto;  
}
```

```
.picture {  
    min-height: 200px;      /* Por defecto, height es 0 */  
    background: □ grey;    /* Simplemente, para verlo visualmente */  
  
    max-width: 1024px;  
    min-width: 800px;  
}
```

Responsive

El viewport del navegador

Parámetro	Valor	Significado
width	device-width	Indica un ancho para el viewport.
height	device-height	Indica un alto para el viewport.
initial-scale	1	Escala inicial con la que se visualiza la página web.
minimum-scale	0.1	Escala mínima a la que se puede reducir al hacer zoom.
maximum-scale	10	Escala máxima a la que se puede aumentar al hacer zoom.
user-scalable	no o fixed yes o zoom	Posibilidad de hacer zoom en la página web.

</Be a
coder>