

# ¡CSS GRID LAYOUT YA ESTÁ AQUÍ!

MANUEL REGO CASASNOVAS ([@regocas](#))

HTML5Spain / 28 Junio 2016



# PRESENTACIÓN

Actualmente implementando **CSS Grid Layout**  
(Chromium/Blink & Safari/WebKit)

Miembro de Igalia



**igalia**

# GRIDS FOR TODAS PARTES

SECTIONS SEARCH U.S. INTERNATIONAL 中文 SUBSCRIBE NOW LOG IN

**The New York Times**  
Tuesday, May 19, 2015 | Today's Paper | Video | 64°F | Nasdaq -0.17% ↓

World U.S. Politics New York Business Opinion Technology Science Health Sports Arts Style Food Travel Magazine Real Estate ALL

**The New York Times** Countries have borders. Stories don't. TRY A DIGITAL SUBSCRIPTION 4 WEEKS JUST €1 GET STARTED

### ISIS Used Sandstorm to Gain Edge in Iraq Attack

By ERIC SCHMITT and HELENE COOPER

The sandstorm during the Islamic State's assault last week on the Iraqi provincial capital of Ramadi delayed American warplanes and kept them from launching airstrikes to help Iraqi forces, as the terrorist group evidently anticipated.

192 Comments

GRAPHIC  
**Islamic State's Finances Are Strong**  
By SARAH ALMUKHTAR  
ISIS has more than enough in its coffers despite expectations that airstrikes and falling oil prices would



Standing over Sonny Liston in the first round of their title fight, Muhammad Ali was heard to say, "Get up and fight, sucker!"  
John Rooney/Associated Press

### 50 Years On, City Tries to 'Get Up and Fight'

By HARVEY ARATON

May 25 is the 50th anniversary of one of the craziest title fights ever, between Muhammad Ali and Sonny Liston in Lewiston, Me. Liston was down on his luck. Now Lewiston is, too.

Comments

### The Opinion Pages

OP-ED CONTRIBUTOR  
**Free Trade Is Not the Enemy**  
By WILLIAM M. DALEY  
What I learned from my experience guiding NAFTA through Congress.

- Editorial: In Egypt, Deplorable Death Sentences
- Op-Ed: How to Prevent an Oil Train Disaster
- Opinion on Facebook »

Room for Debate: Arctic Priorities  
What's the best way to balance drilling and environmental concerns in the North Pole?

OP-ED COLUMNISTS

- Brooks: Learning From Mistakes
- Nocera: Chemo for the Planet
- Op-Ed: Give Up on Netanyahu, Go to the United Nations



VERLAPPING 3 COLUMN GRID

# 1996

NEW IN BUSINESS

**BANKING  
CENTER**

# The New York Times

ON THE WEB

knicks<sup>NEW!</sup>  
celtics  
one on one  
newyork.boston.com

"All the News That's Fit to Print"

Thursday, November 21, 1996

## SECTIONS

- Front Page ● CyberTimes
- Politics ● Business
- Editorials/Letters ● Op-Ed
- Arts & Leisure ● Travel
- Real Estate ● Job Market
- Diversions ● Web Specials

## NEWS BY CATEGORY

CLASSIFIEDS | FORUMS

SERVICES | SEARCH

TABLE OF CONTENTS



Norman Gives Clinton Some Tips

## LATE NEWS UPDATE

700,000 Additional  
Refugees Reported  
Seen Inside Zaire

## IN CYBERTIMES

Wired Truckers:  
Big Rigs Roll  
Into Cyberspace

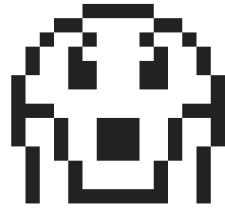
# HÅKON WIUM LIE (JUNIO, 1995)

Grids! Let the style sheet carve up the canvas into golden rectangles, and use an expert system to lay out the elements!! Ok, drop the expert system and define a set of simple rules that we hardcode.. whoops! But grids do look nice!

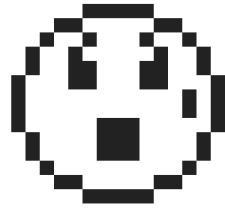
-h&kon

# EVOLUCIÓN

**<TABLE>**

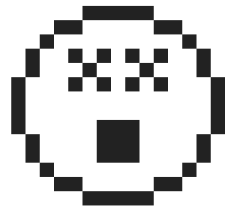


**FLOAT**

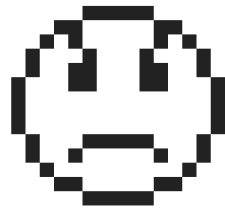




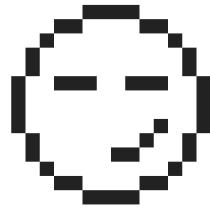
**DISPLAY: INLINE-BLOCK;**



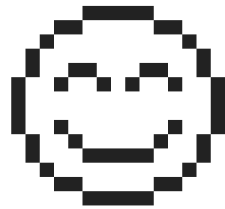
```
DISPLAY: TABLE;
```



# CSS FRAMEWORKS



**DISPLAY: FLEX;**



**DISPLAY: GRID;**





# INTRODUCCIÓN

# CONCEPTOS

Título

Menú

Contenido

Pie

# LÍNEAS

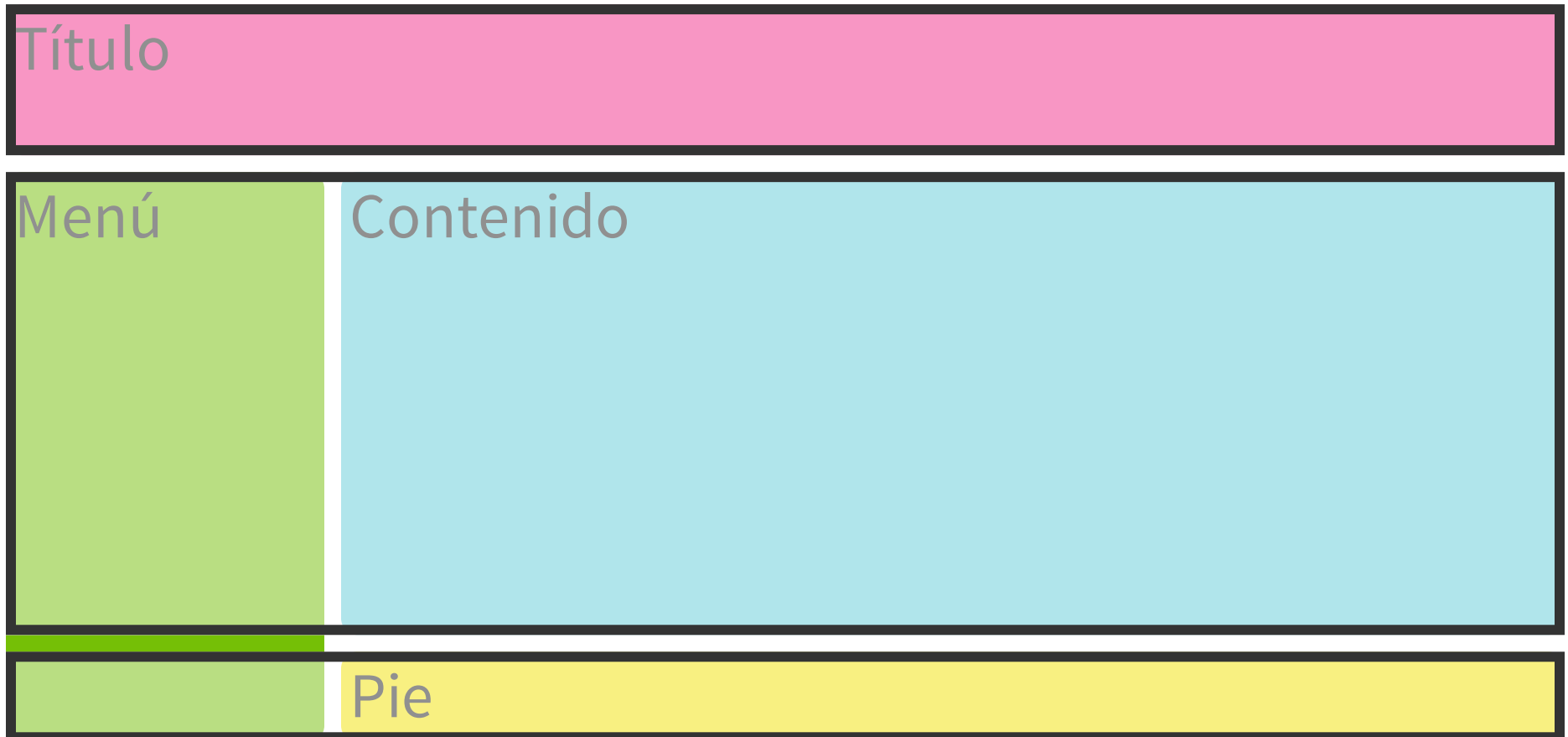




**TRACKS**

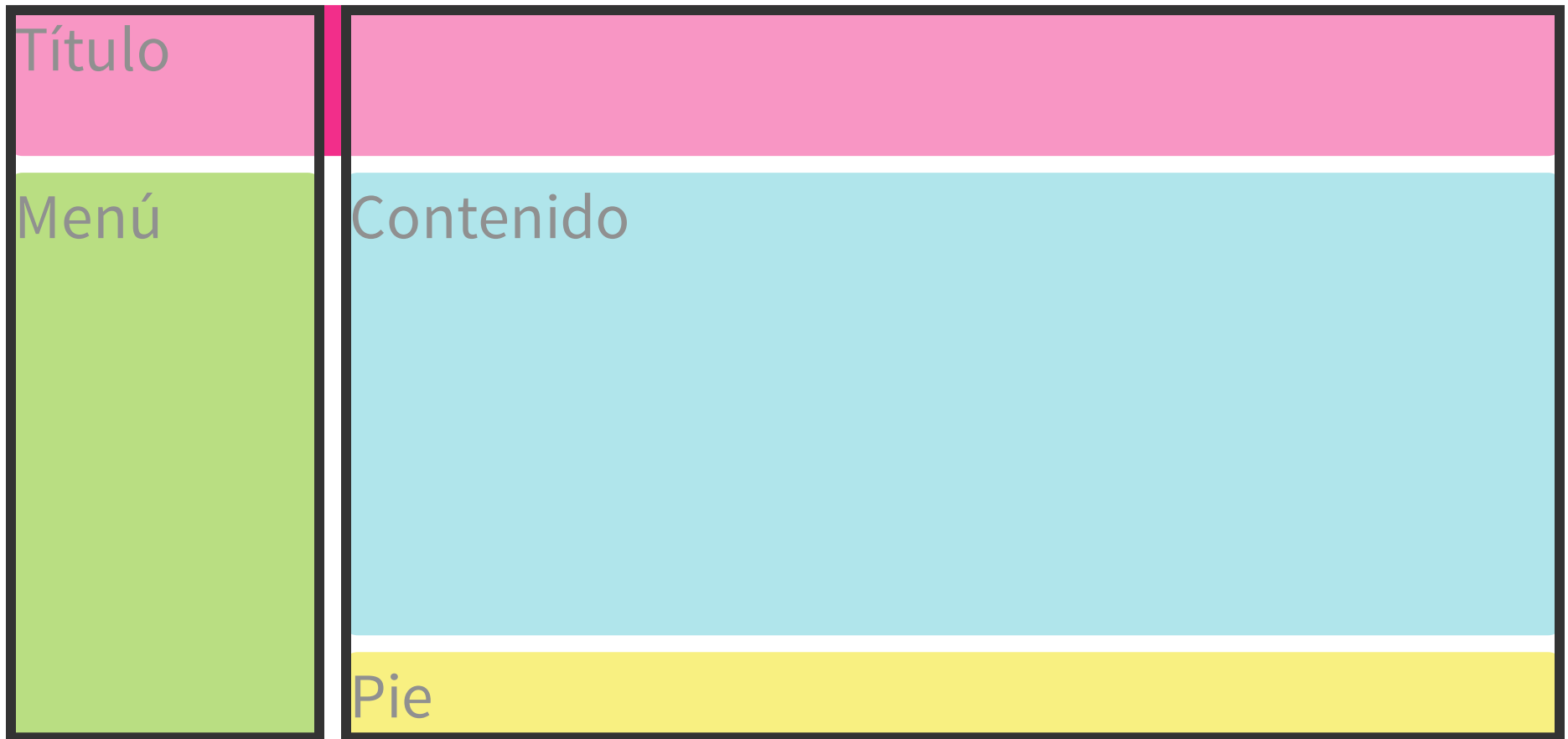
# TRACKS

## FILAS

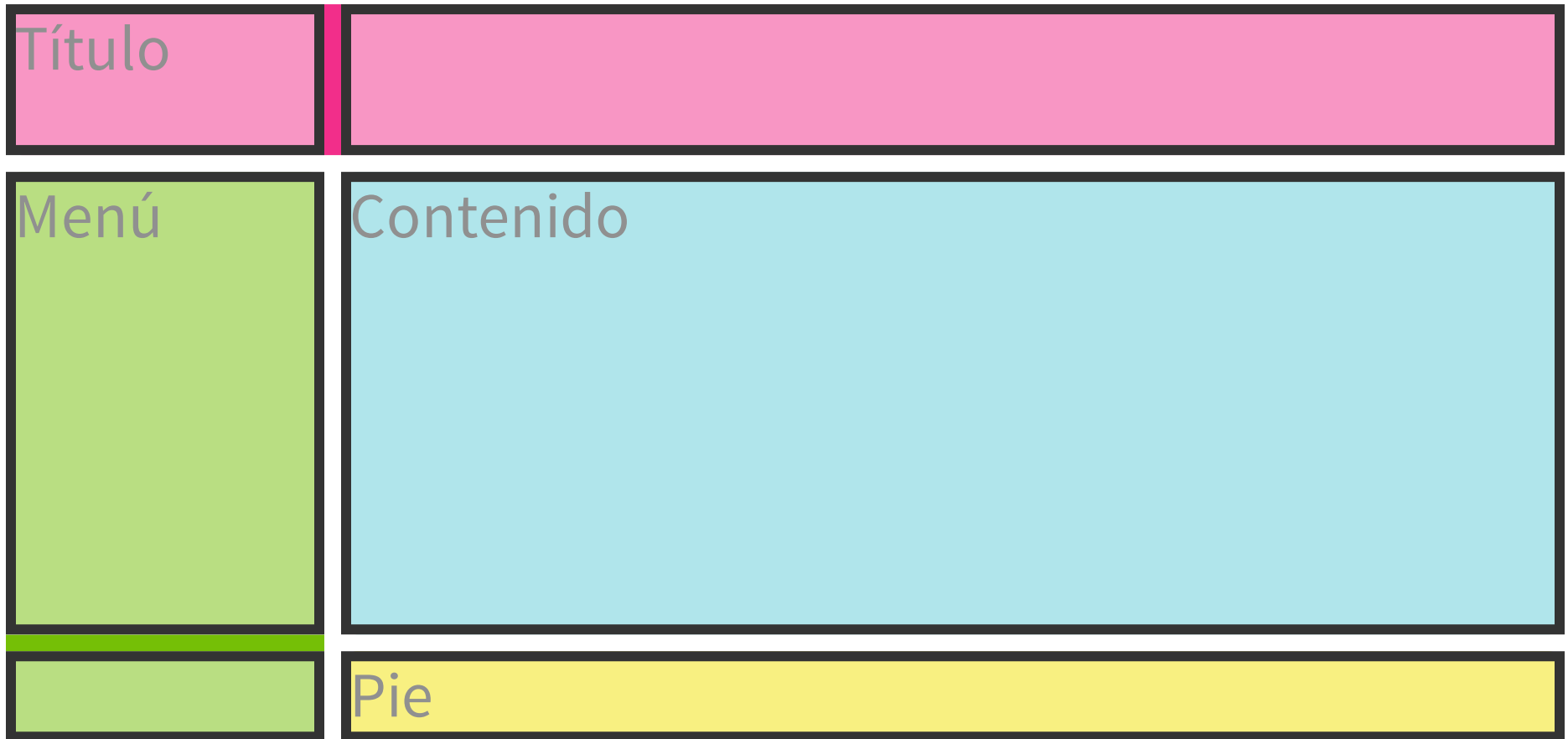


# TRACKS

## COLUMNAS



# CELIDAS



# ÁREAS





**SYNTAX**

14

**DISPLAY: GRID;**

Rompe el flujo habitual de la página web

# TAMAÑO DE LOS TRACKS (FILAS/COLUMNAS)

`grid-template-columns` & `grid-template-rows`

¡Creación de cajas desde CSS!



# TAMAÑO DE LOS TRACKS

```
<div class="grid">  
  <div class="a">A</div>  
  <div class="b">B</div>  
  <div class="c">C</div>  
  <div class="d">D</div>  
</div>
```



```
.grid { display: grid;  
  grid-template-columns: ;
```

# COLOCACIÓN DE ÍTEMS

`grid-column` & `grid-row`

Orden del DOM  $\neq$  Orden visual



# COLOCACIÓN DE ÍTEMS

```
<div class="grid">  
  <div class="a">A</div>  
  <div class="b">B</div>  
  <div class="c">C</div>  
</div>
```

A

B

C

```
.grid { display: grid;  
        grid: 100px 100px / 200px 200px; }
```

# LÍNEAS DEL GRID CON NOMBRE

Utilizar identificadores personalizados para las líneas

Una línea puede tener varios nombres

# LÍNEAS DEL GRID CON NOMBRE

```
<div class="grid">  
  <div class="a">A</div>  
  <div class="b">B</div>  
  <div class="c">C</div>  
</div>
```

A

B

C

```
.grid { display: grid;  
  grid-template-columns:
```

# ÁREAS DEL GRID

grid-template-areas

ASCII ART

# ÁREAS DEL GRID

```
<div class="grid">
  <div class="a">A</div>
  <div class="b">B</div>
  <div class="c">C</div>
  <div class="d">D</div>
</div>
```

A

B

C

D

```
.grid { display: grid;
        grid-auto-columns: 100px; grid-auto-rows: 75px;
        grid-template-areas: "header header"
                             "nav      main   "
                             "footer footer"; }

.a { grid-area: header; }
.b { grid-area: main; }
.c { grid-area: nav; }
.d { grid-area: footer; }
```

# ALINEAR Y JUSTIFICAR

Especificación [CSS Box Alignment](#)

¡Centrado horizontal & vertical!



# ALINEAR Y JUSTIFICAR ÍTEMS

```
<div class="grid">  
  <div class="a">A</div>  
  <div class="b">B</div>  
  <div class="c">C</div>  
</div>
```



```
.grid { display: grid; grid: 100px 100px / 200px 200px;  
  align-items: ; justify-items: ; }
```

# ALINEAR Y JUSTIFICAR TRACKS

```
<div class="grid">  
  <div class="a">A</div>  
  <div class="b">B</div>  
  <div class="c">C</div>  
</div>
```



```
.grid { display: grid; grid: 100px 100px / 150px 150px;  
  align-content: ;
```

# SEPARACIÓN ENTRE TRACKS

`grid-row-gap` & `grid-column-gap`

# SEPARACIÓN ENTRE TRACKS

```
<div class="grid">
  <div class="a">A</div>
  <div class="b">B</div>
  <div class="c">C</div>
  <div class="d">D</div>
  <div class="e">E</div>
</div>
```



```
.grid { display: grid;
        grid: 100px 100px / 100px 100px 100px;
```

# COLOCACIÓN AUTOMÁTICA

`grid-auto-flow`

# COLOCACIÓN AUTOMÁTICA

```
<form>
  <label>Etiqueta</label>
  <input>
  <label>Checkbox</label>
  <input type="checkbox">
  <button>Enviar formulario</butt
</form>
```

Etiqueta

Checkbox

Enviar formulario

```
form { }
label { }
```

# REPEAT ( ) AUTOMÁTICO

Valores `auto-fill` & `auto-fit` para `repeat ( )`

# REPEAT ( ) AUTOMÁTICO

```
<div class="grid">  
  <div class="a">A</div>  
  <div class="b">B</div>  
  <div class="c">C</div>  
  <div class="d">D</div>  
  <div class="e">E</div>  
</div>
```



```
.grid { display: grid;  
        grid-template-columns: repeat(3, 100px);
```



# GRIDS ADAPTATIVOS (RESPONSIVE)

El tamaño de los tracks puede ser flexible

Media Queries

# GRIDS ADAPTATIVOS (RESPONSIVE)

```
.grid {
  display: grid;
  grid: 200px 1fr / 100px 1fr auto;
  grid-template-areas: "header header"
                       "menu  main  "
                       "menu  footer";
}

@media (max-width: 400px) {
  .grid {
    grid: 1fr / 100px 1fr 100px auto;
    grid-template-areas: "header"
                         "main  "
                         "menu  "
                         "footer"; }
}
```

---

# GRIDS ADAPTATIVOS (RESPONSIVE)

Título

Menú

Contenido

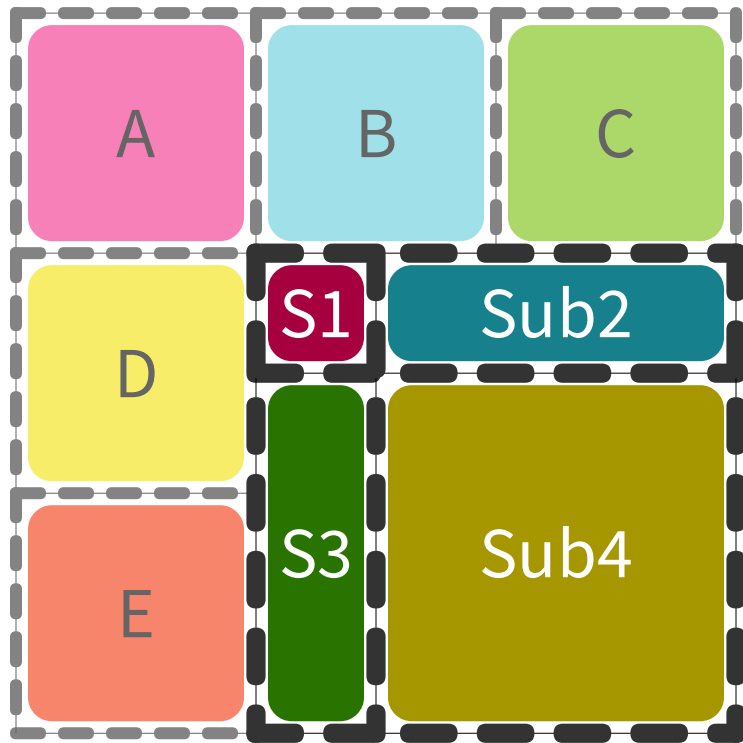
Pie



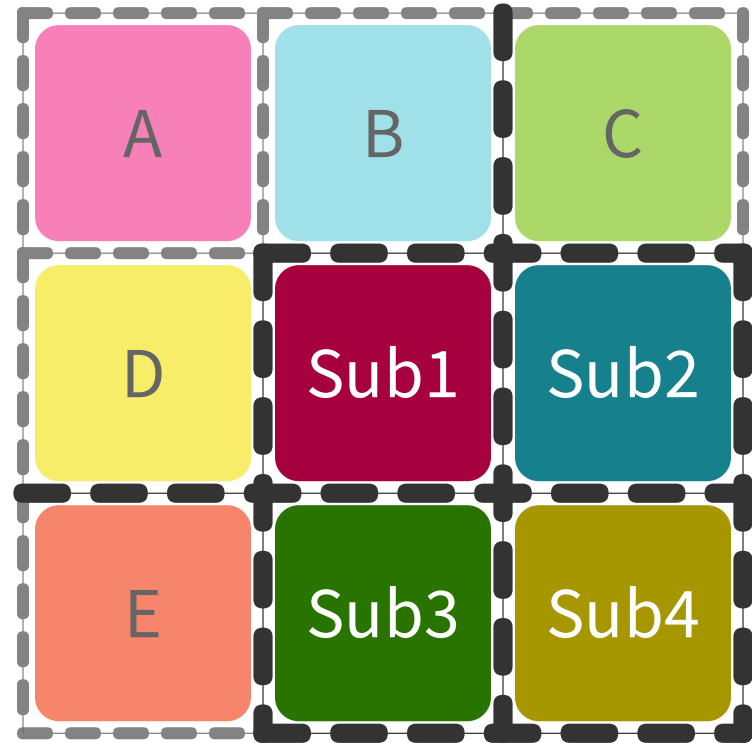
**FUTURO**

# SUBGRIDS

## Grid Anidado



## Subgrid



# SUBGRIDS

```
<form><ul>
  <li><label>Etiqueta</label><input></li>
  <li><label>Checkbox</label><input type="checkbox">
  <li><button>Enviar formulario</button></li>
</ul></form>
```

Etiqueta

Checkbox

Enviar formu

```
ul { display: grid; }
li { display: grid; }
```

**¿CÓMO FUNCIONAN?**

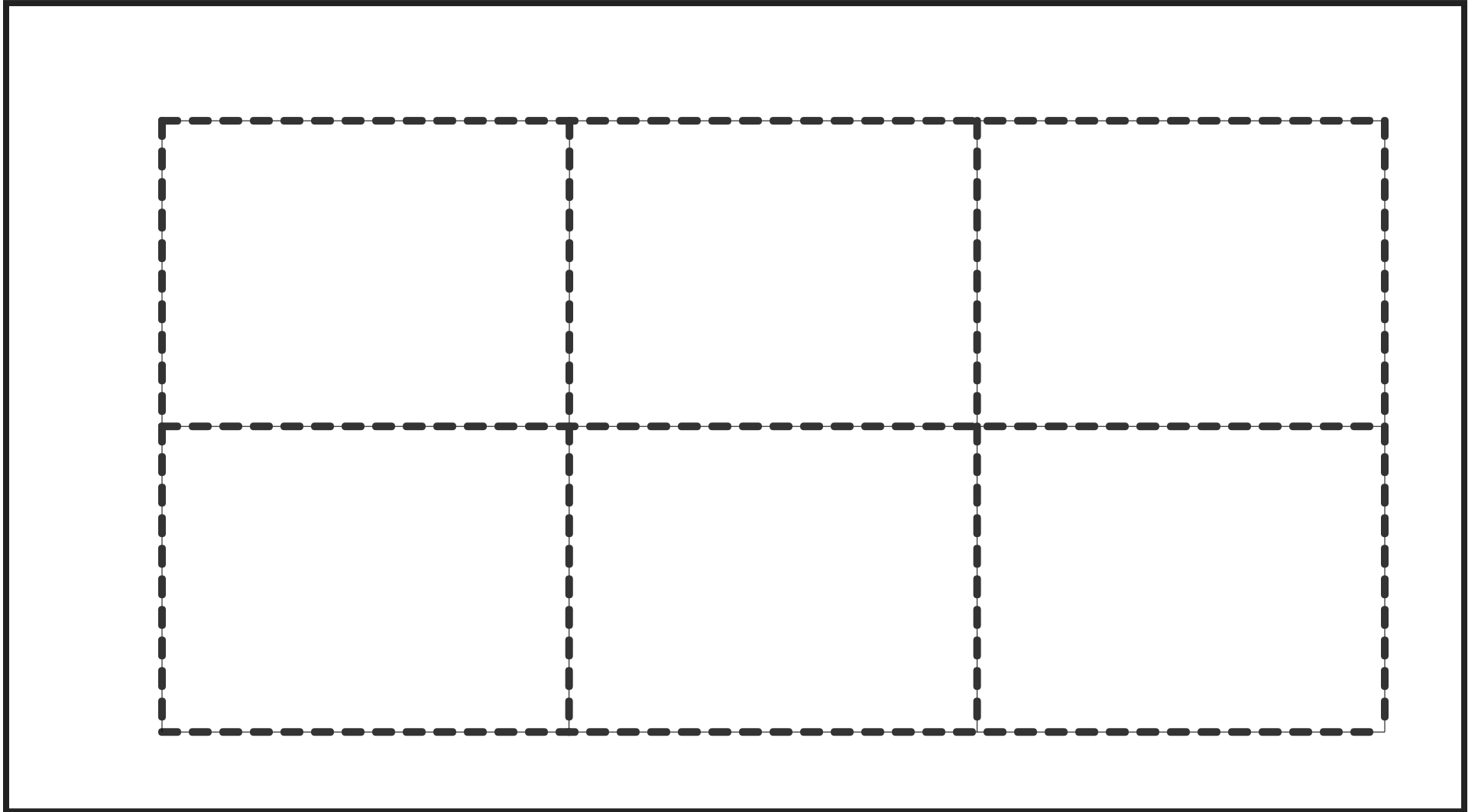
# EJEMPLO

```
<div class="grid">
  <div class="title">Title</div>
  <div class="nav">Nav</div>
  <div class="main">Lorem ipsum...</div>
  <div class="aside">Ad</div>
  <div class="aside">Adword</div>
</div>
```

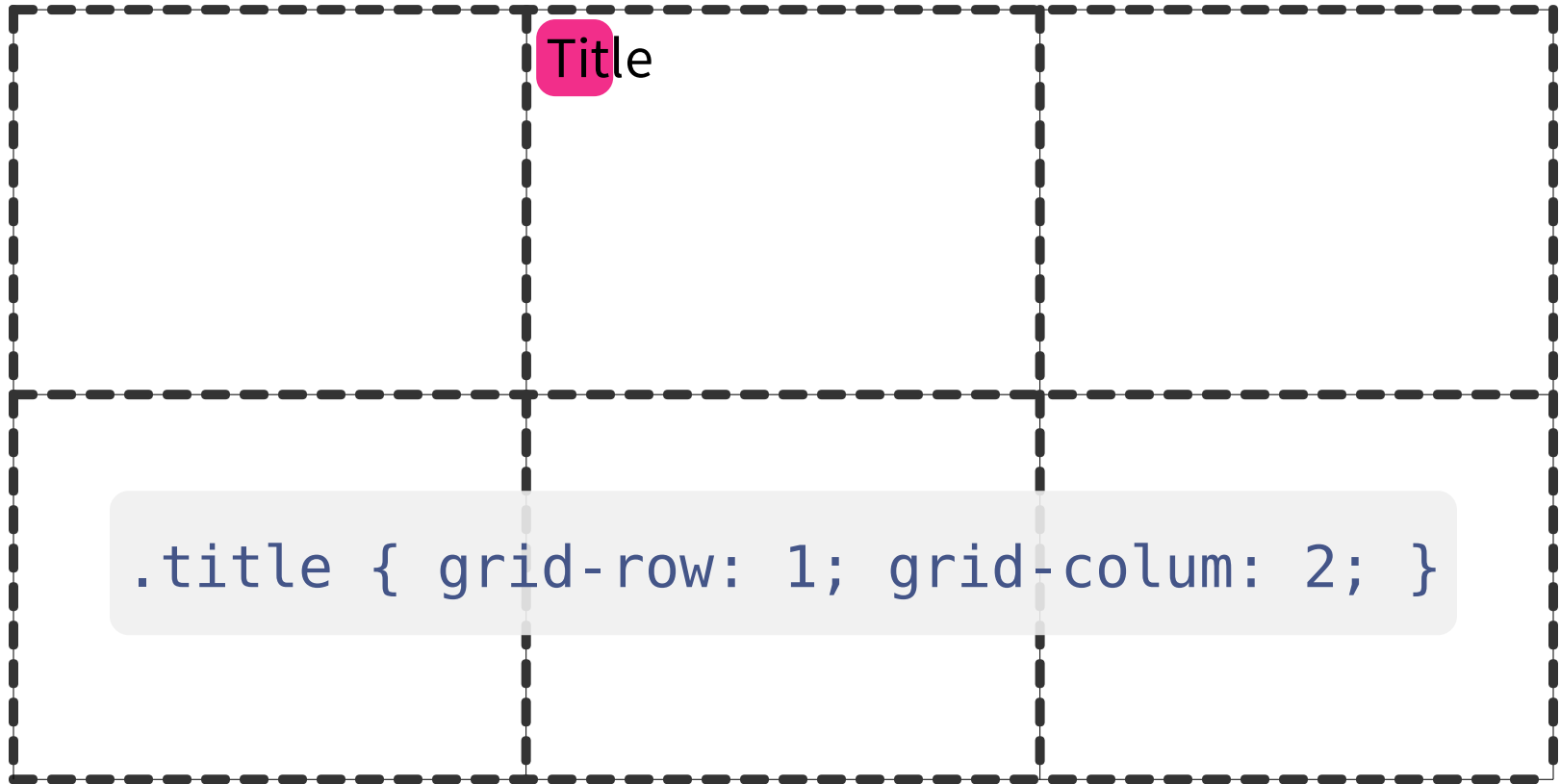
```
.grid { display: grid;
        width: 800px;
        grid-template-columns: 200px 1fr auto;
        grid-template-rows: 100px auto; }
.title { grid-row: 1; grid-column: 2; }
.nav    { grid-row: 2; grid-column: 1; }
.main   { grid-row: 2; grid-column: 2; }
.aside  {                grid-column: 3; }
```



# GRID VACÍO

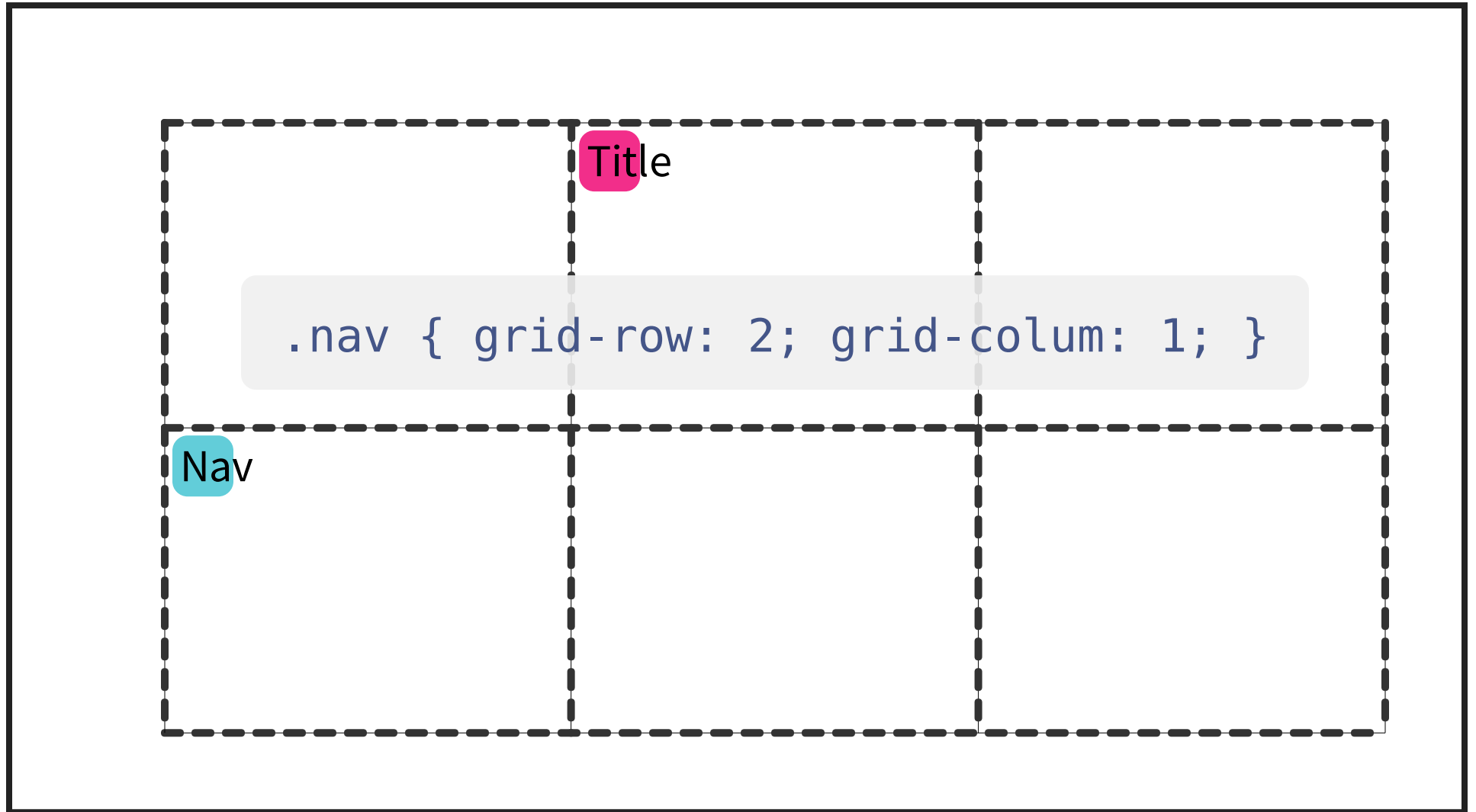


# COLOCACIÓN DE ÍTEMS

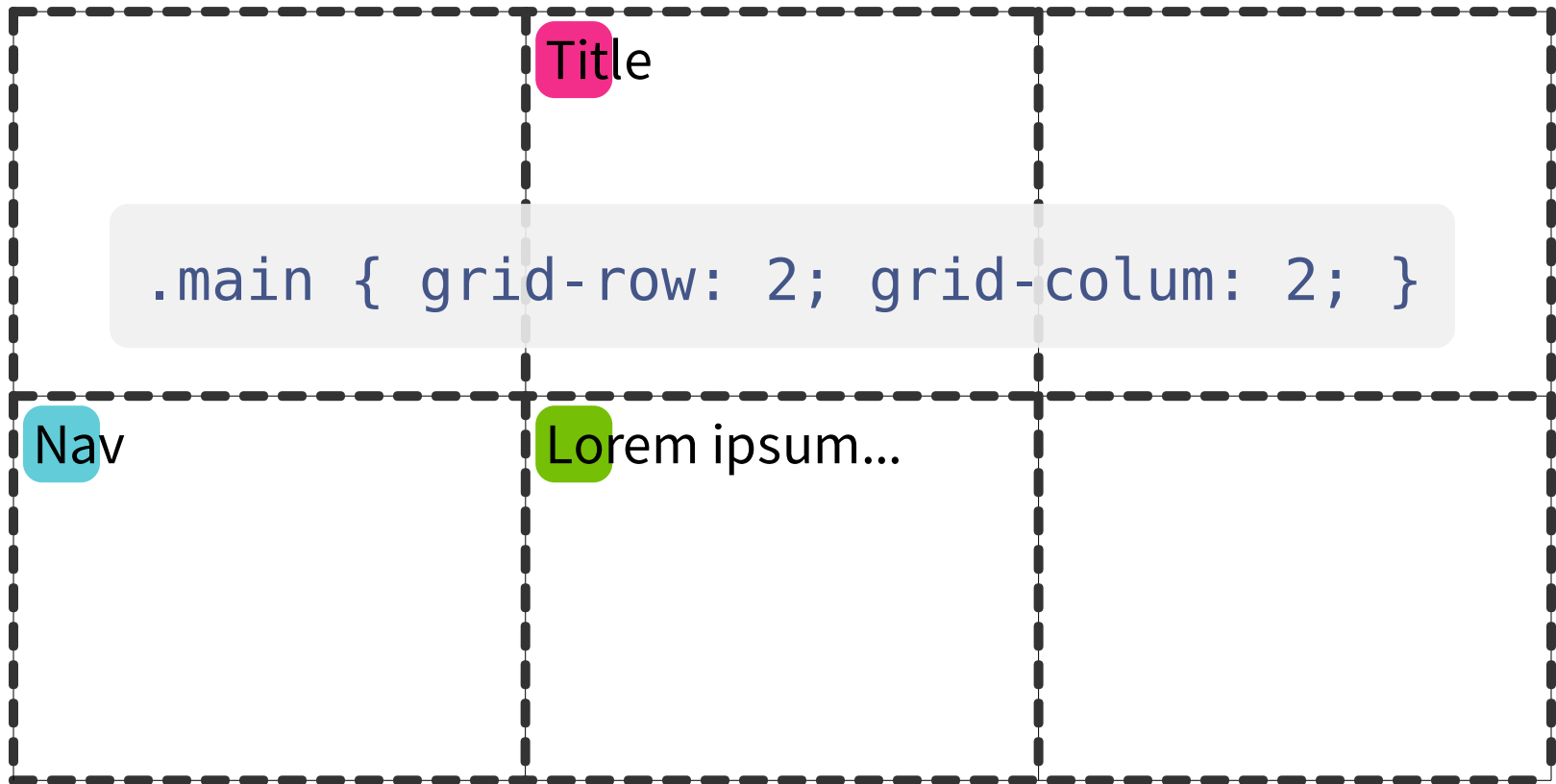


```
.title { grid-row: 1; grid-column: 2; }
```

# COLOCACIÓN DE ÍTEMS

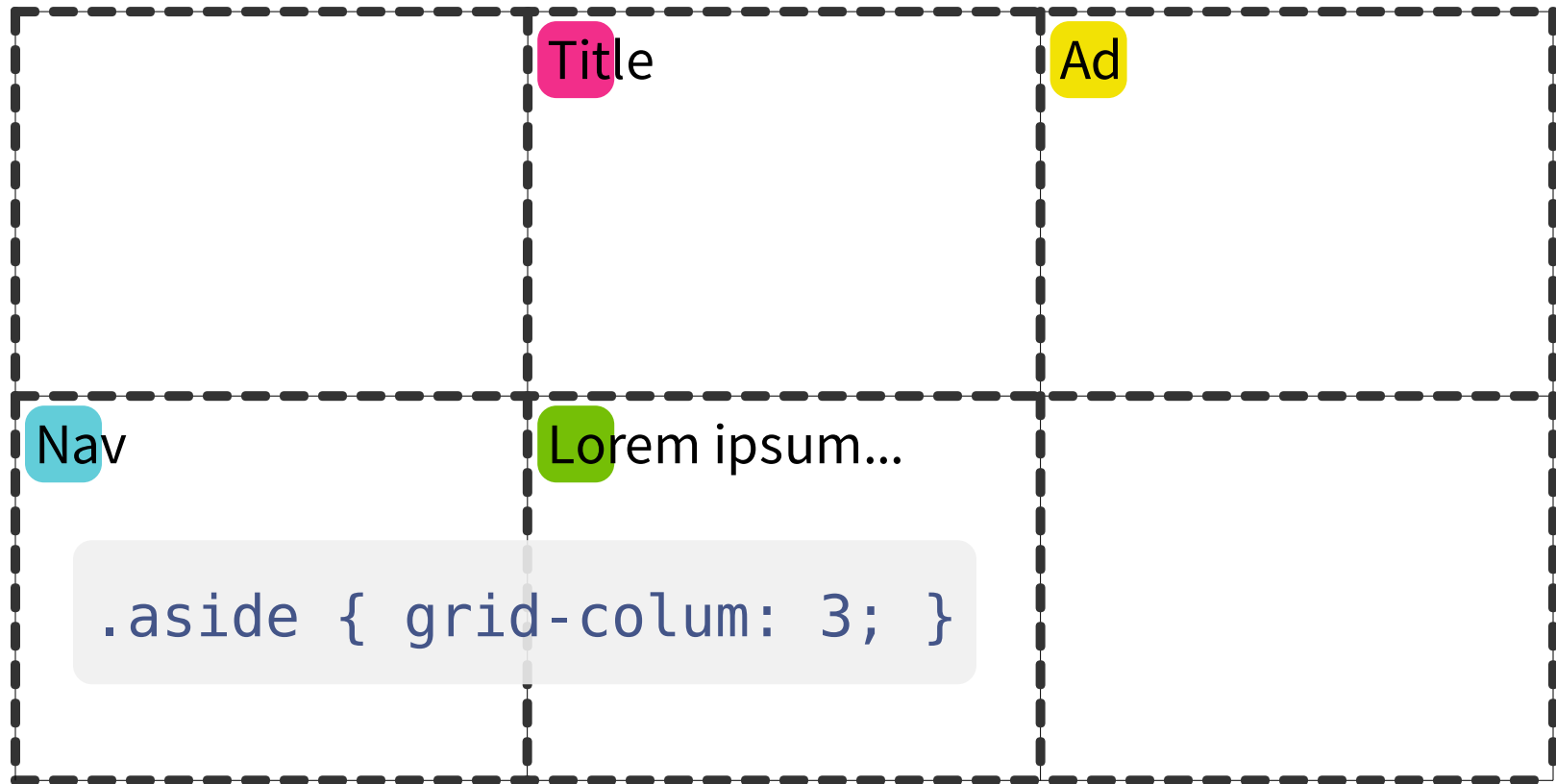


# COLOCACIÓN DE ÍTEMS

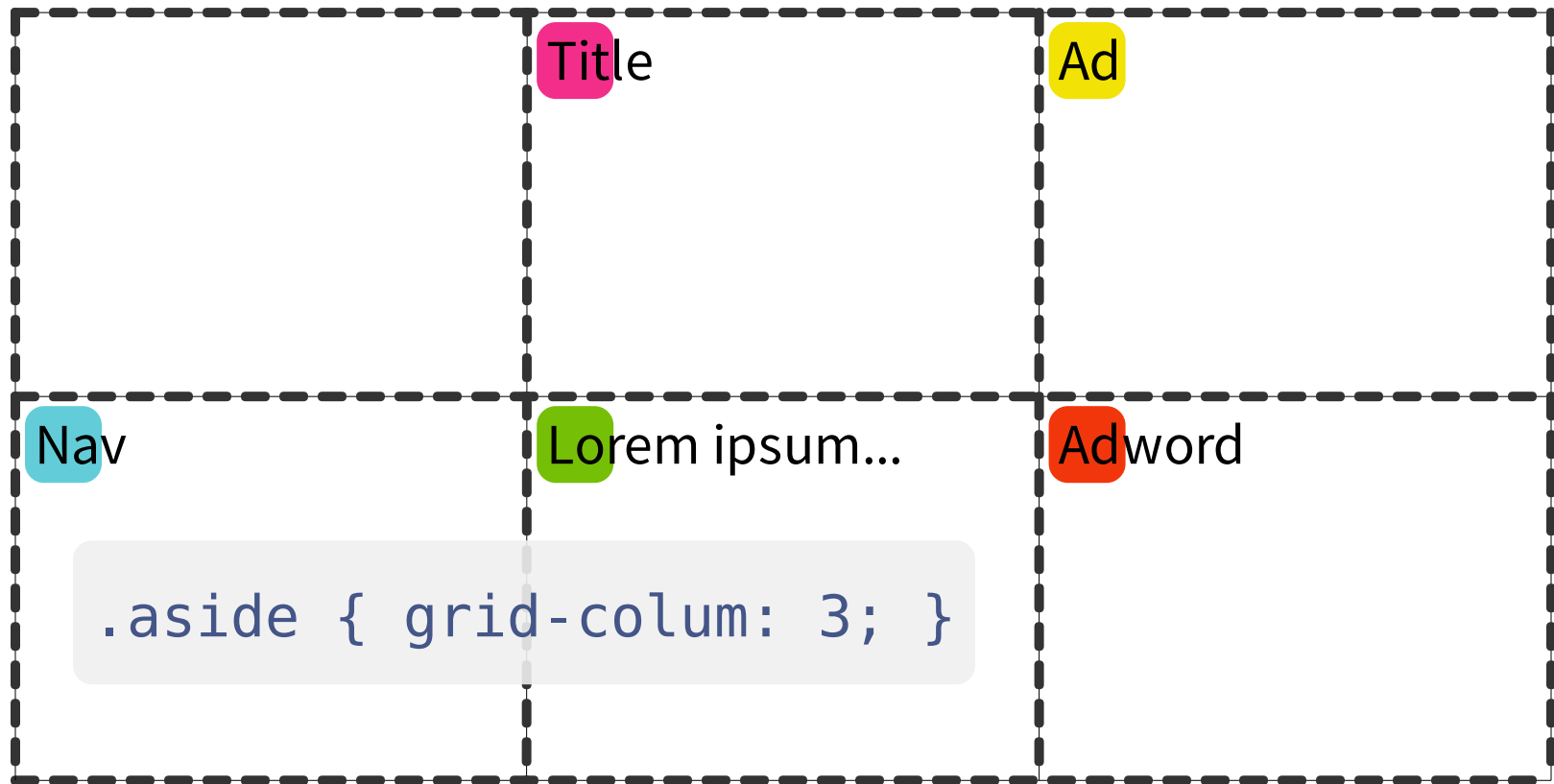


```
.main { grid-row: 2; grid-column: 2; }
```

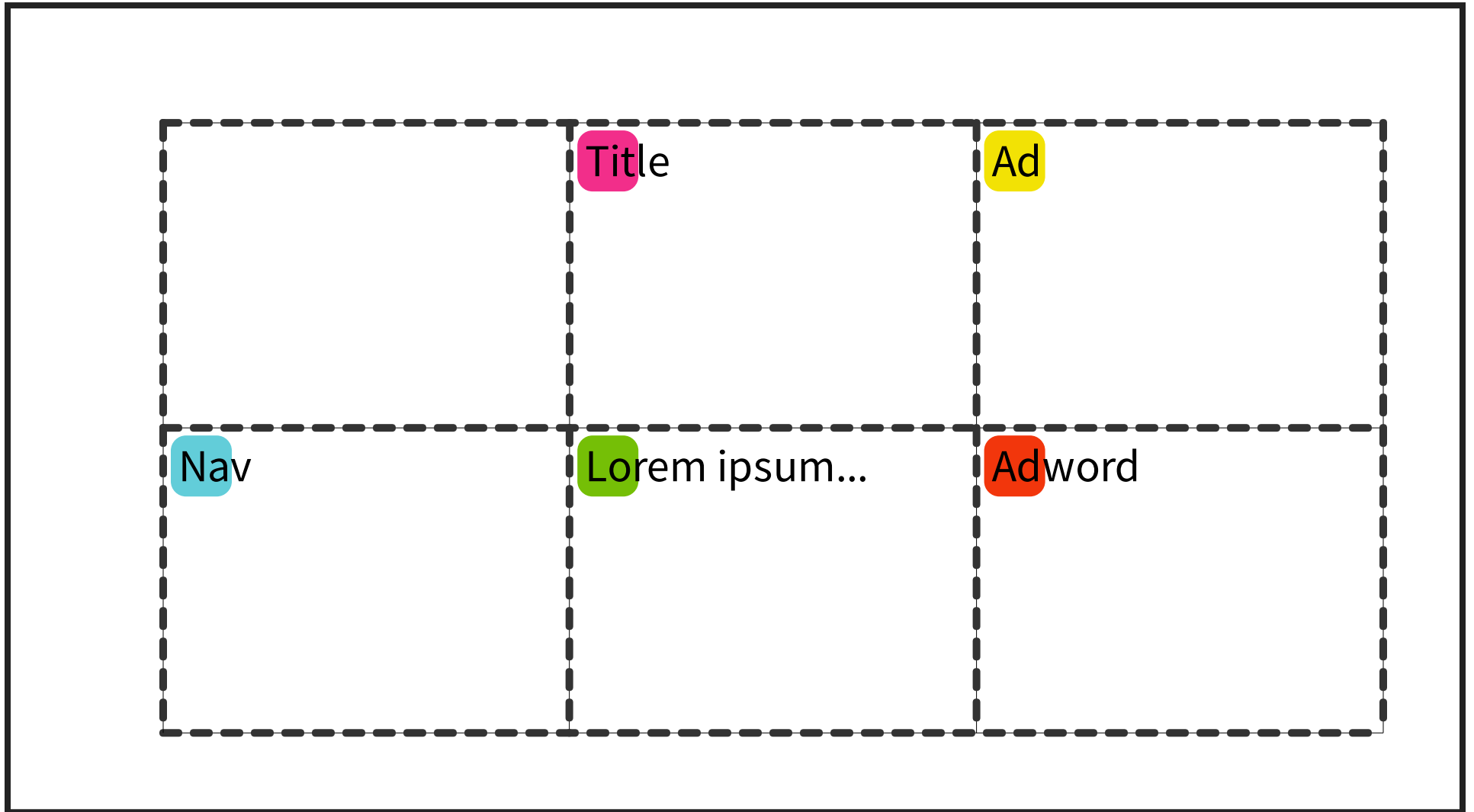
# COLOCACIÓN DE ÍTEMS



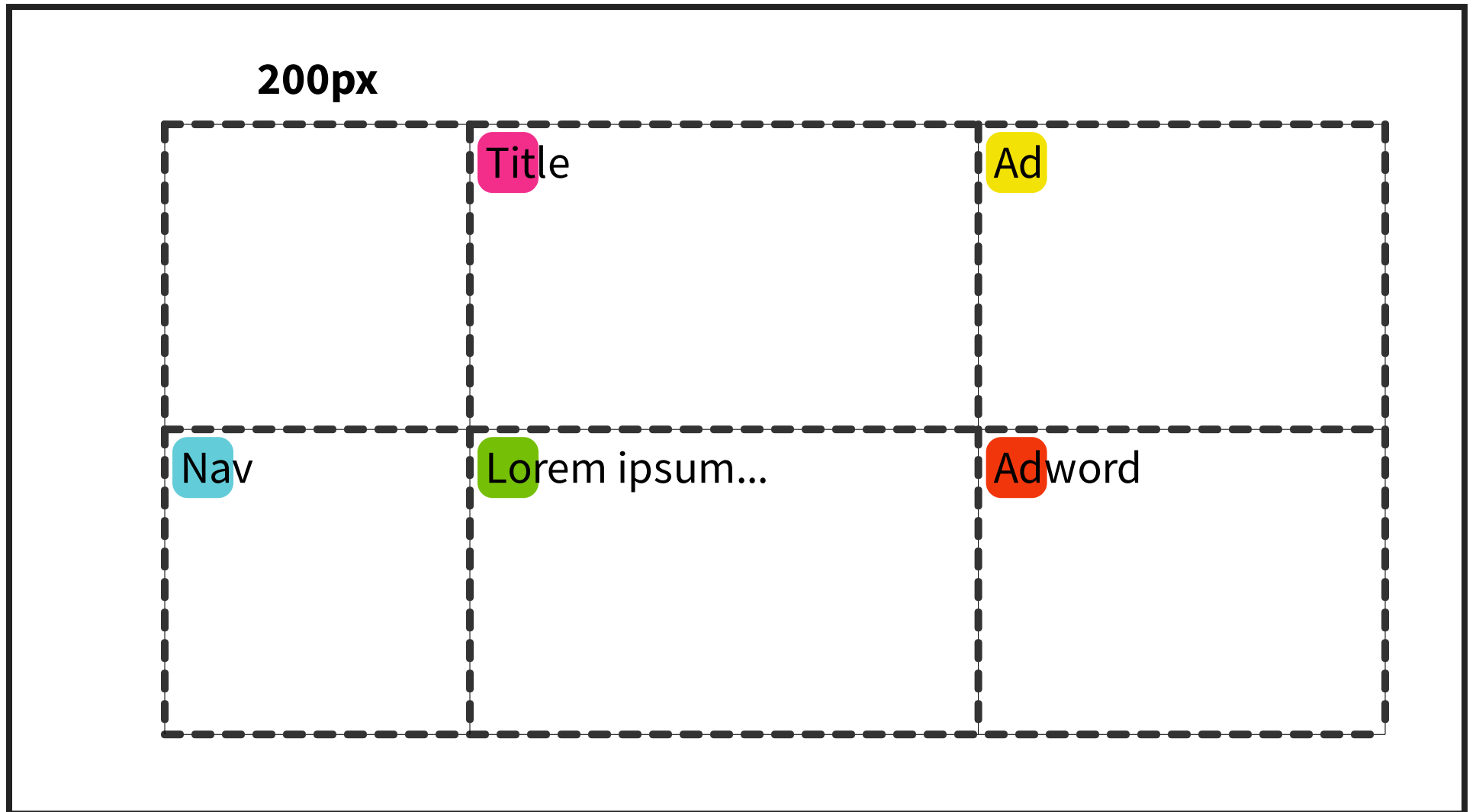
# COLOCACIÓN DE ÍTEMS



# COLOCACIÓN DE ÍTEMS

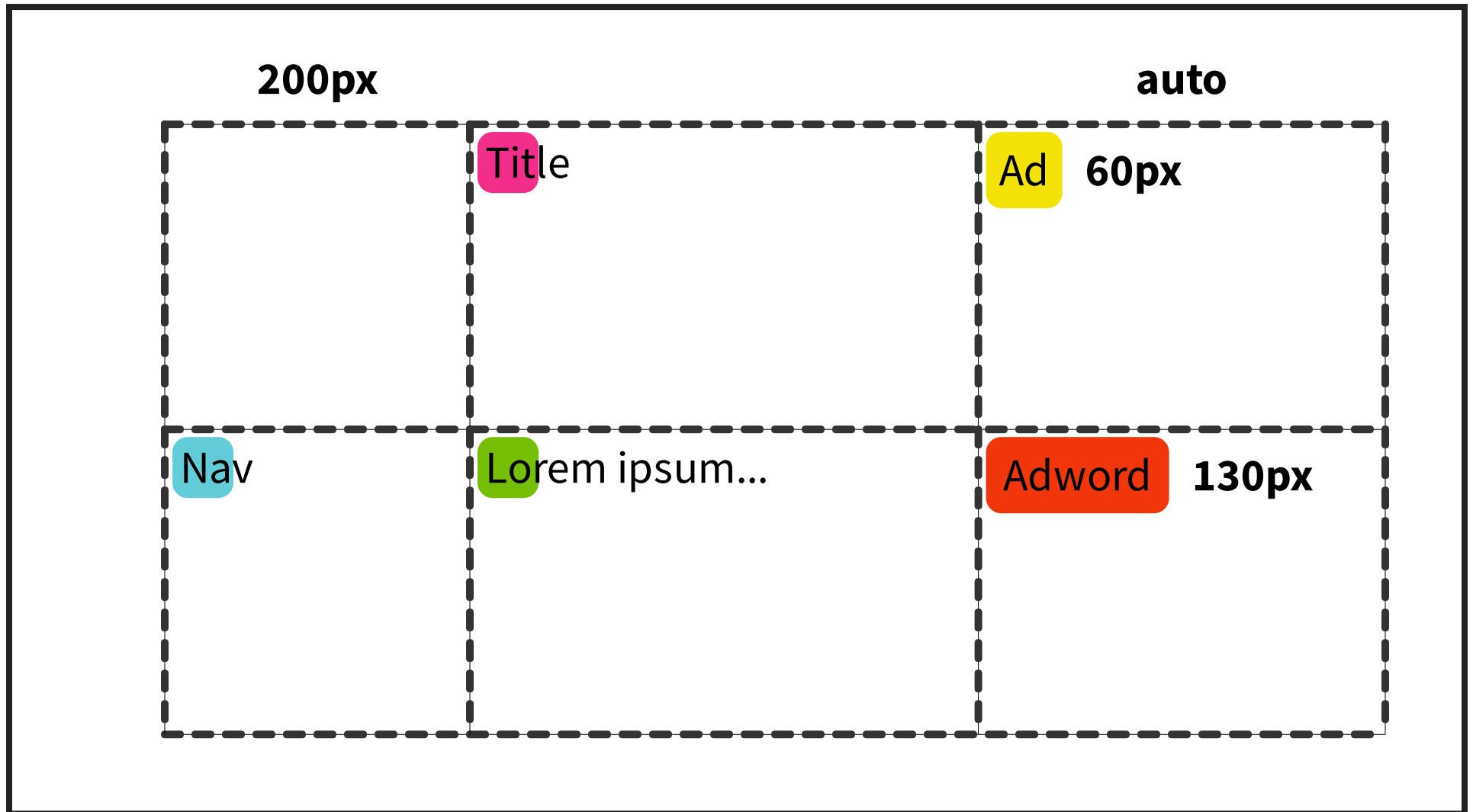


# COLUMNA DE TAMAÑO FIJO

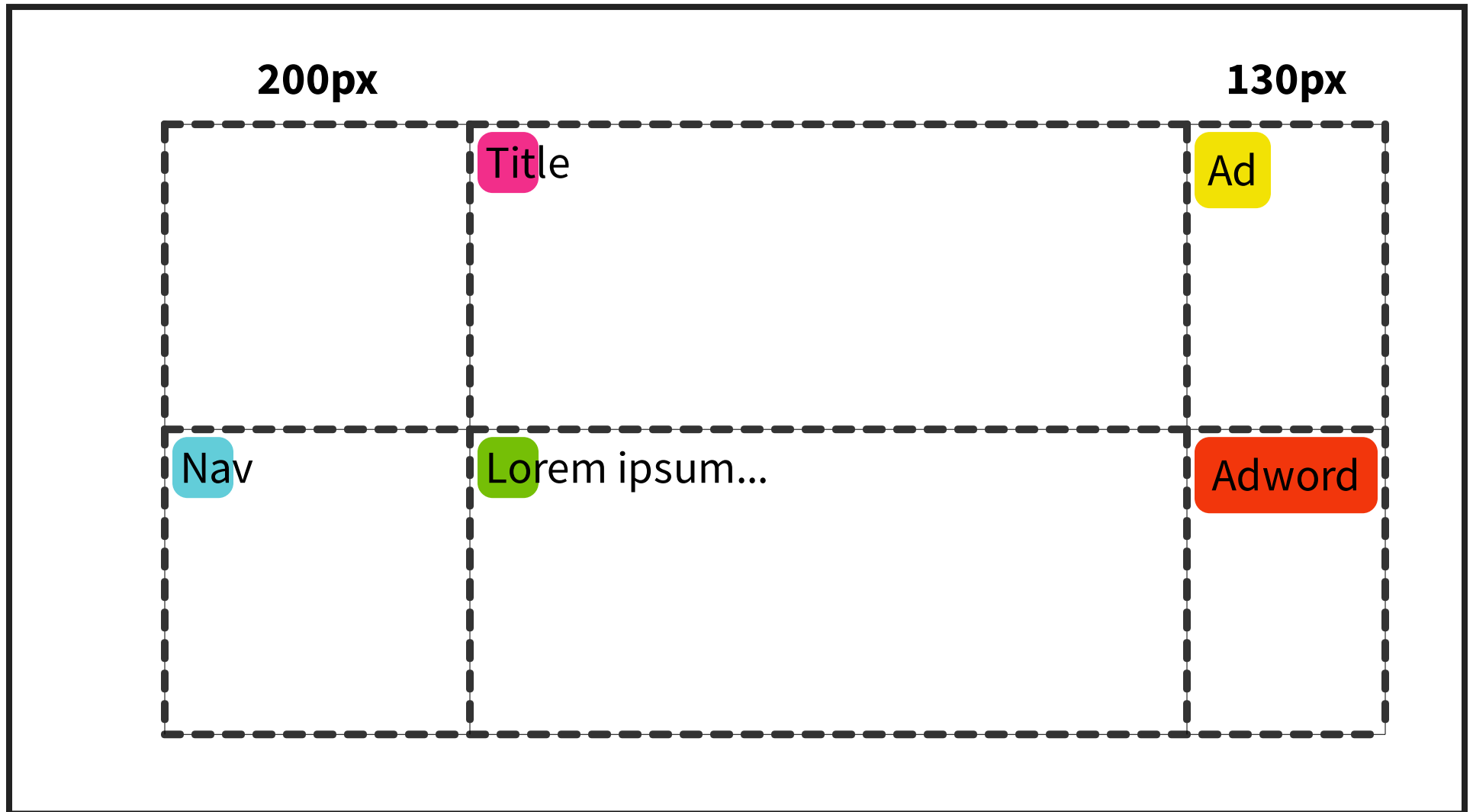




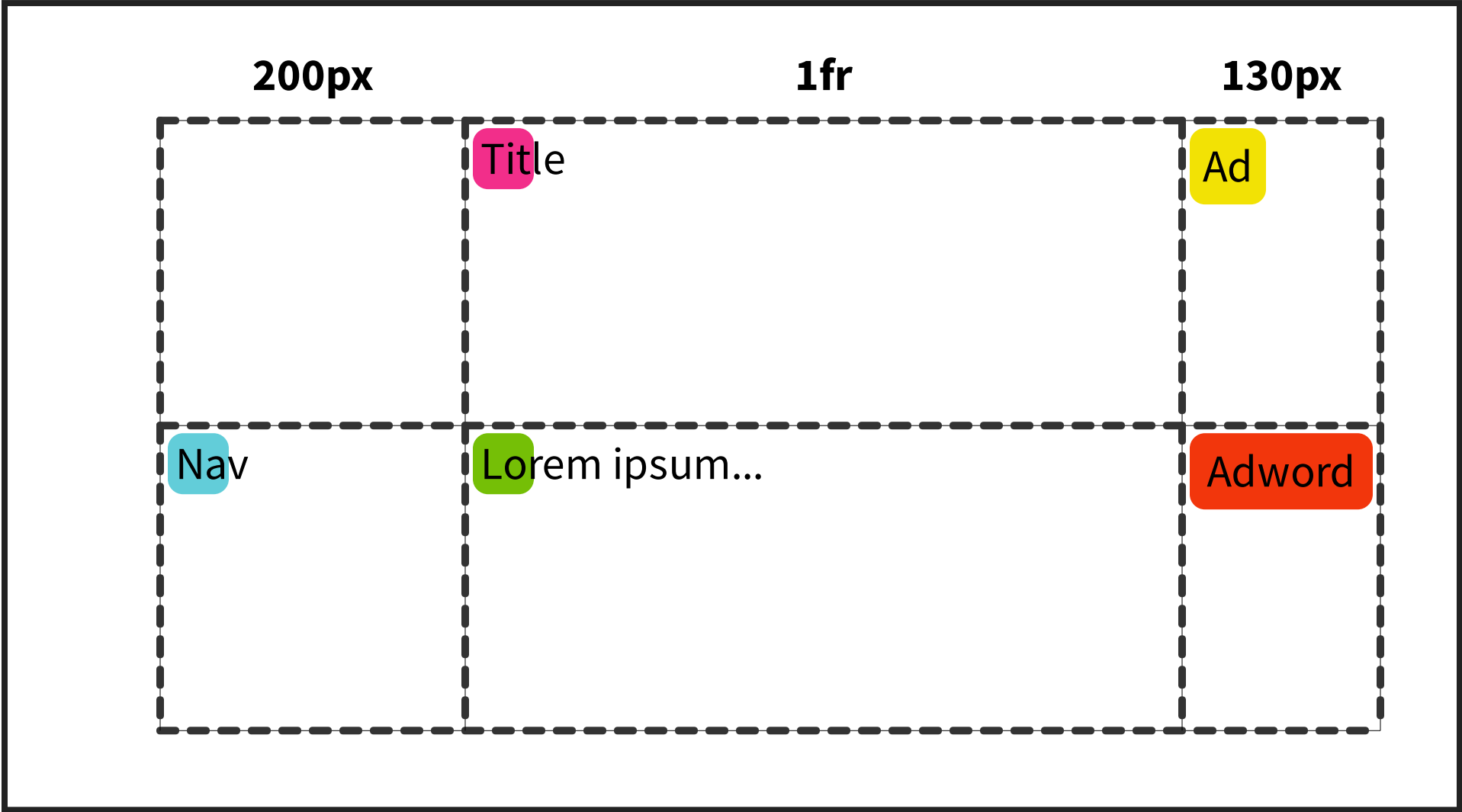
# COLUMNA BASADA EN CONTENIDO



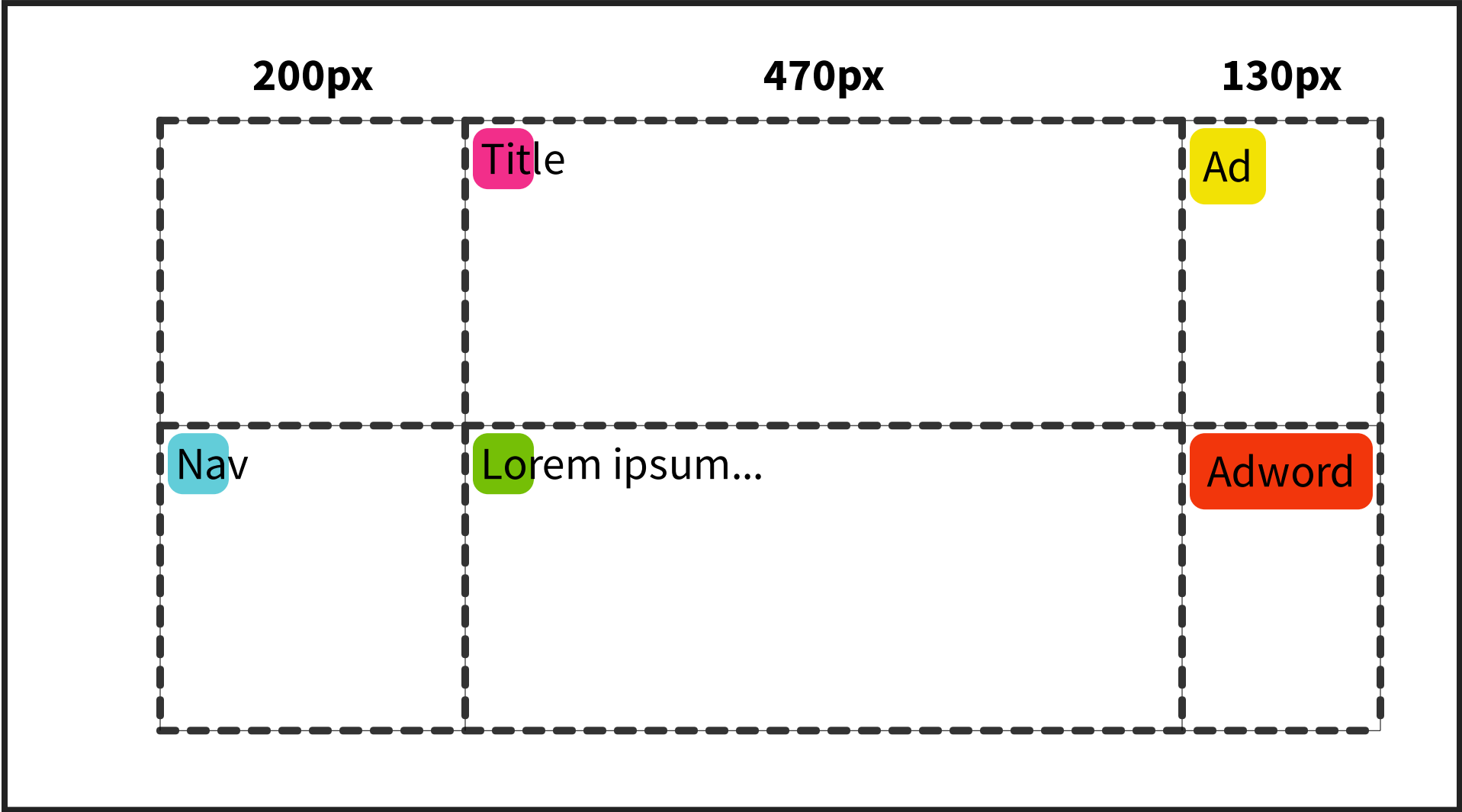
# COLUMNA BASADA EN CONTENIDO



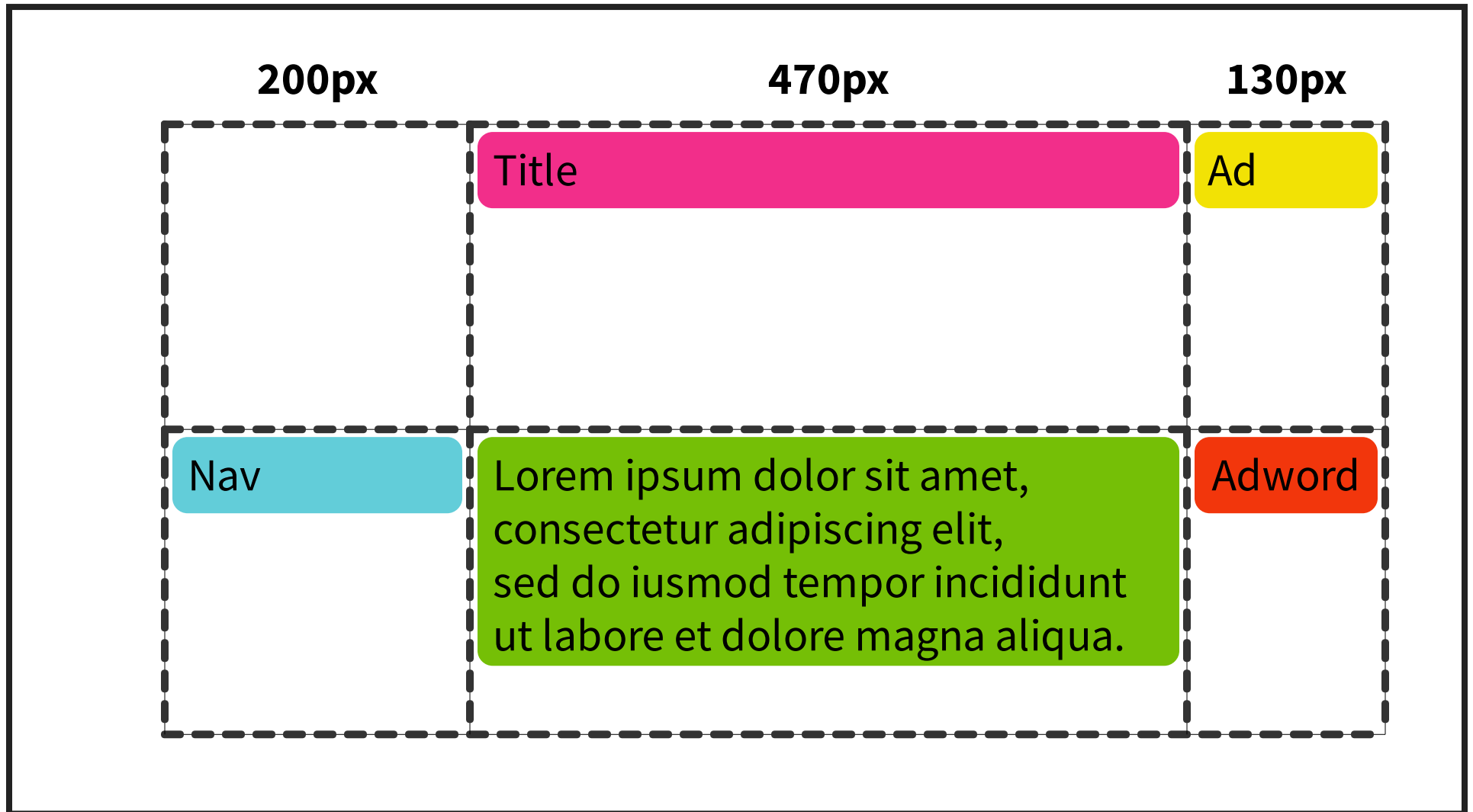
# COLUMN FLEXIBLE



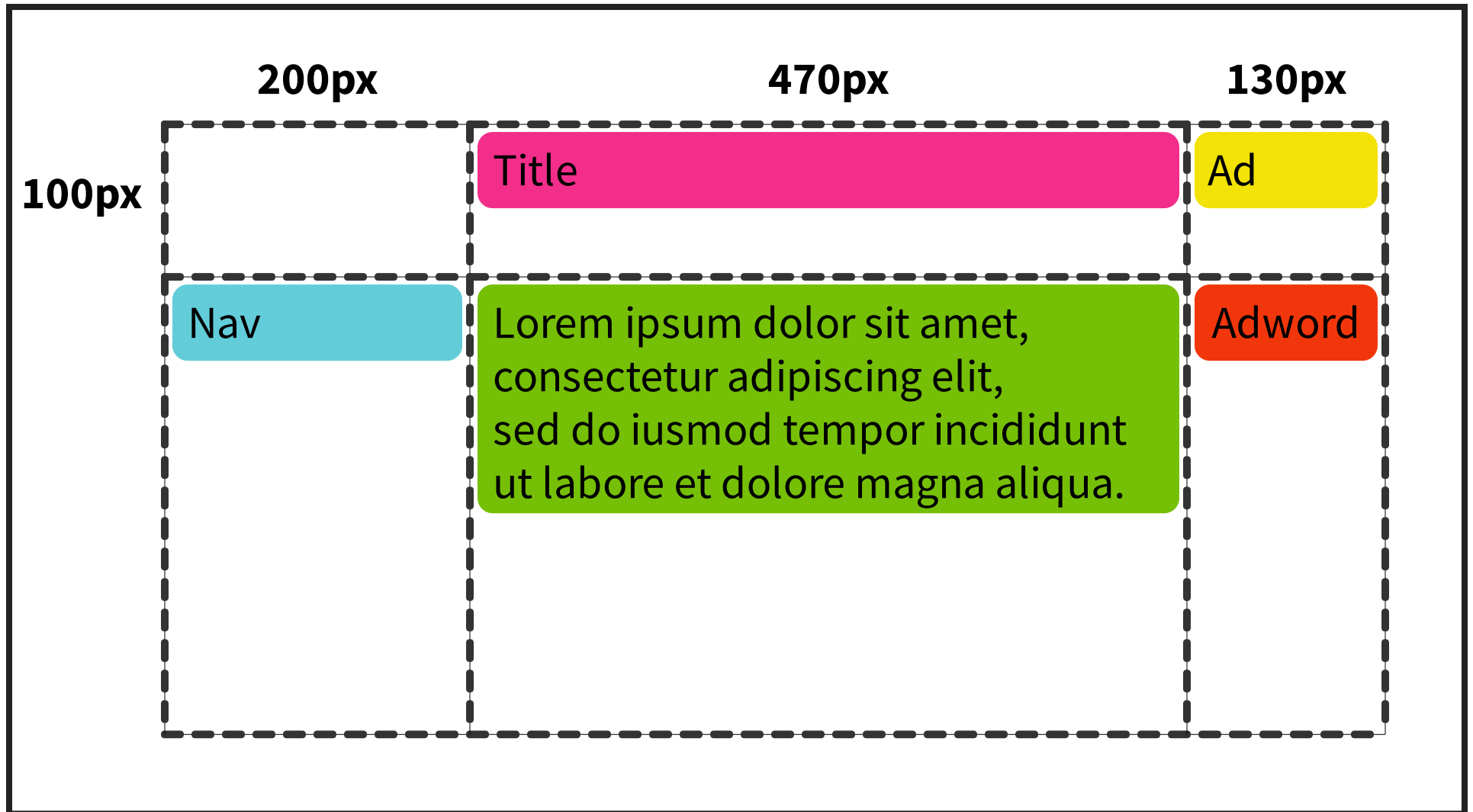
# COLUMN FLEXIBLE



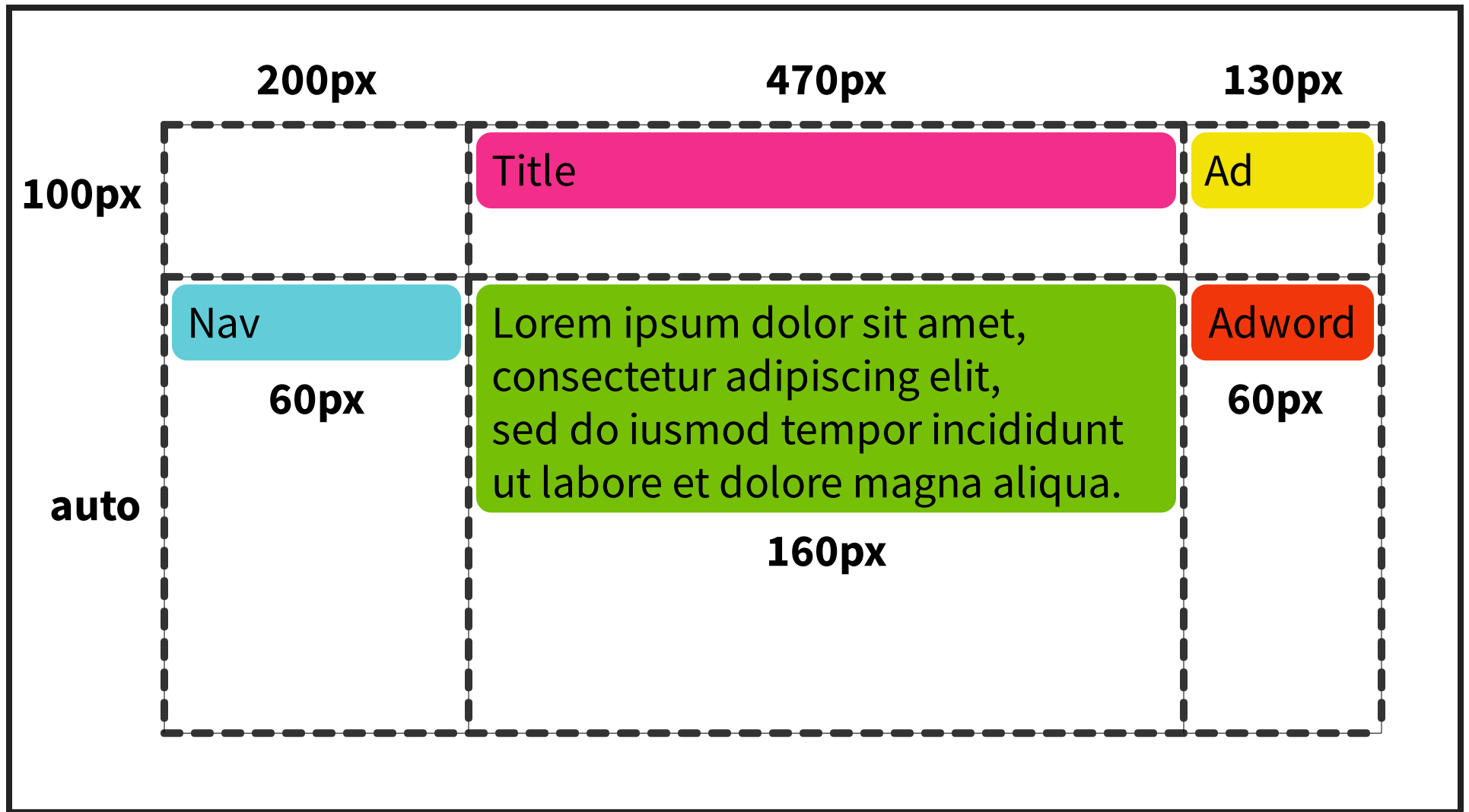
# HACER LAYOUT DE LOS ÍTEMS



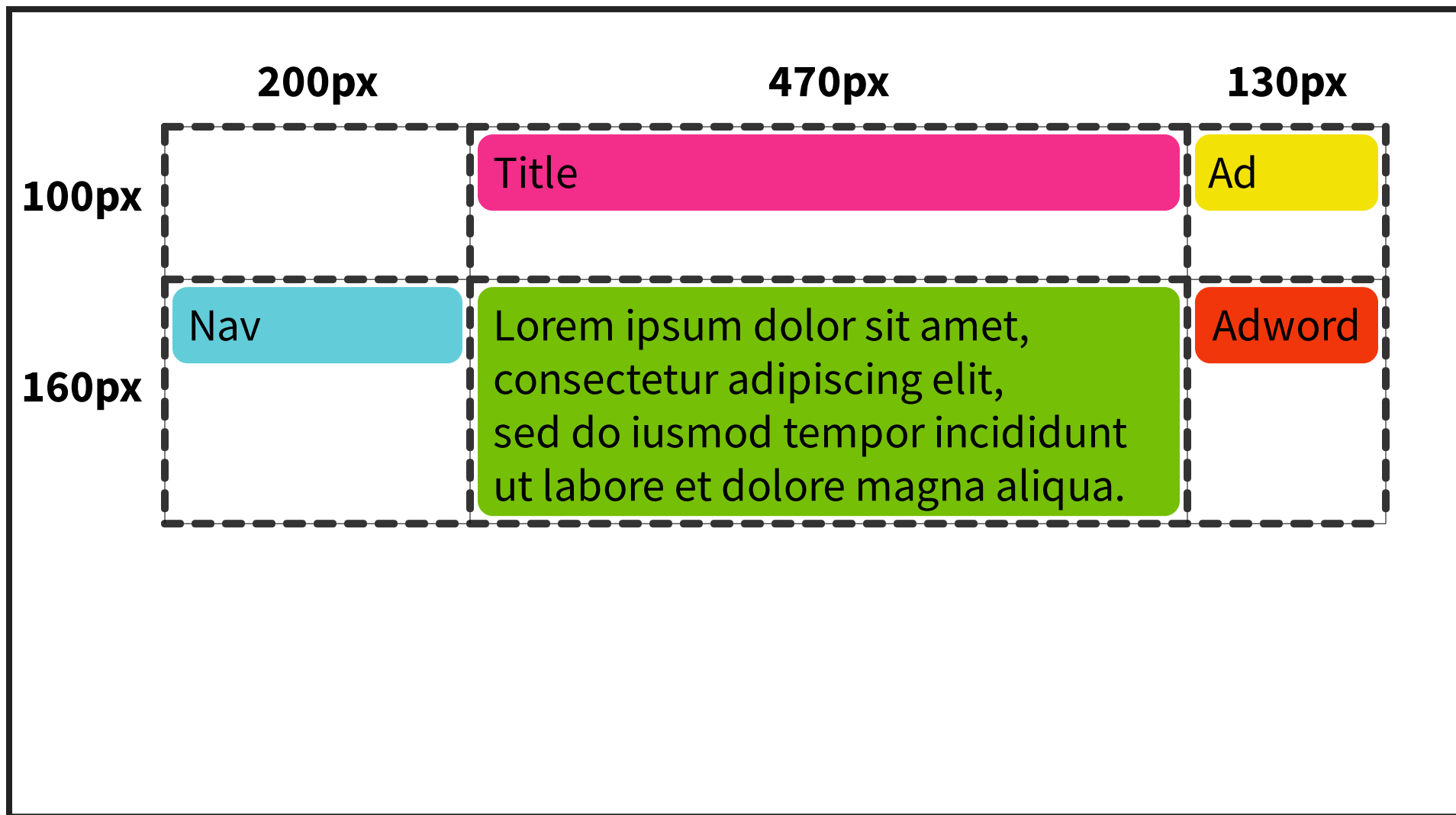
# FILA CON TAMAÑO FIJO



# FILA BASADA EN EL CONTENIDO

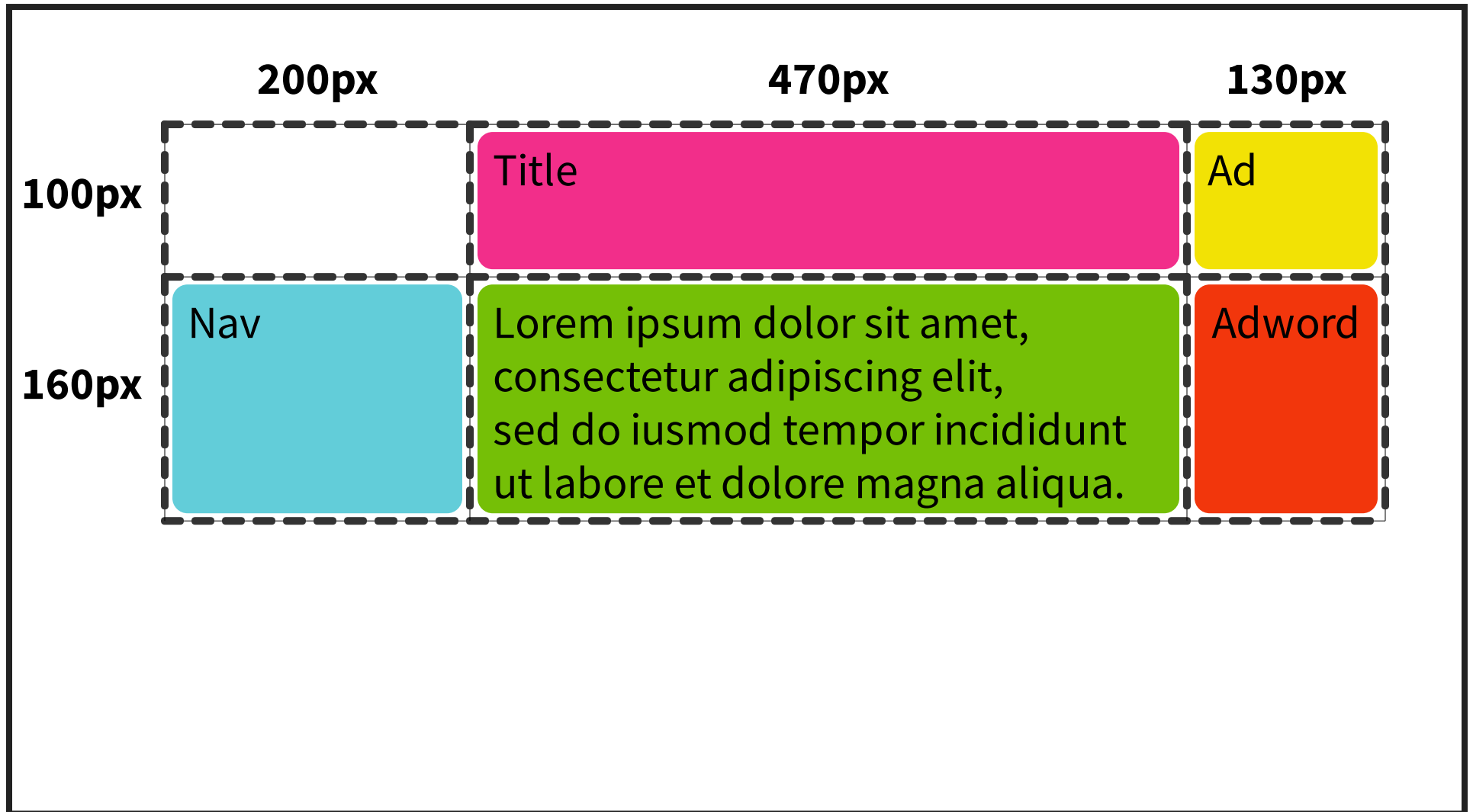


# FILA BASADA EN EL CONTENIDO





# ESTIRAR ÍTEMS





**RENDIMIENTO**

# TRACKS FIJOS VS BASADOS EN CONTENIDO

`200px` es más rápido que `auto`

# TRACKS BASADOS EN CONTENIDO VS FLEXIBLES

`auto` es más rápido que `1fr`

# ESTIRAR VERTICALMENTE

El estirado vertical (**stretch**) en ítems de tamaño automático es más lento que las otras opciones (ej. **start**)



# NO ESTIRAR HORIZONTALMENTE

Evitar que los ítems de tamaño automático se estiren horizontalmente (e.g. `start`) es más lento





**ESTADO**

# ESPECIFICACIÓN W3C

CSS Grid Layout - <https://drafts.csswg.org/css-grid/>

La empezó Microsoft en 2010

Último *Working Draft* 19 Mayo 2016

W3C Test Suite



# ¿PUEDO USAR GRID?

## CSS Grid Layout - WD

Global 0% + 7.46% = 7.46%  
 unprefix: 0%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for lay out into columns and rows using a set of predictable sizing behaviors

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			1 29						
			1 45						
			1 48					4.3	
		3 45	1 49			8.4		4.4	
8		3 46	1 50			9.2		4.4.4	
2 11	2 13	3 47	1 51	9.1	1 37	9.3	8	50	50
	2 14	3 48	1 52	TP	1 38				
		3 49	1 53		1 39				
		3 50	1 54						

# NAVEGADORES



Implementación  
antigua desde IE 10

Con prefijo: `-ms`



Casi lista

🚩 *Experimental Web Platform  
Features*



Casi como Chrome

Activada por defecto en  
Safari Technology Preview



Casi lista también

🚩 *layout.css.grid.enabled*

Polyfill: <https://github.com/FremyCompany/css-grid-polyfill>

# EJEMPLOS

- <http://igalia.github.io/css-grid-layout/> by Igalia
- <http://gridbyexample.com/> by Rachel Andrew

# AGRADECIMIENTOS



**igalia**

**Bloomberg**

Igalia y Bloomberg trabajando juntos por mejorar la web

C S S **¡GRACIAS!** R I D

- Twitter: [@regocas](#)
- Correo: [rego@igalia.com](mailto:rego@igalia.com)
- Blog: <http://blogs.igalia.com/mrego/>

L A Y O U T