



Plain Concepts Tech Day

20.12.2017

CSS Grid Layout - ¡Se acabaron los hacks, larga vida al CSS!

IGNACIO VILLANUEVA - @ignaciodenuovo
FRONT-END DEVELOPER

Plain Concepts Tech Day

plain concepts³

- 9:00 – 9:45 CSS Grid Layout - ¡Se acabaron los hacks, larga vida al CSS!
- 9:45 - 10:30 Desarrollo de aplicaciones multiplataforma utilizando Xamarin
- 10:30 - 11:15 Docker: de principante a pro
- 11:15 - 11:25 ¿Qué es la Fundación Theodora?
- 11:25 - 11:45 Break
- 11:45 - 12:30 Los autobuses de madrid en tiempo real con Azure Relay, .Net Core y una Raspberry Pi
- 12:30 - 13:15 Swagger Code Generation: Cuando front encontró a back
- 13:15 - 14:00 Entrega continua con Visual Studio Team Services

Qué veremos

1. CSS Layout (Un poco de historia)
2. ¿Qué debemos conocer?
3. Tablas
4. Floats
5. Contenido flexible
6. Inline-block
7. Display: table
8. Flexbox
9. CSS Grid Layout



Ignacio Villanueva

FRONT-END DEVELOPER

Front-end Developer especializado en arquitectura y front-end performance and responsive web design.

Co-host en [WeCodeSign Podcast](#) y apasionado del Open Source.

[@ignaciodenuovo](#)

[www.ignaciodenuovo.com](#)

1. CSS Layout (Un poco de historia)

**CSS
IS
AWESOME**

2. ¿Qué debemos conocer?

¿Y si no sabemos CSS?

- **Box Model**
- **Position:** static, relative, absolute, fixed, sticky...
- **Unidades de medida:** px, %, em, rem, fr, ch...

Cosas que DEBEMOS plantearnos

- ¿Cuándo va a cambiar el **contenido**?
- ¿Cuándo va a cambiar el **viewport**?
- ¿Cuándo va a cambiar el **contexto**?

Esto es la WEB

- No tenemos tamaños fijos
- Cada día salen nuevos **dispositivos**
- Tenemos distintos **usuarias/os**



3. Tablas

¿Quién no se ha peleado con TABLAS?

```
▼<table cellspacing="0" cellpadding="0" class="GH3H4YHCMC" style>
  ▼<tbody>
    ▼<tr>
      ▼<td align="left" style="vertical-align: top; ">
        ▼<div class="GH3H4YHCLC">
          <div class="GH3H4YHCNC">Popular This Week</div>
        ▼<table cellspacing="0" cellpadding="0">
          ▼<tbody>
            ▼<tr>
              ►<td align="left" style="vertical-align: middle; ">...</td>
              ►<td align="left" style="vertical-align: middle; ">...</td>
              ►<td align="left" style="vertical-align: middle; ">...</td>
            </tr>
          </tbody>
        </table>
      </div>
    </td>
  </tr>
  <tbody>
</table>
▼<table cellspacing="0" cellpadding="0" class="GH3H4YHCMC" style="display: none; ">
  - ... -
```

Las TABLAS son para datos

Si no lo hacéis así, estáis haciéndolo mal

PROBLEMAS

- Baja **mantenibilidad**
- Problemas de **accesibilidad**
- **Marcado** restringido

4. Floats

¿Qué es y qué no es un FLOAT?

LEFT

RIGHT

Los FLOATS se usaban para todo

Hemos flotado imágenes, creado layouts, Grids...

PROBLEMAS

- Los elementos flotados

SOLUCIONES

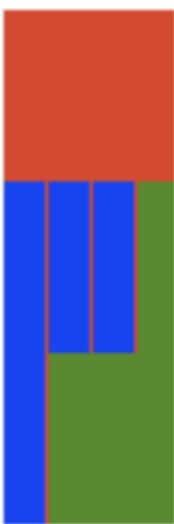
- Clearfix (**Hack**)
- Display: flow-root
- Overflow: hidden (**Ojo**)

5. Contenido flexible

Tener MUCHO cuidado con el sub-pixel rendering

<https://www.palantir.net/blog/responsive-design-s-dirty-little-secret>

Opera 9 Safari 3 IE 6 IE 7 Firefox 3 Firefox 2



12px
12px
12px
12px

12px
12px
12px
12px

13px
13px
13px
13px

12.5px
12.5px
12.5px
12.5px

12.4667px
12.4667px
12.4667px
12.4667px

<https://johnresig.com/blog/sub-pixel-problems-in-css/>

6. Inline-block

PROBLEMAS

- Hacks por todos lados 

7. Display: table

display: table nos ha salvado mil veces



Ejemplo de display: table

```
<h2>Display: Block</h2>
<section>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</section>

<h2>Display: Table / Table-Cell</h2>
<section class="table">
  <div></div>
  <div>(width 40%)</div>
  <div></div>
  <div></div>
  <div></div>
</section>
```

```
section {
  background: red;
  margin: 1em 0;
}

div {
  background: lightblue;
  padding: 1em;
}

.table {
  border-spacing: 0.5em;
  display: table;
  width: 100%;
}

div {
  display: table-cell;
}

:nth-child(2) {
  width: 40%;
}
```

Display: Block

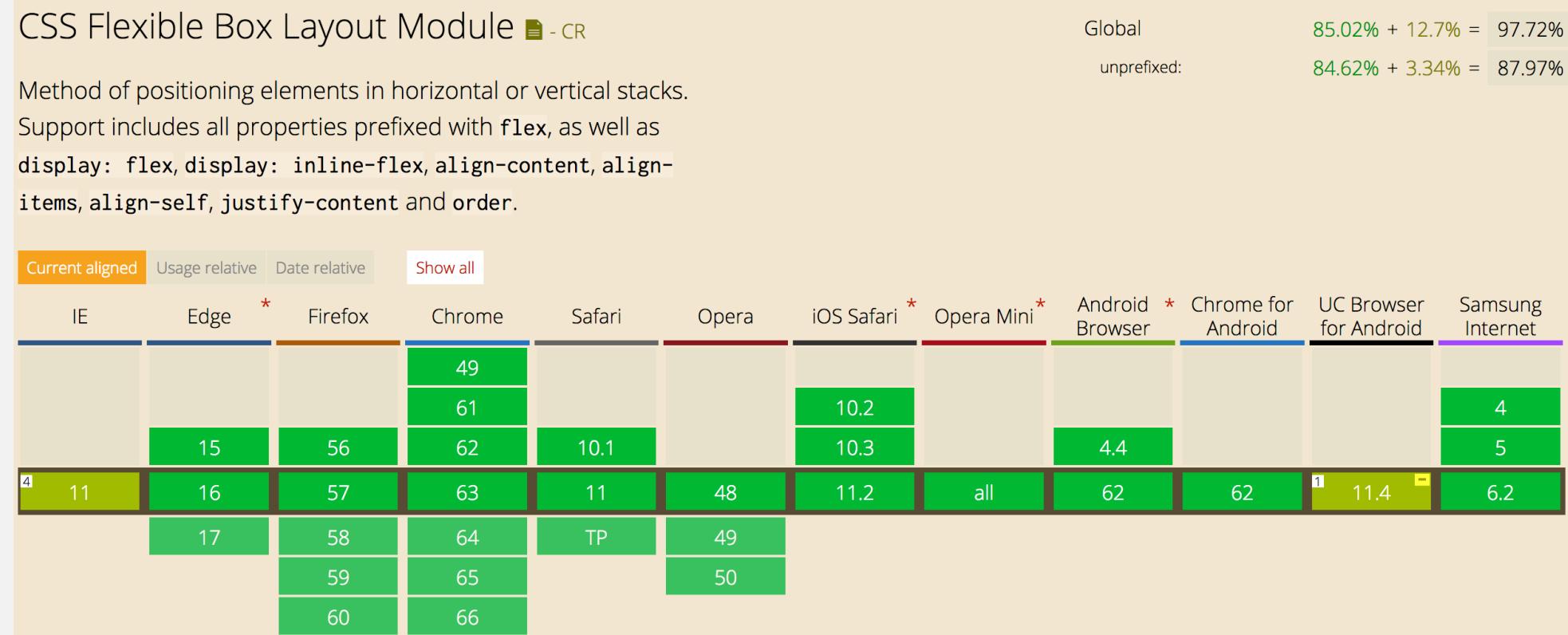
```
div  
div  
div  
div  
div
```

Display: Table / Table-Cell



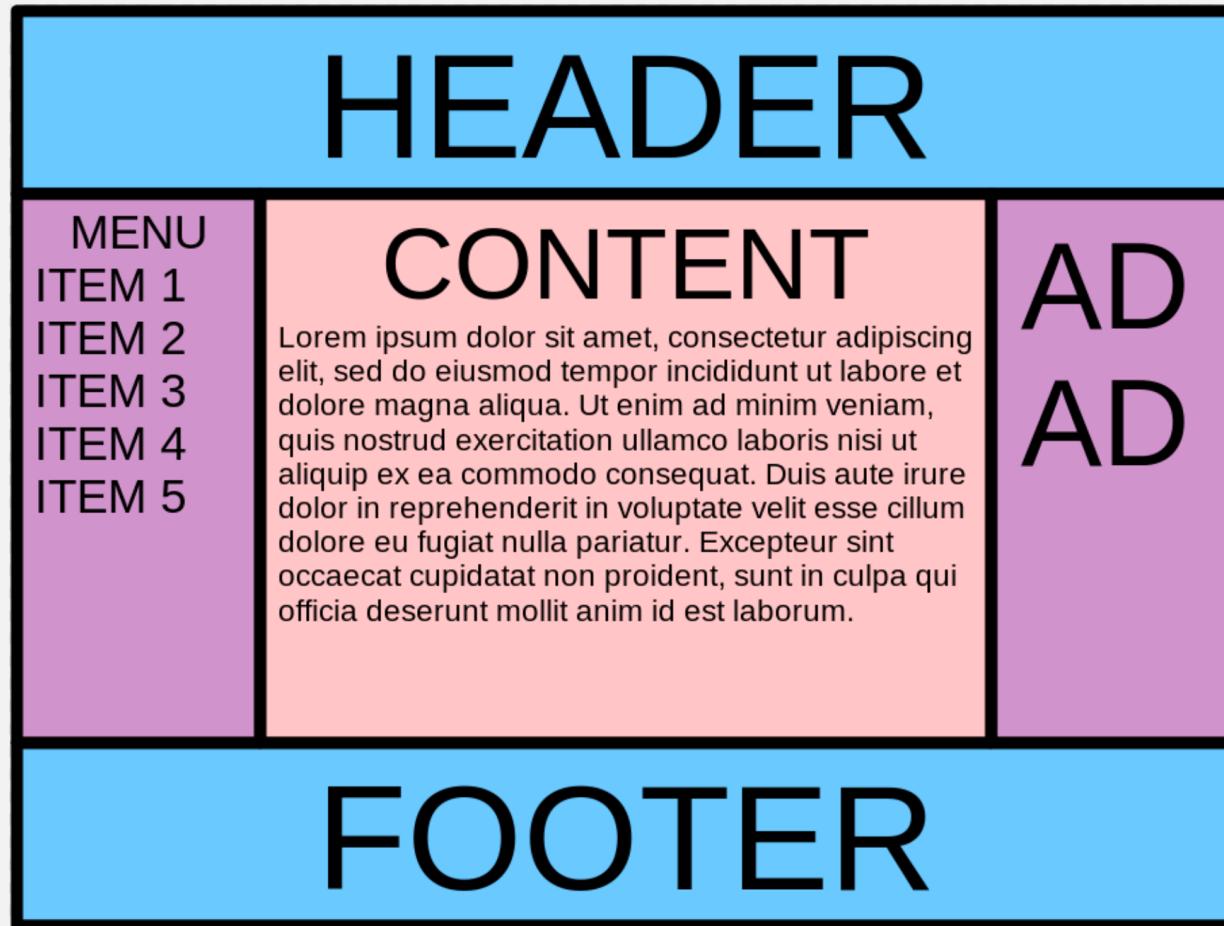
8. Flexbox

Caniuse.com de FLEXBOX



Lo más importante es el PADRE

Manejamos el espacio extra



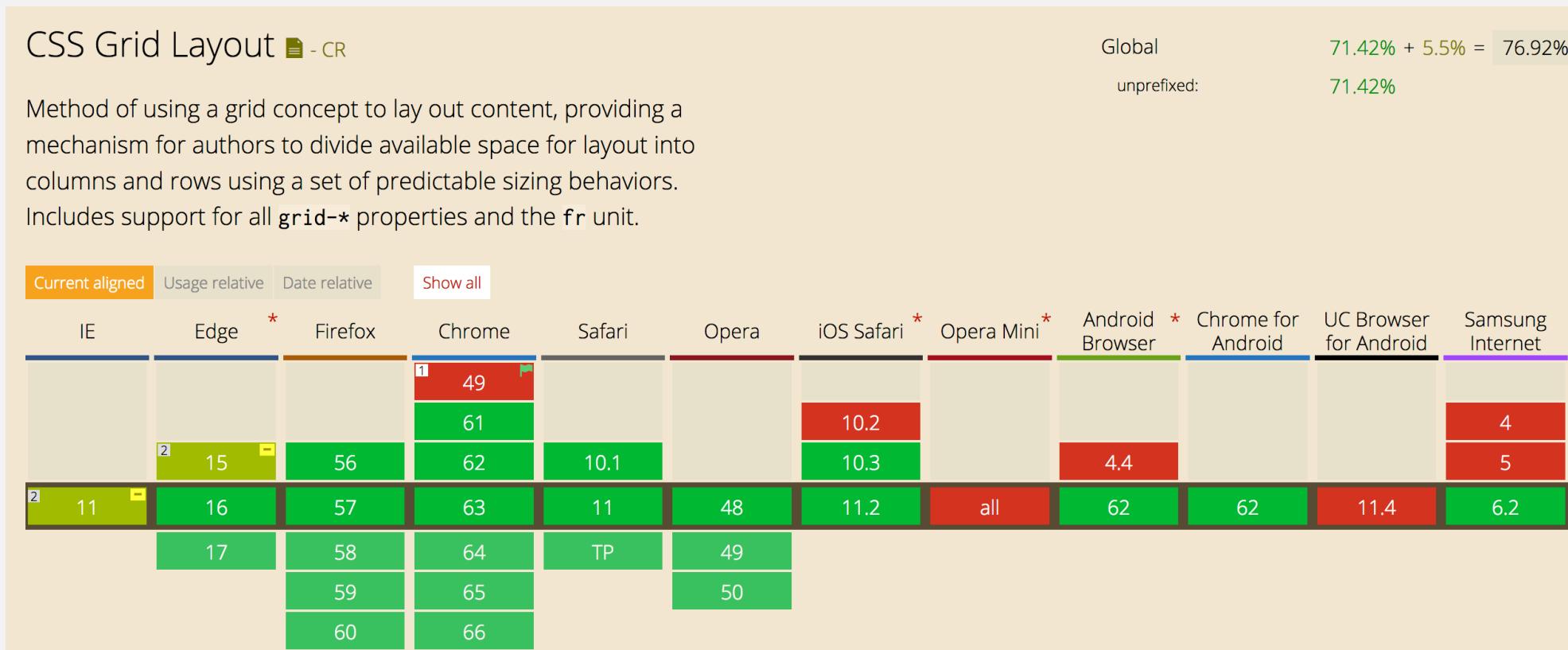
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

A tener en CUENTA

- Flujo en **una** dimensión
- ¡Ojo con el **nesting!** (importa y mucho)
- Layouts **inline** super fáciles
- Poco **performance** para layouts grandes (paginas)

9. CSS Grid Layout

Caniuse.com de GRID LAYOUT



Lo más importante es el PADRE

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

¡Un momento! ¿Y los GRIDS?

- 12 **columnas** iguales
- Ayuda con la **consistencia**
- Más **hacks** 😱
- **Columnas y márgenes**

960 - Bootstrap - Susy

Unidades

% defines a fraction of total space
vw defines a fraction of the viewport
fr defines a fraction of remaining space

Columnas

```
.container {  
    display: grid;  
    grid-template-columns: 300px 1fr 300px;  
}
```

Filas

```
.container {  
    display: grid;  
    grid-template-columns: 300px 1fr 300px;  
    grid-template-rows: 200px 200px 200px;  
}
```

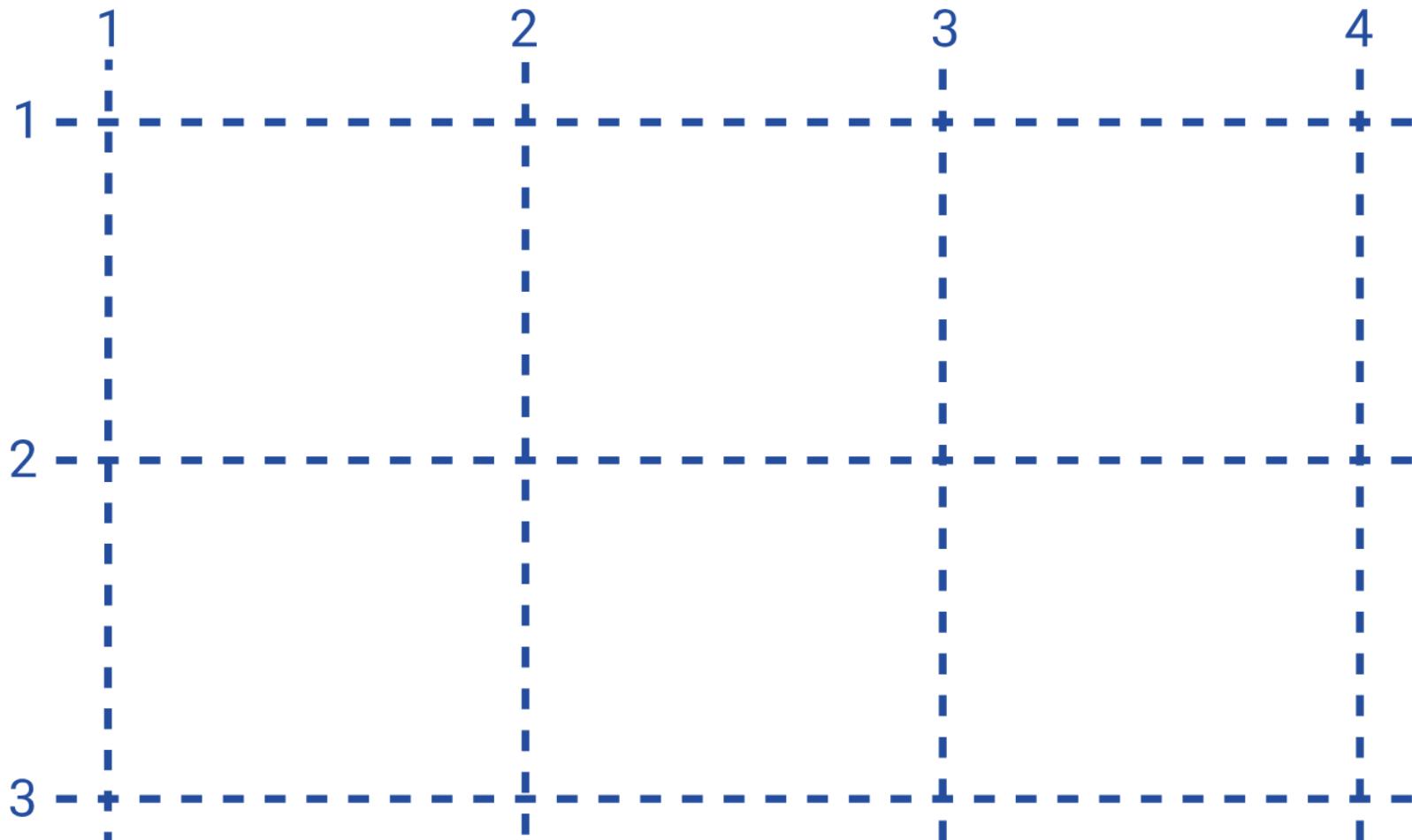
Filas automáticas

```
.container {  
    display: grid;  
    grid-template-columns: 300px 1fr 300px;  
    grid-template-rows: 200px 200px 200px;  
    grid-auto-rows: 350px;  
}
```

Gap

```
.container {  
    display: grid;  
    grid-column-gap: 2rem; /* gaps de columna */  
  
    grid-row-gap: 33px; /* gaps de fila */  
  
    grid-gap: 33px; /* gaps */  
}
```

Lines



Lines (de nuevo)

```
grid-area: 2 / 2 / 3 / 3;  
/* grid-row-start / grid-column-start / grid-row-end / grid-column-end */
```

Span

```
.item {  
  grid-row: 1 / span 2;  
}
```

Repeat

```
.container {  
    display: grid;  
    grid-template-columns: 100px repeat(3,  
    1fr) 300px repeat(3, 50px);  
    grid-gap: 33px;  
}
```

minmax()

```
.container {  
    grid-template-columns: minmax(auto, 1fr) 2fr 3em;  
    grid-template-rows: minmax(200px, auto);  
}
```

grid-auto-flow

grid-auto-flow: dense;

auto-fill and auto-fit

```
grid-template-columns: repeat(auto-fill, 150px);  
/* empty space */
```

```
grid-template-columns: repeat(auto-fit, 150px);  
/* collapse empty tracks */
```

repeat(), minmax(), and auto-fit / auto-fill

```
.container {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));  
  grid-template-rows: 200px 200px 200px;  
  grid-gap: 16px;  
}
```

Posicionando items

```
.item {  
    grid-column-start: tres;  
    grid-column-end: cuatro;  
    grid-row-start: dos;  
    grid-row-end: tres;  
}
```

Grid areas

```
.container {  
    grid-template-areas: "header header"  
                        "content sidebar"  
                        "footer footer";  
    grid-template-rows: 150px 1fr 100px;  
    grid-template-columns: 1fr 200px;  
}
```

¿Qué MÁS?

- Items **anónimos**
- Nested grids
- Order 🤯
- Dirección (ltr - rtl)
- **@supports**
- **CSS align**

10. Recursos

¡Por donde EMPIEZO?

- Awesome Grid
- Colección de CodePens
- Artículo en Octuweb
- La BBC como ejemplo

plain concepts

¡GRACIAS!

www.ignaciодenuevo.com

@ignaciодenuevo



¡Estamos CONTRATANDO!

careers@plainconcepts.com

www.plainconcepts.com/careers

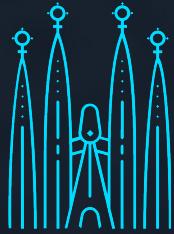
plain concepts

www.plainconcepts.com



MADRID

Paseo de la Castellana 163, 10°
28046 Madrid. España
T. (+34) 91 5346 836



BARCELONA

Carrer Compte d'Urgell 240 4° A
08036 Barcelona. España
T. (+34) 93 7978 566



BILBAO

Calle Ledesma 10-bis 3°
48001 Bilbao. España
T. (+34) 94 6073 371



SEVILLA

Avenida de la innovación s/n
Edificio Renta Sevilla, 3° A
41020 Sevilla. España
T. (+34) 600 58 75 43



DUBAI

Dubai Internet City. Building 1
73030 Dubai. EAU
T. (+971) 4 551 6653



LONDON

Impact Hub Kings Cross
24B York Way, N1 9AB
London. UK



SEATTLE

8201 164th Ave NE
Redmond, WA 98052
Washington
T. (+1) 206 708 1285