

Inicio (/css/) / Diseño (Layout) (/css/layout-responsive.html)
/ Grid Layout (/css/layout-responsive/grid-layout.html) / -- Qué es Grid Layout

-- Qué es Grid Layout

CSS Grid Layout es una ^{cuadrícula} especificación que sirve para crear rejillas bidimensionales. Te da la posibilidad de establecer en una página web una cuadrícula y colocar, con precisión, los elementos de la página dentro de ella.

El ^{cuadrícula} módulo CSS de Rejilla (*Grid Layout*) y el de Caja Flexible (*FlexBox*) son complementarios. La maquetación, tal y como la conocemos ha sufrido una enorme revolución desde que todos los navegadores actuales más utilizados soportan estos dos módulos (ver en Can I Use: Grid Layout (<http://caniuse.com/#search=Grid>) y FlexBox (<http://caniuse.com/#search=Flex>)).

- CSS Grid Layout ^{cuadrícula} actúa sobre la propia caja grid estructurando su espacio, y FlexBox no. En flexbox todo está enfocado a la gestión de sus ítems de forma lineal, mientras que en Grid Layout sus ítems se gestionan de forma bidimensional.
- CSS Grid Layout genera una rejilla que divide el espacio del propio elemento en distintos compartimentos desde el CSS (sin necesidad de tocar el HTML) y controla la ocupación de estas zonas virtuales por los ítems del grid.

Como hemos dicho, Flexbox está pensado para el diseño en una sola dimensión, así que las cosas se pueden manejar en una línea, un ítem al lado de otro de forma ininterrumpida.

Mientras que CSS Grid Layout está diseñado para una disposición en dos dimensiones, es decir, sus ítems no tienen que ubicarse uno al lado del otro, pueden ocupar zonas o áreas rectangulares.

Se pueden utilizar conjuntamente: Grid Layout para colocar los bloques principales de la página y FlexBox para los elementos de interfaz de usuario más pequeños.

Nota: las propiedades clear, float y vertical-align no serán aplicadas en los elementos con este tipo de display. **grid o flexbox**

Conceptos básicos que usa CSS Grid Layout

Grid container o **contenedor padre (grid)**: Es el **elemento contenedor** o **caja** en el que se define la rejilla o **cuadrícula**.

Grid Item o **contenedor hijo**: Los **elementos hijos directos** de la **caja padre grid**.

Grid lines o **líneas de la rejilla**: Son los **divisores horizontales** y **verticales**. Se utilizan para crear "grid tracks", "grid cells" y "grid areas". Por defecto y para identificarlas tienen un **índice numérico**, también, opcionalmente, se les puede **dar nombres específicos**.

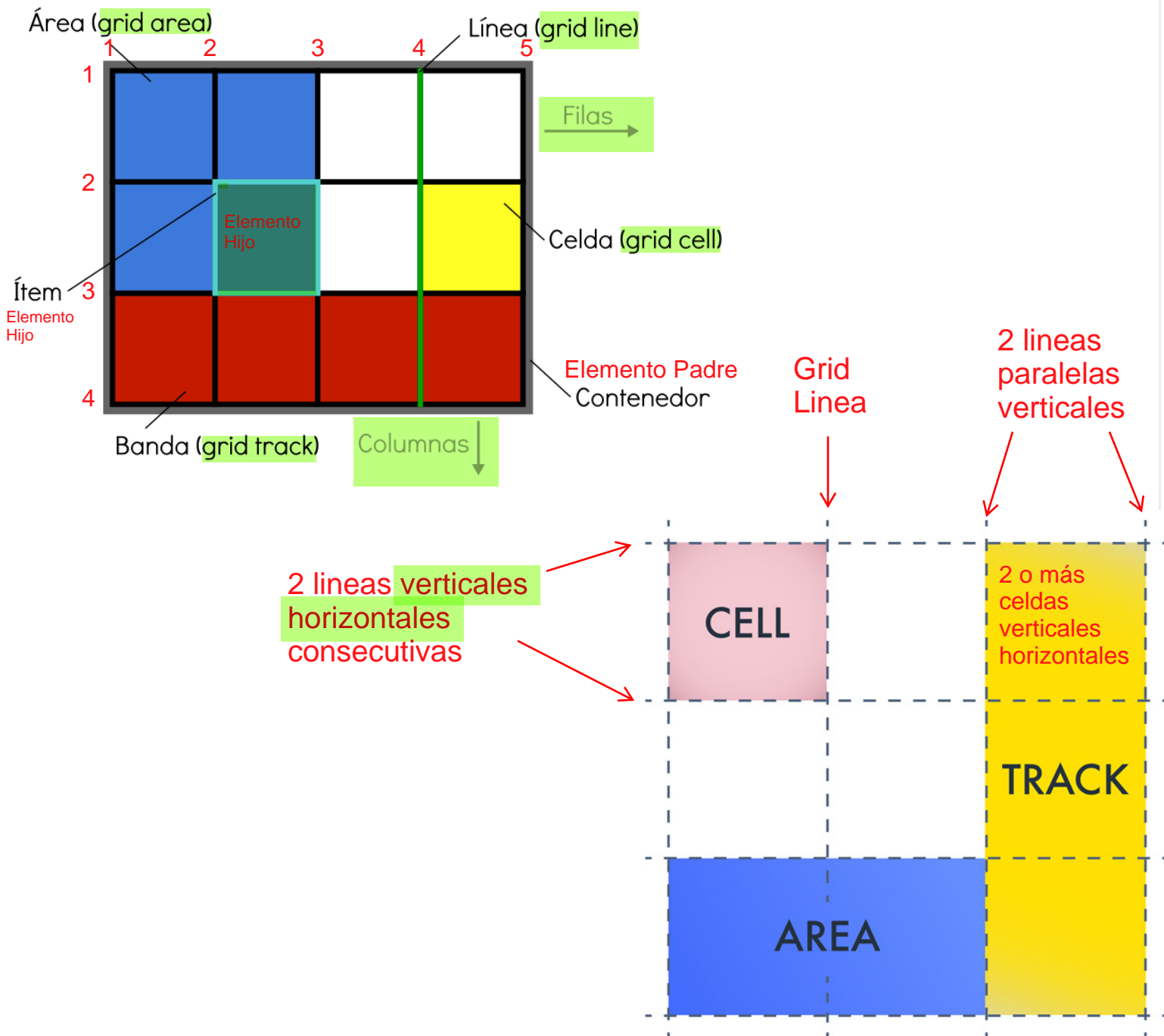
Grid cell o **celda de la rejilla**: Es el **espacio definido** por **dos líneas horizontales** y **dos verticales consecutivas**. Es decir, una **celda de la cuadrícula**.

Grid track o **zona de la rejilla**: Es el espacio entre **dos líneas paralelas, horizontales** o **verticales**. Es decir, son dos o más celdas horizontales o verticales juntas.

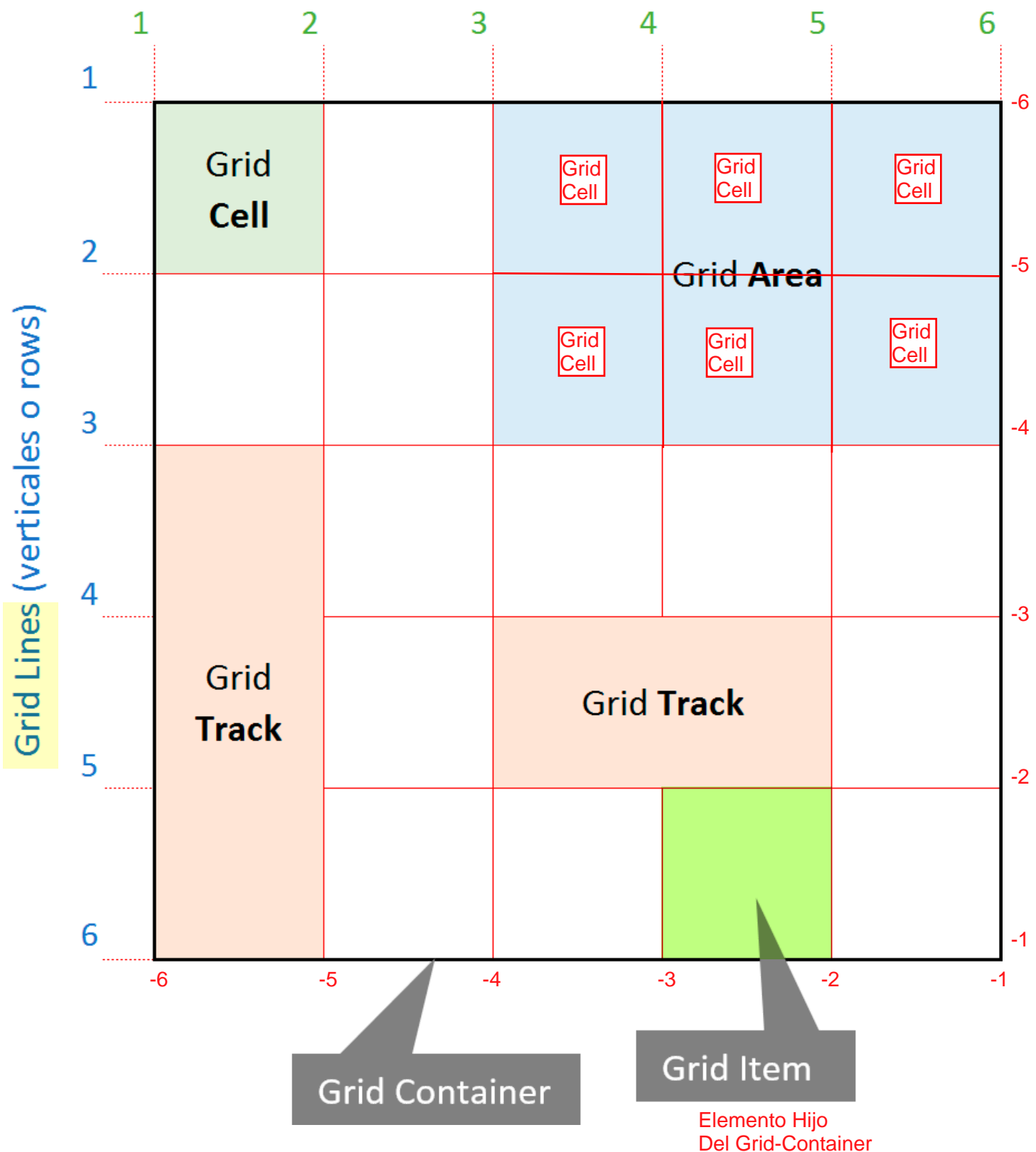
Grid area o **área de la rejilla**: Es una **figura** que cubre **cuatro** o **más celdas adyacentes, horizontales** y **verticales**. Es decir, lo que indica su propio nombre, un **área rectangular** o **cuadrada de la cuadrícula**.

Tanto las **áreas**, como a las **líneas**, pueden tener "**nombre**" además de referirnos a ellas por su número.

Para aclarar un poco más estos términos, la **figura** muestra todos ellos de forma gráfica (**grid container** con **25 grid items**).



Grid Lines (horizontales o columnms)



display: grid: al igual que ocurre con Flexbox, esta propiedad la tendrá el padre de todos los elementos que vayan a formar nuestra rejilla o grid.

grid-template-columns y **grid-template-rows**: se definen en ellas el tamaño y número de las columnas y las filas que van a formar parte de nuestra cuadrícula o grid. El tamaño puede ser expresado en px, % o en fr, una nueva unidad de medida que representa la fracción de espacio disponible dentro del grid.

Vemos que, por ejemplo, para crear una cuadrícula de 3x2 (3 filas y 2 columnas) solo tenemos que crear un contenedor y escribir este simple código css:

```
.container {
  display: grid;
  grid-template-rows: 1fr 1fr 1fr;
  grid-template-columns: 1fr 1fr;

  grid-gap: 4px;
  padding: 4px;
  outline: 1px solid #000;
}
.item {
  padding: 10px;
  outline: 1px solid #f00;
}
```

Un esquema es una línea que se dibuja alrededor de los elementos, FUERA de los bordes, para que el elemento 'se destaque'

Observación: lo que no está en negrita lo he puesto para que se visualice el contenedor y las celdas.

Y en el archivo .html, este otro código:

```
<div class="container">
  <div class="item a">A</div>
  <div class="item b">B</div>
  <div class="item c">C</div>
  <div class="item d">D</div>
  <div class="item e">E</div>
  <div class="item f">F</div>
</div>
```

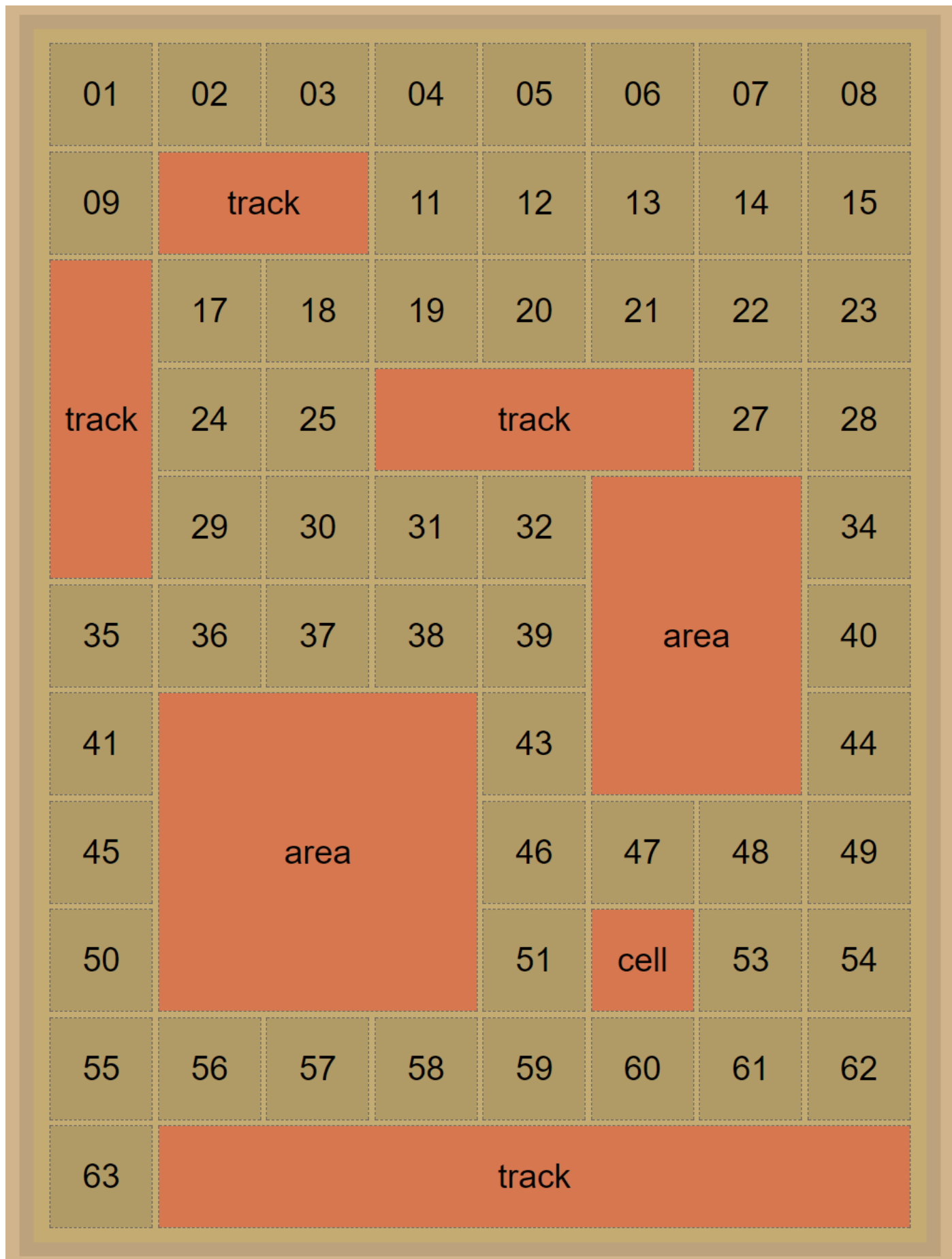
Observación: las clases y las letras las he puesto para que se vea algo dentro de las celdas.

El resultado, en el navegador, es:

Grid-Container
elemento padre

A Grid-Items - elemento hijo	B
C	D
E	F

De manera que, con CSS Grid Layout, podemos hacer cuadrículas todo lo complejas que queramos, sin apenas esfuerzo:



Siguiente » (</css/layout-responsive/grid-layout/18-propiedades-grid-container.html>)

 Layout

■ [Qué es el Layout \(/css/layout-responsive/17-qué-es-el-layout.html\)](/css/layout-responsive/17-qué-es-el-layout.html)

- -- Qué es Grid Layout (/css/layout-responsive/grid-layout/15-qué-es-el-css-grid-layout.html)
- Propiedades: Grid container (/css/layout-responsive/grid-layout/18-propiedades-grid-container.html)
- Propiedades: Grid Items (/css/layout-responsive/grid-layout/19-propiedades-grid-items.html)
- -- Qué es CSS FlexBox (/css/layout-responsive/flex-box/22-qué-es-css-flexbox.html)
- Propiedades: Flex container (/css/layout-responsive/flex-box/21-propiedades-flex-container.html)
- Propiedades: Flex Items (/css/layout-responsive/flex-box/20-propiedades-flex-items.html)
- -- Qué es Media query (/css/layout-responsive/media-queries/28-qué-es-una-media-querie.html)
- Ejemplos responsive (/css/layout-responsive/media-queries/29-ejemplos-responsive.html)

Recursos ...

- Visual Studio code (<https://code.visualstudio.com/>)
- Sublime Text (<https://www.sublimetext.com/>)
- CSS LINT (<http://csslint.net>)
- Normalize (<https://necolas.github.io/normalize.css/>)

