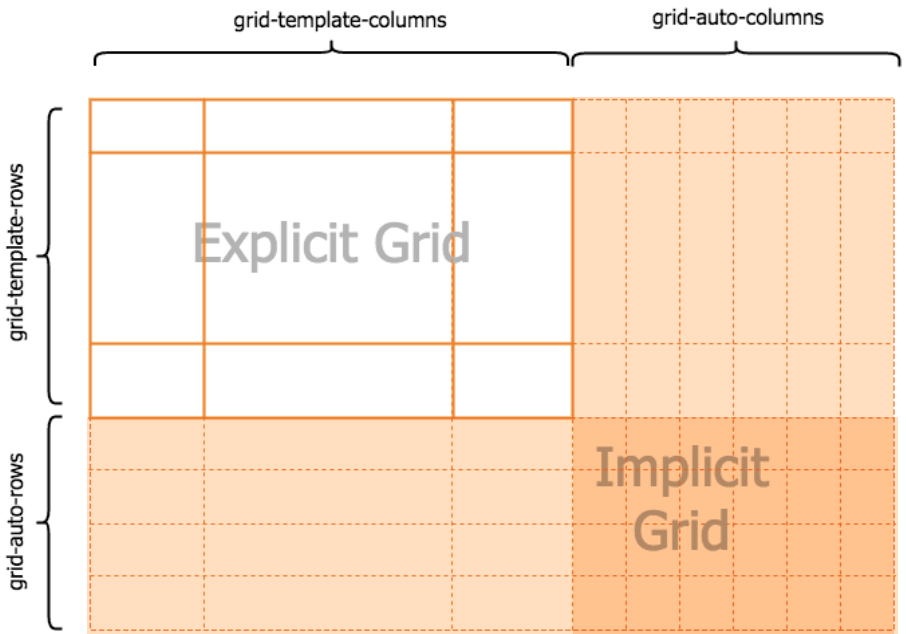
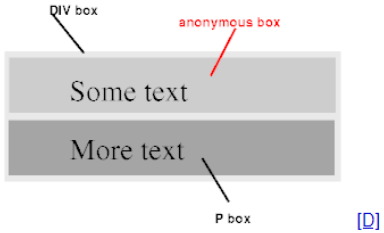
Grid-cell
Una celda de cuadrícula es la intersección de una fila de cuadrícula y una columna de cuadrícula.
Es la unidad más pequeña de la cuadrícula a la que se puede hacer referencia al colocar elementos de la cuadrícula.

Grid-Items
Los elementos 'Grid-items' de un 'Grid-Container' son cuadrados o cajas que representan su contenido en flujo.
Cada elemento secundario en el flujo de un contenedor de cuadrícula 'Grid-Container'
se convierte en un elemento de cuadrícula 'Grid-Item' ,
y cada secuencia contigua de 'text run' esta envuelto
en un bloque anónimo contenedor de bloque de grid items

Si toda la secuencia de texto secundario 'text run' solo contiene espacios en blanco
(Caracteres que pueden verse afectados por la propiedad de white-space)
En cambio, no se representa (como si se visualizaran sus nodos de texto: ninguno).



tween Explicit and In



[D]

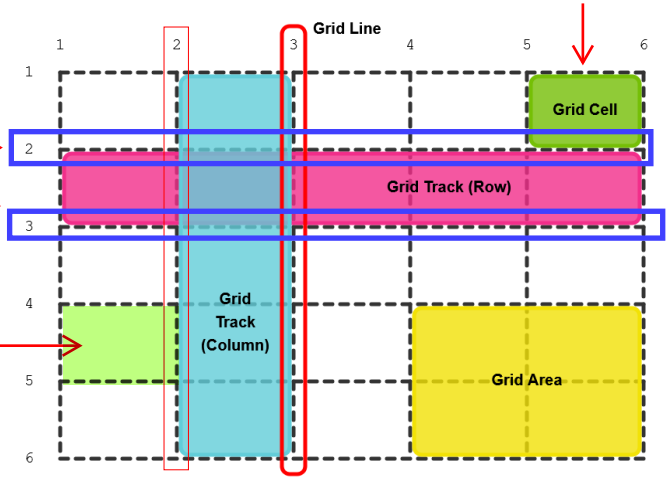
Diagram showing the three boxes, of which one is anonymous, for the example above.

finalmente nos da la habilidad de definir
Grid Layout (https://www.w3.org/TR/css-grid-1/) finally gives us the ability to define
cuadriculas en CSS y colocar los elementos dentro de 'Grid-Cells' Esto por sí solo es grandioso pero el hecho de que no
grids in CSS and place items into grid cells. This on its own is great, but the fact that we
tenemos que especificar cada track
don't have to specify each track (https://www.w3.org/TR/css-grid-1/#grid-track-
nosotros no tenemos que colocar todos los elementos manualmente creando un nuevo modulo incluso mejor
concept) and we *don't have to* place every item manually makes the new module even
son bastante flexibles adaptar sus elementos 'items'
better. Grids are flexible enough to adapt to their items.

Este es todo el manejo por el que que llamamos 'Explicito' y 'Implicito' GRID
This is all handled by the so called explicit and implicit grid.

GRID TRACK es un término genérico para una columna o fila de cuadrícula 'grid' ; en otras palabras, es el espacio entre dos líneas de cuadrícula adyacentes 'grid-lines'.
A cada 'GRID-TRACK' se le asigna una función de tamaño, el cual controla como de ancho o alto puede crecer la columna o fila, y por lo tanto como de separadas están sus 'GRID-LINES' delimitadoras. Las 'GRID-TRACK' adyacentes pueden estar separadas por 'GRID-GAP', pero por lo demás están compactadas estrechamente.

Unidad Simple Grid Container



Grid-Line

> ***Hey!*** All code samples in this post are accompanied by images in order to display grid lines and tracks. If you want to tinker with the code yourself, I recommend you download Firefox Nightly (https://www.mozilla.org/en-US/firefox/channel/desktop/) because it currently has the best DevTools (https://hacks.mozilla.org/2017/06/new-css-grid-layout-panel-in-firefox-nightly/) to debug grids.

GRID-ITEMS es TODO elemento DESCENDIENTE de GRID-CONTAINER

Asignamos un tamaño fijo a las filas y columnas

Explicit Grids

Nosotros podemos definir un fijado numero de filas y track 'bandas/franjas' que forma una cuadricula para usar las propiedades

We can define a fixed number of lines and tracks that form a grid by using the properties `grid-template-rows`, `grid-template-columns`, and `grid-template-areas`. This manually defined grid is called *the explicit grid*.

(https://codepen.io/matuzo/pen/OjMGMY)

An explicit grid with 4 vertical tracks (columns) and 2 horizontal tracks (rows). (View Pen (https://codepen.io/matuzo/pen/OjMGMY))

```
.grid {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr 1fr;
  grid-template-rows: 100px 100px;
  grid-gap: 20px;
}
```

css

genera - grid track horizontal

genera - grid track vertical

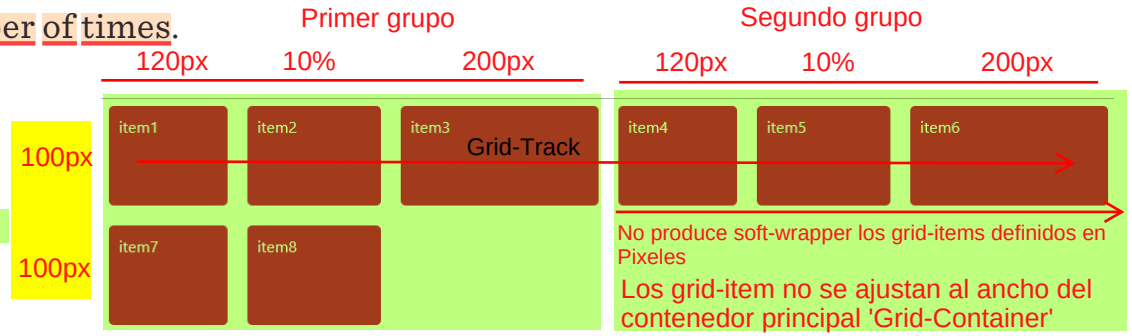
Repeating tracks

When we define `grid-template-columns: 1fr 1fr 1fr 1fr;` we get four vertical tracks bandas/franjas each with a width of `1fr`. We can automate that by using the `repeat()` notation like so `grid-template-columns: repeat(4, 1fr);`. The first argument specifies the number of repetitions, the second a track list, which is repeated that number of times.

A track *list*? Yes, you can actually repeat multiple tracks.

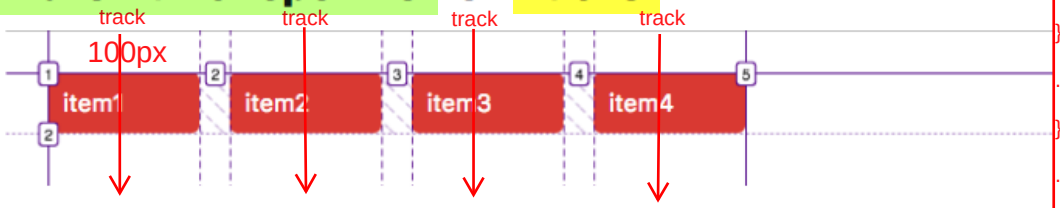
2 veces -> 120px 10% 200px
y si no hay suficiente espacio
pasa a la siguiente fila

```
.grid {
  display: grid;
  grid-template-columns: repeat(2, 120px 10% 200px);
  grid-template-rows: 100px 100px;
  grid-gap: 20px;
}
```



Bandas

Automatic repetition of tracks



An explicit grid with 4 vertical tracks each 100px wide, generated by the repeat notation. (View Pen)

```
.grid {
  display: grid;
  grid-gap: 20px;
}

.grid1 {
  grid-template-columns: repeat(4, 100px);
}

.grid2 {
  display: none;
}
```

franjas o bandas

Automatic repetition of tracks

(https://codepen.io/matuzo/pen/VzeNpE)

An explicit grid with 4 vertical tracks each 100px wide, generated by the repeat notation. (View Pen (https://codepen.io/m

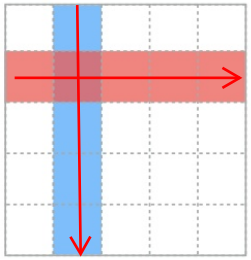
La notacion de repeticion es bastante manejable pero puede ser incluso más automatizada

The repeat notation is quite useful, but it can be automated even further. Instead of setting a fixed number of repetitions we can use the `auto-fill` and `auto-fit` keywords.

más. En lugar de ajustando un fijado numero de repeticiones nosotros podemos usar el y

Grid *track*

The space between two or more adjacent grid ***lines****.*
Row tracks are horizontal,
Column tracks vertical.



Auto-filling tracks

Ajusta la cantidad de grid-items al ancho total del Grid-Container

La palabra clave de auto-fill crea tantas track como caben en el contenedor de cuadrícula sin causar el 'grid' de desbordamiento ella.

The `auto-fill` keyword creates as many tracks as fit into the grid container without causing the grid to overflow it.

(https://codepen.io/matuzo/pen/PKZgmr)

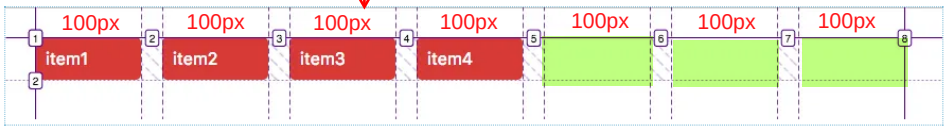
Repetition of as many vertical tracks with a width of 100px as fit into the grid container. (View Pen (https://codepen.io/matuzo/pen/PKZgmr))

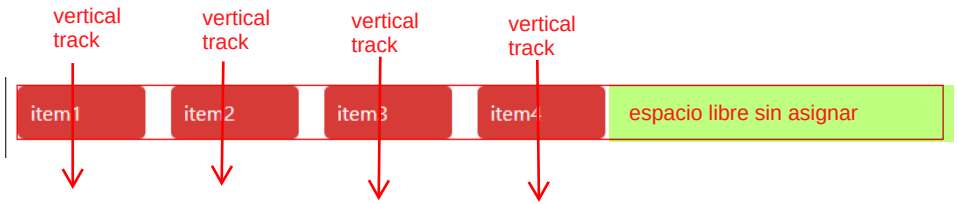
```
.grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, 100px);
  grid-gap: 20px;
}
```

css

crea tanto grid-track verticales como haya definidos

GRID-CONTAINER





repeat() dentro de column solo creara una 'grid-track' con eje horizontal que atravesara toda la fila y ademas contendra todas las columnas separadas

Fijate que

Note that `repeat(auto-fill, 1fr);` will only create one track because a single track with a width of `1fr` already fills the whole grid container.

solo creara una 'track' porque una simple track

con un ancho de 1fr ya llena todo el contenedor del Grid-Container

Solo hay una fila con 4 elementos 'grid-items' DIVs para ubicar los elementos

Auto-fitting tracks

ESTIRA LOS GRID-ITEMS HASTA LLEGAR AL FINAL DEL GRID-CONTAINER

comportamiento es el mismo camino como

excepto que despues

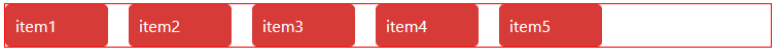
The `auto-fit` keyword behaves the same way as `auto-fill`, except that after grid item placement it will only create as many tracks as needed and any empty repeated track collapses.

coloca ellos solo crear tantas filas como necesitamos y 'track' vacias repetidas colapsar grid-track

La palabra clave auto-fit se comporta de la misma manera como auto-fill excepto que después de la colocación grid-item que sólo va a crear tantas track como necesite y algunas vacias track colapsa.

```css
.grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, 100px);
  grid-gap: 20px;
}
```

Espacio sin colapsar



Espacio colapsado

In the example used in this section the grid will look the same with `repeat(auto-fit, 100px);` and `repeat(4, 100px);`. The difference is visible when there are more than 4 grid items.

estira esta track desde la linea 5 hasta la 12

Se han estirado los elementos

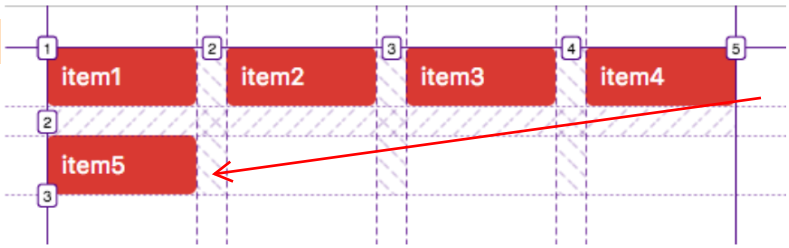
Si hay mas elementos

If there are more items, `auto-fit` creates more columns.

(https://codepen.io/matuzo/pen/RZrOXw)

la notacion repetir con la palabra clave auto-fit crea tantas grid-track como necesites y tantos fit como quepa el grid container

The repeat notation with the auto-fit keyword creates as many tracks as needed and as many as fit into the grid container. (https://codepen.io/matuzo/pen/RZrOXw))



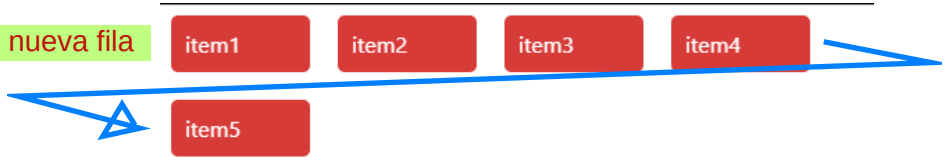
Por otro lado

Si un ajustado numero de 'track' verticales es usado en la notacion de repeticion y el numero de 'elementos' excede este valor más filas son añadidas

On the other hand, if a fixed number of vertical tracks is used in the repeat notation and the number of items exceeds this value more rows are added. You can read more about that in the next section: implicit grids (#implicit-grids) .

```
.grid {
  display: grid;
  grid-template-columns: repeat(4, 100px);
  grid-gap: 20px;
}
```

Crea una nueva fila



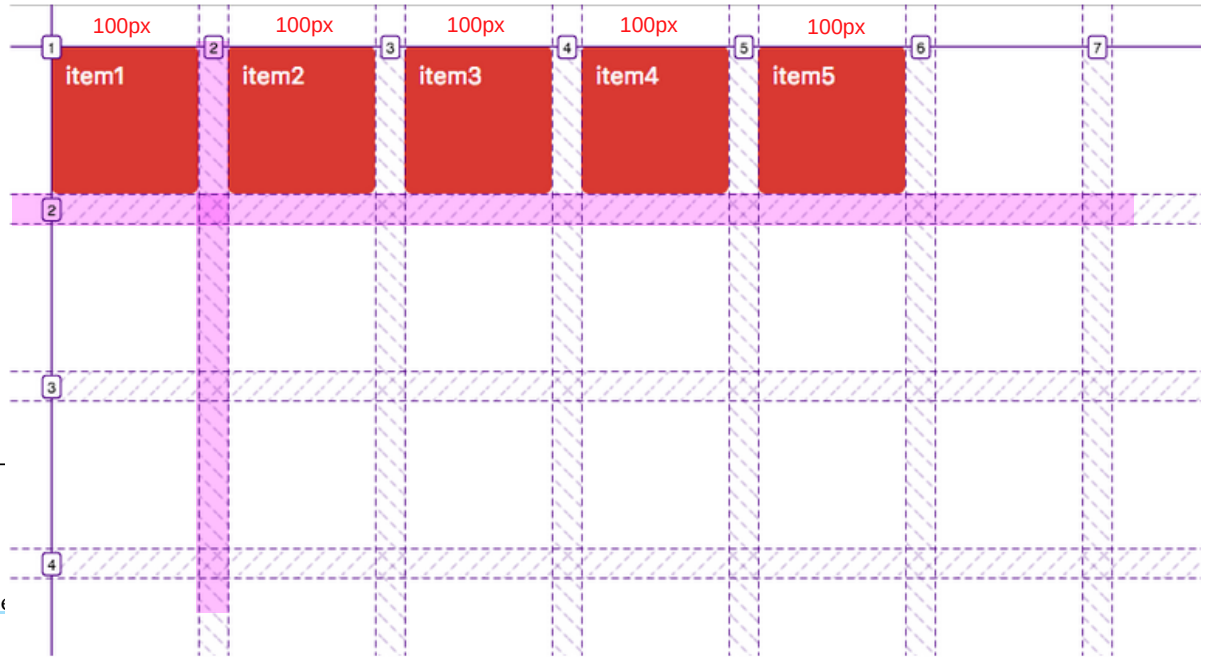
(https://codepen.io/matuzo/pen/ZJQZgr)

If there are more items than vertical tracks, more rows are added. (View Pen (https://codepen.io/matuzo/pen/ZJQZgr))

I've used `grid-template-columns` in the above examples out of convenience, but all rules also apply to `grid-template-rows` .

```
.grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, 100px);
  grid-template-rows: repeat(auto-fill, 100px);
  grid-gap: 20px;
  height: 100%;
}

html, body {
  height: 100%;
}
```

```
<section class="grid">
  <div class="item">item1</div>
  <div class="item">item2</div>
  <div class="item">item3</div>
  <div class="item">item4</div>
  <div class="item">item5</div>
</section>
```



(https://codepen.io/matuzo/pen/prgVmW)

The repeat notation with the auto-fill keyword on both axes. (View Pen (https://cod

Todas las filas y columas autogeneradas fuera del GRID-CONTAINER definido mediante grid-template-rows/columns

Grid-Items son elementos como DIVS del Grid-Container

```
<div class='container'>
  <div class='item'></div>
  <div class='item'></div>
</div>
```

→ GRID-ITEMs

Implicit Grids

Si hay mas 'grid items' que 'grid-cell' en el grid container o cuando 'grid-item' esta colocado fuera del

If there are more grid items than cells in the grid or when a grid item is placed outside of the explicit grid, the grid container automatically generates grid tracks by adding grid lines to the grid. The explicit grid together with these additional implicit tracks and lines forms the so called *implicit grid*.

grid explicito

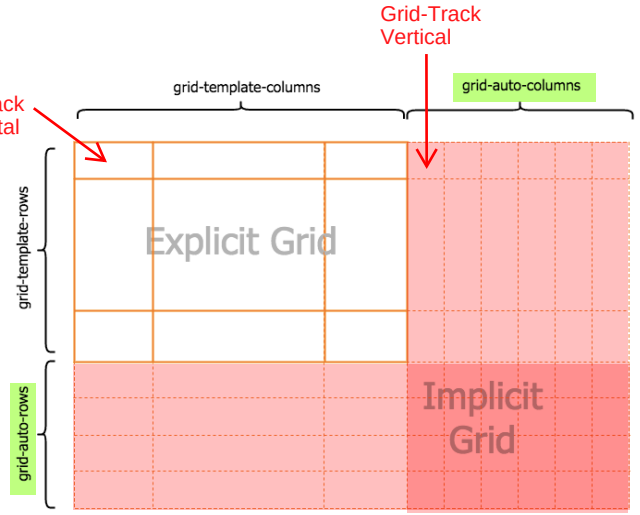
junto con esas implicitas bandas adicionales

formas de lineas

Grid Implicito es una ampliacion del GRID-CONTAINER fuera del contenedor principal

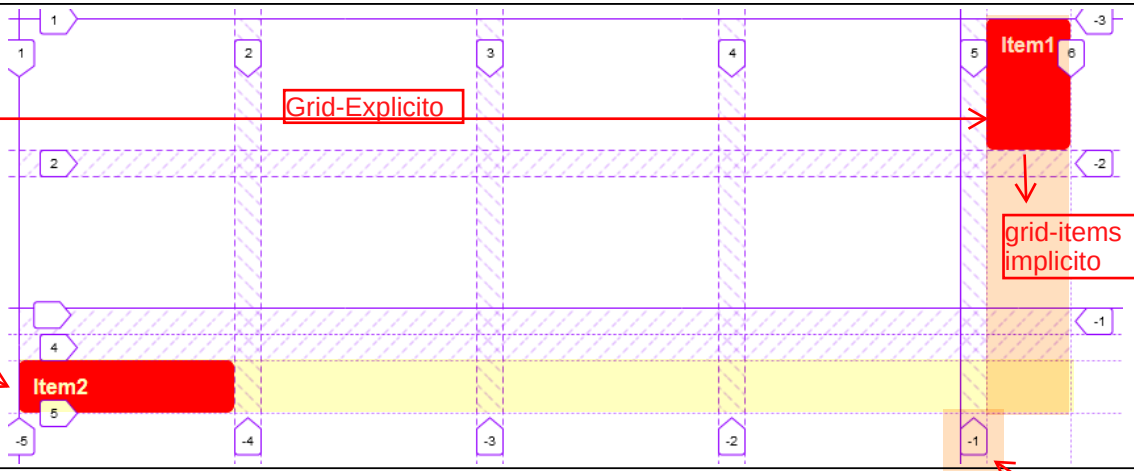
(https://codepen.io/matuzo/pen/vJLwNY)

Implicit Grid - Todos los 'grid-items' que estan fuera del 'Grid-Container' que esta colocado fuera del 'Grid-Explicito' y que generan nuevas 'Grid-Track' y 'Grid-lines'



Two items placed outside of the explicit grid causing the creation of implicit lines and tracks (View Pen (https://codepen.io/matuzo/pen/vJLwNY))

```
.item:first-child {
  grid-column-start: -1;
}

.item:nth-child(2) {
  grid-row-start: 4;
}
```



css

```
<section class="grid">
  <div class="item">item1</div>
  <div class="item">item2</div>
</section>

.grid {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-template-rows: 100px 100px;
  grid-gap: 20px;
  max-width: 800px;
  border: 1px solid #f00;
}
```

Los anchos y altos de los track implicitos esta establecidos o fijados automaticamente.

Ellos son solo suficiente grande como para ajustarse a los elementos de 'grid-items'

The widths and heights of the implicit tracks are set automatically. They are only big enough to fit the placed grid items, but it's possible to change this default behavior.

grid-column-start: -1;

Grid Track Vertical

Grid Track Horizontal

Sizing implicit tracks

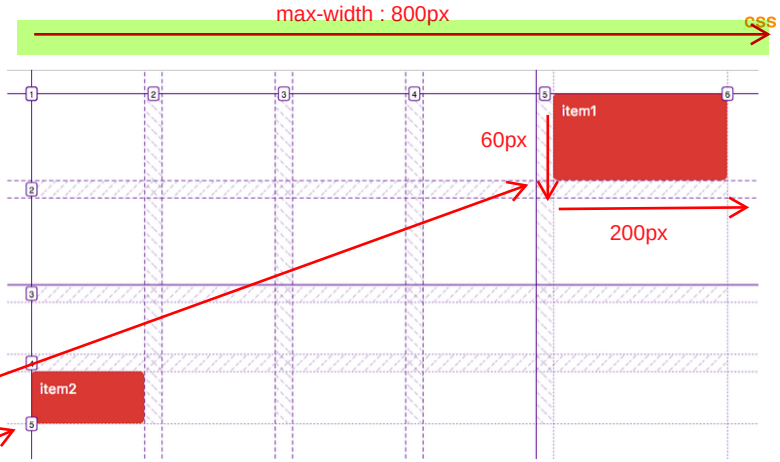
propiedad nos da el control final el tamaño de implicitas 'track'

The grid-auto-rows and grid-auto-columns properties give us control over the size of implicit tracks.

grid-auto-columns: General el tamaño de la columna automaticamente
grid-auto-rows: Genera el tamaño de la fila automaticamente

```css
.grid {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-template-rows: 100px 100px;
  grid-gap: 20px;
  grid-auto-columns: 200px;
  grid-auto-rows: 60px;
}
```

```
.grid {
 display: grid;
 grid-template-columns: repeat(4, 1fr);
 grid-template-rows: 100px 100px;
 grid-gap: 20px;

 grid-auto-columns: 200px;
 grid-auto-rows: 60px;
 max-width: 800px;
 border: 1px solid #f00;

 /* Or use minmax for more flexibility:

  grid-auto-columns: minmax(200px, auto);
  grid-auto-rows: minmax(60px, auto);

 */
}

.item:first-child {
 grid-column-start: -1;
}

.item:nth-child(2) {
 grid-row-start: 4;
}
```



Implicit tracks will now always have a width of `200px` and a height of `60px`, no matter if the grid item fits or not.

si el elemento de la cuadrícula encaja o no.

no importa

(https://codepen.io/matuzo/pen/KvVLVy)

Fixed widths and heights for implicit tracks (View in CodePen (https://codepen.io/matuzo/pen/KvVLVy))

You can make sized implicit tracks more flexible by specifying a range using the `minmax()` notation.

```css
.grid {
  grid-auto-columns: minmax(200px, auto);
  grid-auto-rows: minmax(60px, auto);
}
```

pero expandira si el contenido lo demanda

Implicit tracks are now at least 200px wide and 60px high, but will expand if the content demands it.

Extendiendo el grid al comienzo

Extending the grid to the start

Banda Implicita no sólo pueden ser agregados al final del grid explicito

Quizas tambien pasa

Implicit tracks may not just be added to the end of the explicit grid. It may also happen

que el grid explicito necesita ser extendido al comiezo

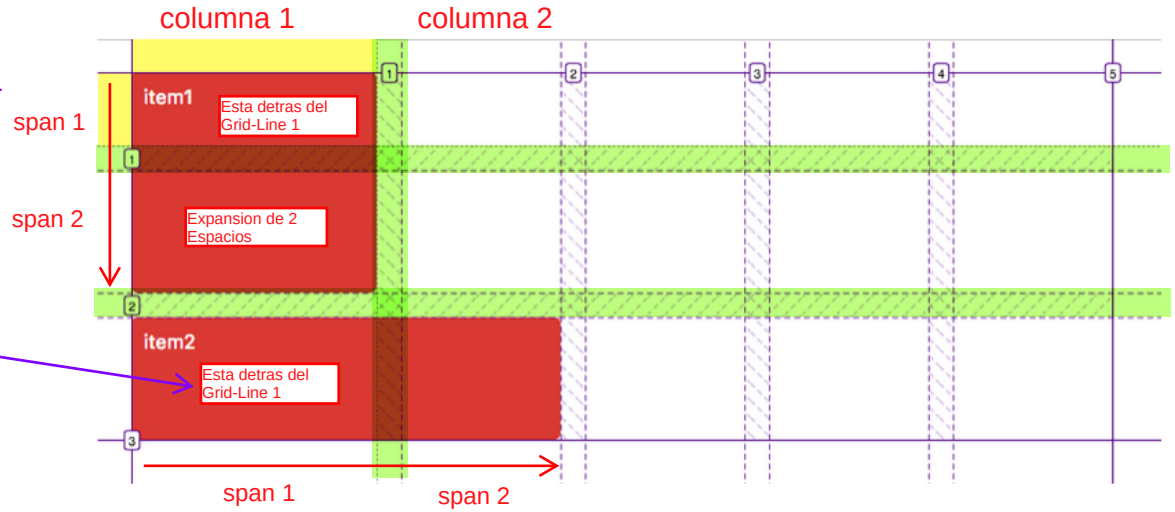
that the explicit grid needs to be extended to the start.

(https://codepen.io/matuzo/pen/BdyJWR)

An implicit grid extended by one row and one column to the start (View Pen (https://codepen.io/matuzo/pen/BdyJWR))

```
display: grid;
grid-template-columns: repeat(4, 1fr);
grid-template-rows: 100px 100px;
grid-auto-columns: 200px;
grid-auto-rows: 60px;
grid-gap: 20px;
max-width: 800px;
}
```

```
.item:first-child {
  grid-row-end: 2;
  grid-row-start: span 2;
}

.item:nth-child(2) {
  grid-column-end: 2;
  grid-column-start: span 2;
}
```



Cada elemento termina sobre la 2º linea y expande 2 celdas una vertical y la otra horizontal

Each item ends on the second line and spans 2 cells (one vertically, the other horizontally). Since there is only one cell before the second line respectively another implicit track is added to the grid at the start of each side.

desde hay solo una celda antes la segunda linea respectivamente otra

implicita franja es añadida al grid al comienzo de cada lado

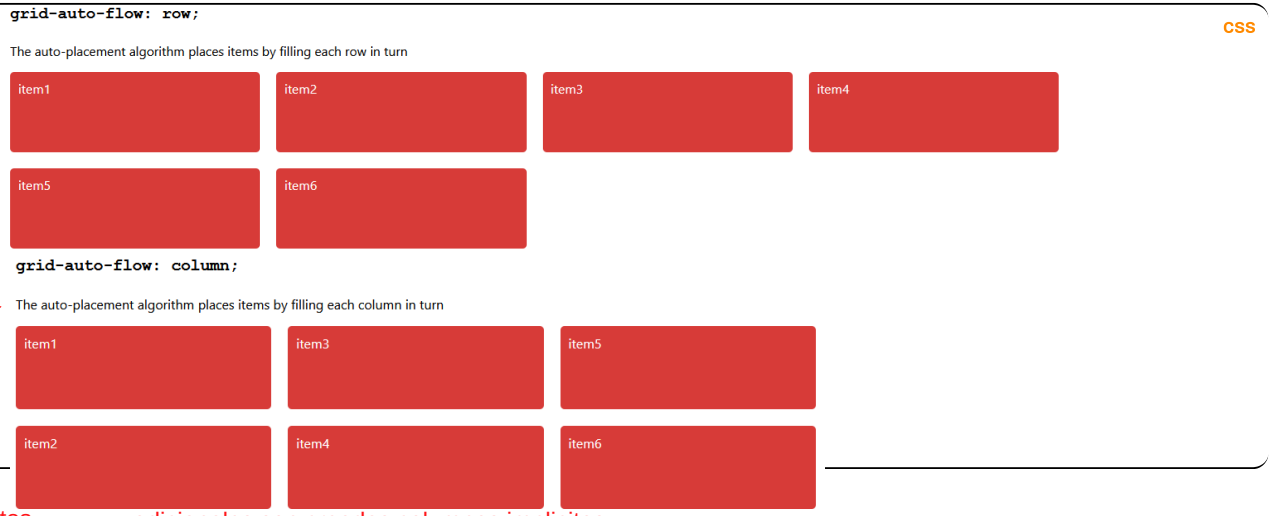
Automatic Placement

As already mentioned, implicit tracks are also added if the number of items exceeds the number of cells. By default, the auto-placement algorithm places items by filling each row consecutively, adding new rows as necessary. We can specify how auto-placed items get flowed into the grid by using the `grid-auto-flow` property.

(https://codepen.io/matuzo/pen/JyGqBP)

Instead of rows, new columns are added if the number of items exceeds the number of cells (View in CodePen (https://codepen.io/matuzo/pen/JyGqBP))

```
.grid {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  grid-template-rows: 100px 100px;
  grid-gap: 20px;
  grid-auto-flow: column;
}
```



Al contrario de filas , columnas estan comenzando llenando con elementos adicionales son creadas columnas implicitas

Instead of rows, columns are being filled with items and additional implicit columns are created.

Not defining an explicit grid

Debido al hecho de que es posible dimensionar automaticamente las celdas usando

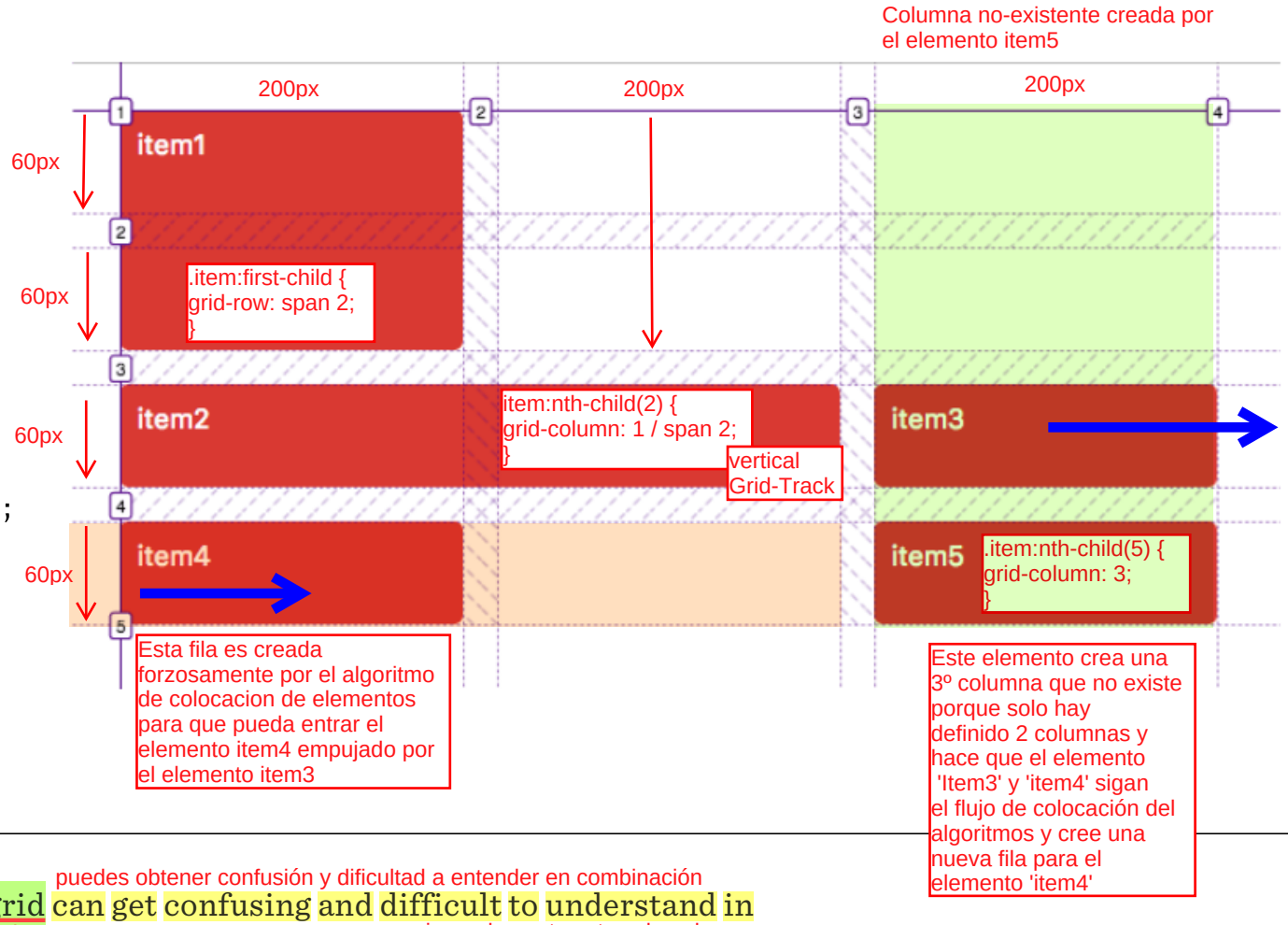
Due to the fact that it's possible to automatically size cells using `grid-auto-rows` and `grid-auto-columns` it's not obligatory to define an explicit grid.

no es obligatorio definir un grid explicito

(https://codepen.io/matuzo/pen/gxPzeY)

An implicit grid without explicit lines and tracks (View in CodePen (https://codepen.io/matuzo/pen/gxPzeY))

```css
.grid {
  display: grid;
  grid-auto-columns: minmax(60px, 200px);
  grid-auto-rows: 60px;
  grid-gap: 20px;
}

.item:first-child {
  grid-row: span 2;
}

.item:nth-child(2) {
  grid-column: 1 / span 2;
}

.item:nth-child(5) {
  grid-column: 3;
}
```

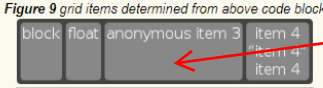


Confiando únicamente en el 'Grid-implicito' puedes obtener confusión y dificultad a entender en combinación con el explicito posicionamiento primer elemento esta colocado

Relying solely on the implicit grid can get confusing and difficult to understand in combination with explicit placement. In this example the first item is placed `auto` and spans 2 rows, the second item is placed explicitly in the first column and spans 2

columns creating a second vertical track. The third and fourth item would actually both be placed automatically in the fourth row, but the fifth item is placed explicitly in the previously non-existent third column. This creates a third vertical track and due to debido Grids auto-placement, the third item moves up a row to fill the space.

Figure 9 grid items determined from above code block



Examples of grid items:

Los GRID-ITEMS son todos los elementos HIJOS del GRID-CONTAINER

```
<div style="display: grid">

    <!-- grid item: block child -->
    <div id="item1">block</div>

    <!-- grid item: floated element; floating is ignored -->
    <div id="item2" style="float: left;">float</div>

    <!-- grid item: anonymous block box around inline content -->
    anonymous item 3

    <!-- grid item: inline child -->
    <span>
        item 4
        <!-- grid items do not split around blocks -->
        <q style="display: block" id=not-an-item>item 4</q>
        item 4
    </span>
</div>
```

Conclusion

e-

This article doesn't cover everything there is to know about the explicit and implicit grid, but it should give you more than a solid understanding of the concept. Knowing why and how implicit lines and tracks a created is vital for working with Grid Layout.

You can find all the examples used in this article in a Collection on CodePen (https://codepen.io/collection/XkLzYO/) .

If you want to learn more about Grids check out The Complete Guide to Grid (https://css-tricks.com/snippets/css/complete-guide-grid/) , Getting Started with CSS Grid (https://css-tricks.com/getting-started-css-grid/) , Grid By Example (https://gridbyexample.com/) and A Collection of Interesting Facts about CSS Grid Layout (https://css-tricks.com/collection-interesting-facts-css-grid-layout/) .