

Cuadrícula CSS

La cuadrícula CSS es un estándar más nuevo que facilita la creación de diseños receptivos complejos. Funciona convirtiendo un elemento HTML en una cuadrícula y le permite colocar elementos secundarios en cualquier lugar dentro.

En este curso, aprenderá los fundamentos de la cuadrícula CSS mediante la creación de diferentes diseños complejos, incluido un blog.

1. Crea tu primera cuadrícula CSS
2. Añadir columnas con grid-template-columns
3. Agregar filas con grid-template-rows
4. Use unidades de cuadrícula CSS para cambiar el tamaño de columnas y filas
5. Crear un espacio entre columnas usando grid-column-gap
6. Crear un espacio entre filas usando grid-row-gap
7. Agregue brechas más rápido con grid-gap
8. Use grid-column para controlar el espaciado
9. Use grid-row para controlar el espaciado
10. Alinee un elemento horizontalmente usando la justificación propia
11. Alinear un elemento verticalmente usando align-self
12. Alinear todos los elementos horizontalmente usando justificar elementos
13. Alinear todos los elementos verticalmente usando elementos de alineación
14. Divide la cuadrícula en una plantilla de área
15. Coloque elementos en áreas de cuadrícula usando la propiedad grid-area
16. Usar grid-area sin crear una plantilla de áreas
17. Reduzca la repetición utilizando la función de repetición
18. Limite el tamaño de los elementos con la función minmax
19. Cree diseños flexibles usando autocompletar

20. Cree diseños flexibles mediante el ajuste automático
21. Use consultas de medios para crear diseños receptivos
22. No pasóCrear cuadrículas dentro de cuadrículas

1. Crea tu primera cuadrícula CSS

Convierta cualquier elemento HTML en un contenedor de cuadrícula configurando su `display` propiedad en `grid`. Esto le da la posibilidad de utilizar todas las demás propiedades asociadas con CSS Grid.

Nota: En CSS Grid, el elemento principal se denomina contenedor y sus elementos secundarios se denominan elementos.

Cambie la visualización del div con la containerclase a grid.

```

1  /* */
2  .div{background:lightblue}
3  .div1{background:#ff8c00}
4  .div2{background:#f00}
5  .div3{background:#ff69b4}
6  .div4{background:#ff00ff}
7
8  .container {
9    font-size: 40px;
10   width: 100%;
11   background: lightgray;
12   /* Only change code below this line */
13
14
15   /* Only change code above this line */
16 }
17
18
19 </style>
20
21 <div class="container">
22   <div class="div1">1</div>
23   <div class="div2">2</div>
24   <div class="div3">3</div>
25   <div class="div4">4</div>
26   <div class="div1">5</div>
27 </div>
28
29 /**
30 * Su salida de prueba irá aquí
31 */

```

Crea tu primera cuadrícula CSS

Convierte cualquier elemento HTML en un contenedor de cuadrícula configurando su `display` propiedad en `grid`. Esto te da la posibilidad de utilizar todas las demás propiedades asociadas con CSS Grid.

Nota: En CSS Grid, el elemento principal se denomina contenedor y sus elementos secundarios se denominan elementos.

Cambia la visualización del div con la clase `container` a grid.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

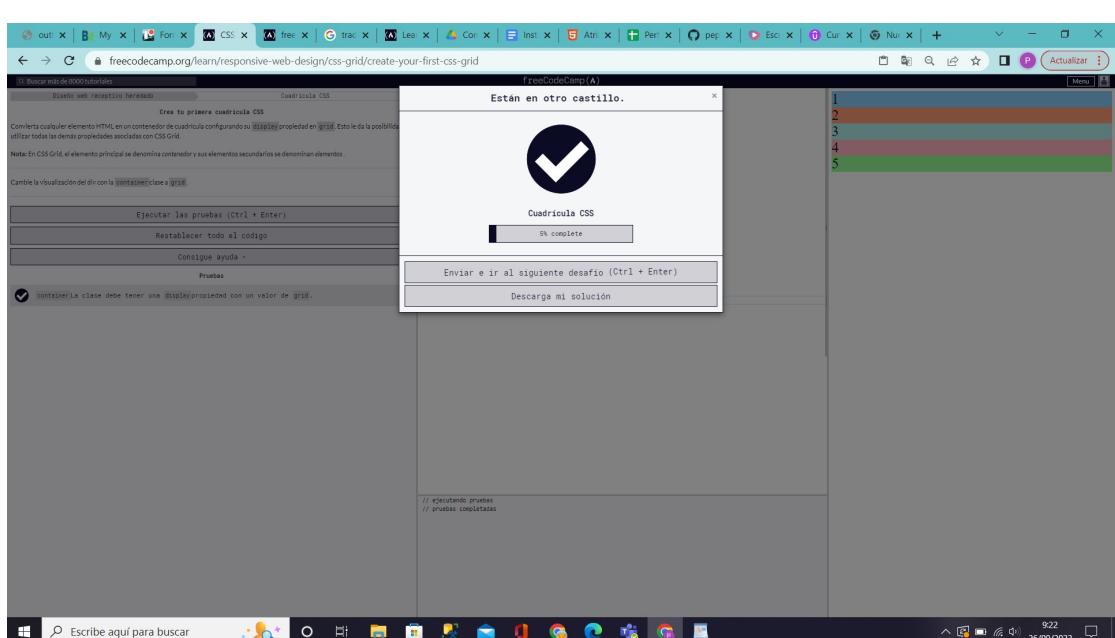
Consigue ayuda -

Pruebas

containerLa clase debe tener una `display`propiedad con un valor de `grid`.

```
1: <style>
2: .d1{background:LightSkyBlue;}
3: .d2{background:LightSalmon;}
4: .d3{background:PaleTurquoise;}
5: .d4{background:LightPink;}
6: .d5{background:PaleGreen;}
7: 
8: <div class="container">
9:   <div class="d1">1</div>
10:  <div class="d2">2</div>
11:  <div class="d3">3</div>
12:  <div class="d4">4</div>
13:  <div class="d5">5</div>
14: </div>
15: /* Only change code above this line */
16: 
17: </style>
18: 
19: <div class="container">
20:   <div class="d1">1</div>
21:   <div class="d2">2</div>
22:   <div class="d3">3</div>
23:   <div class="d4">4</div>
24:   <div class="d5">5</div>
25: </div>
26: /* Salida de prueba irá aquí
27: */

/*
```



```
<style>

.d1{background:LightSkyBlue;}

.d2{background:LightSalmon;}

.d3{background:PaleTurquoise;}

.d4{background:LightPink;}

.d5{background:PaleGreen;}
```

```

.container {
    font-size: 40px;
    width: 100%;
    background: LightGray;
    /* Only change code below this line */

display: grid;

    /* Only change code above this line */
}

</style>

<div class="container">
    <div class="d1">1</div>
    <div class="d2">2</div>
    <div class="d3">3</div>
    <div class="d4">4</div>
    <div class="d5">5</div>
</div>

```

2. Añadir columnas con grid-template-columns

La simple creación de un elemento de cuadrícula no te lleva muy lejos. También debe definir la estructura de la cuadrícula. Para agregar algunas columnas a la cuadrícula, use la `grid-template-columns`propiedad en un contenedor de cuadrícula como se muestra a continuación:

```

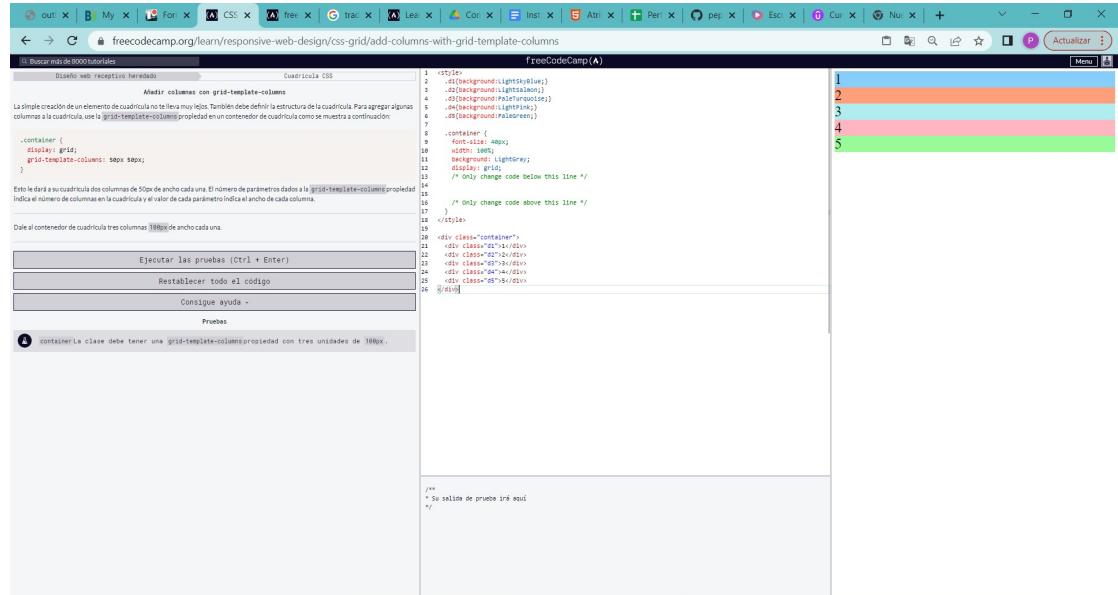
.container {
    display: grid;
    grid-template-columns: 50px 50px;
}

```

Esto le dará a su cuadrícula dos columnas de 50px de ancho cada una. El número de parámetros dados a la `grid-template-columns`propiedad indica el número de columnas en la

cuadrícula y el valor de cada parámetro indica el ancho de cada columna.

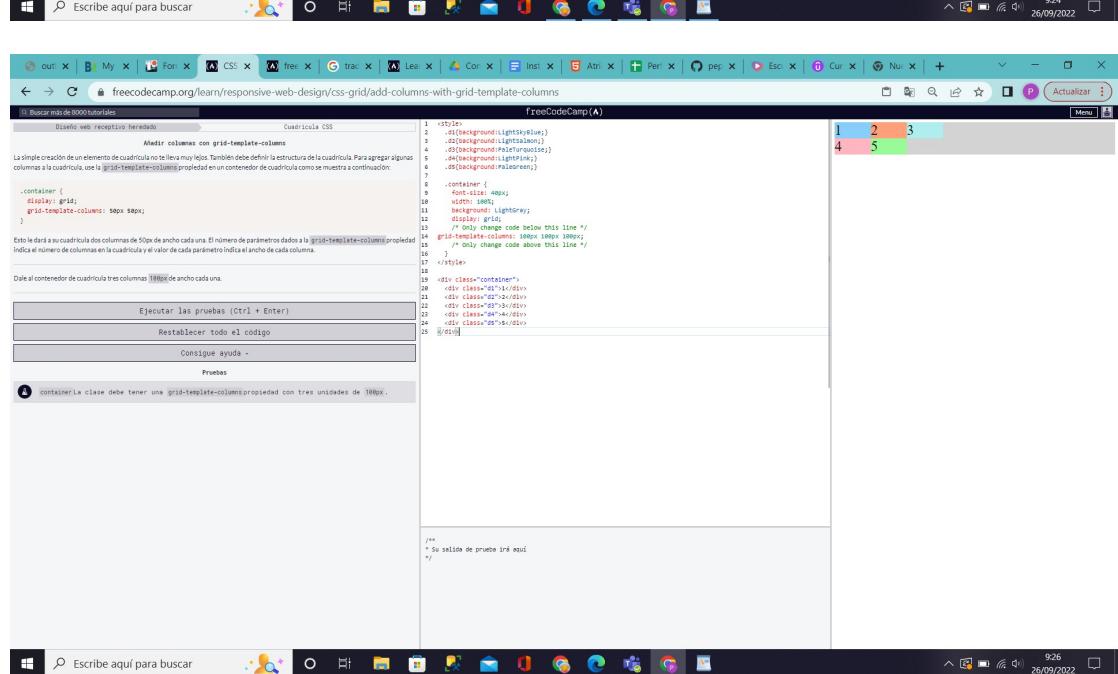
Dale al contenedor de cuadrícula tres columnas 100px de ancho cada una.



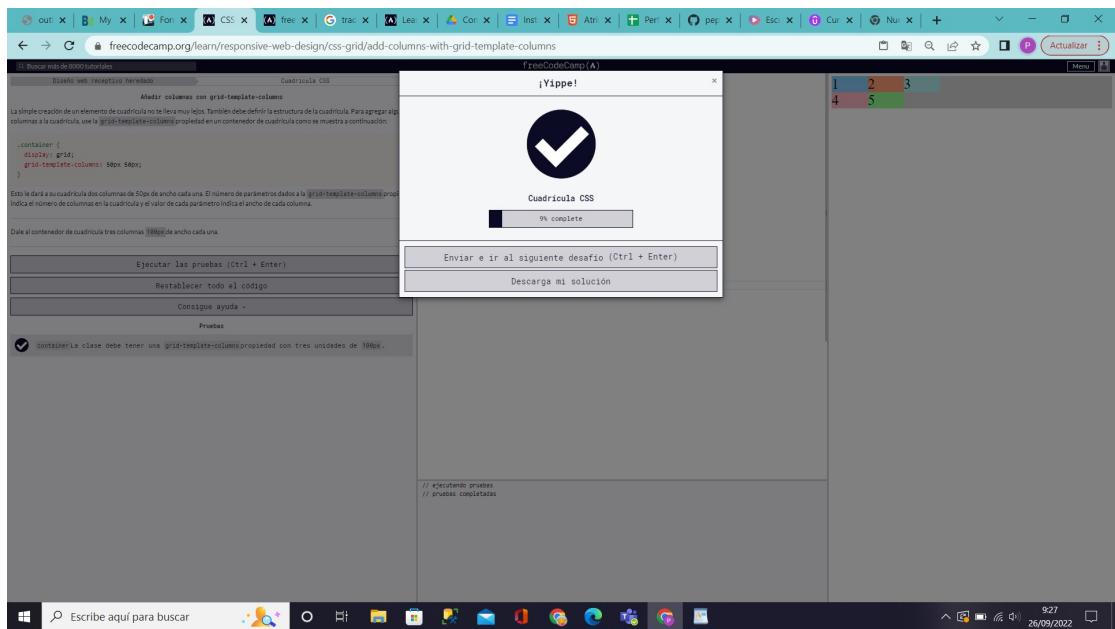
The screenshot shows a browser window with the URL freecodecamp.org/learn/responsive-web-design/css-grid/add-columns-with-grid-template-columns. The page title is "Cuadrícula CSS". The code editor contains the following CSS:

```
1 .container {  
2   display: grid;  
3   grid-template-columns: 50px 50px;  
4 }  
5  
6 .container {  
7   background-color: #f0f0f0;  
8   width: 100%;  
9   height: 100px;  
10  /* Only change code below this line */  
11 }  
12  
13 /* Only change code above this line */  
14 }  
15  
16 .container {  
17   display: grid;  
18   grid-template-columns: 1fr 1fr 1fr;  
19 }  
20  
21 <div class="container">  
22   <div class="c1">1</div>  
23   <div class="c2">2</div>  
24   <div class="c3">3</div>  
25   <div class="c4">4</div>  
26   <div class="c5">5</div>  
27 </div>  
28  
29 /* Su salida de prueba irá aquí  
30 */
```

The browser's developer tools show the rendered grid with five colored boxes labeled 1 through 5. Box 1 is blue, Box 2 is orange, Box 3 is teal, Box 4 is pink, and Box 5 is green. The browser status bar at the bottom right shows the date as 26/09/2022 and the time as 9:24.



This is a second screenshot of the same browser window, showing the same code and rendered grid. The browser status bar at the bottom right shows the date as 26/09/2022 and the time as 9:26.



```
<style>
```

```
.d1{background:LightSkyBlue;}
```

```
.d2{background:LightSalmon;}
```

```
.d3{background:PaleTurquoise;}
```

```
.d4{background:LightPink;}
```

```
.d5{background:PaleGreen;}
```

```
.container {
```

```
    font-size: 40px;
```

```
    width: 100%;
```

```
    background: LightGray;
```

```
    display: grid;
```

```
/* Only change code below this line */
```

grid-template-columns: 100px 100px 100px;

```
/* Only change code above this line */
```

```
}
```

```
</style>
```

```
<div class="container">
```

```

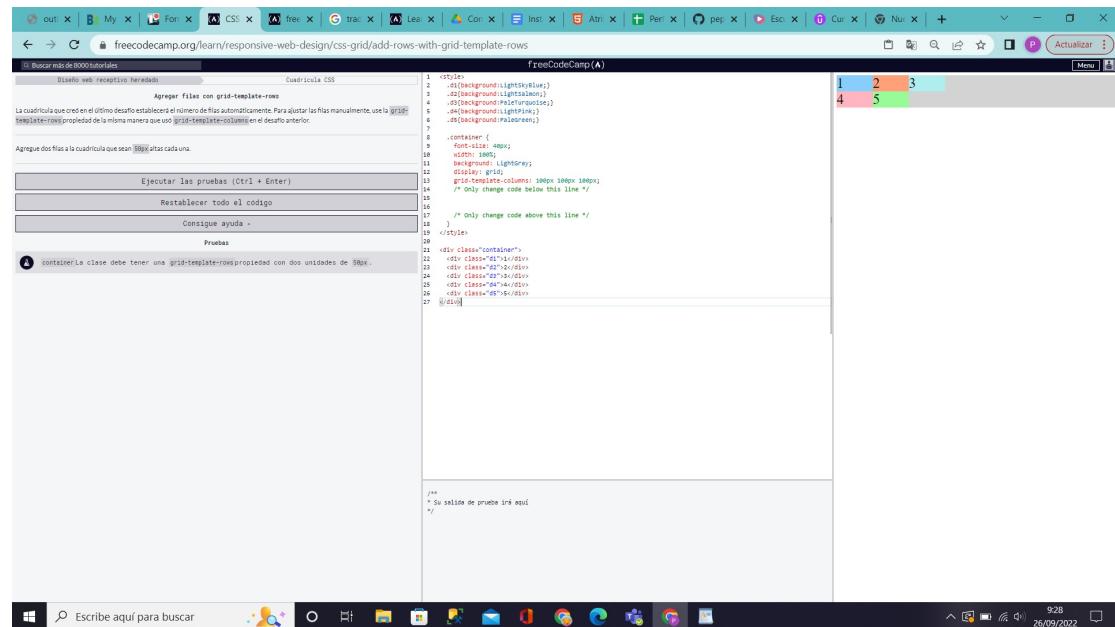
<div class="d1">1</div>
<div class="d2">2</div>
<div class="d3">3</div>
<div class="d4">4</div>
<div class="d5">5</div>
</div>

```

3. Agregar filas con grid-template-rows

La cuadrícula que creó en el último desafío establecerá el número de filas automáticamente. Para ajustar las filas manualmente, use la `grid-template-rows` propiedad de la misma manera que usó `grid-template-columns` en el desafío anterior.

Agregue **dos filas** a la cuadrícula que sean **50px** altas cada una.



The screenshot shows a browser window with a code editor for a CSS grid. The CSS code is as follows:

```

1  /* styles */
2  .container {
3    background-color: lightblue;
4    display: grid;
5    grid-template-columns: 100px 100px 100px 100px 100px;
6    grid-template-rows: 50px 50px;
7    justify-items: center;
8    font-size: 40px;
9    color: black;
10   background: lightgrey;
11   display: grid;
12   grid-template-columns: 100px 100px 100px;
13   /* Only change code below this line */
14
15   /* Only change code above this line */
16
17   /* Only change code below this line */
18
19   /* Only change code above this line */
20
21   /* Only change code below this line */
22   <div class="container">
23     <div class="d1">1</div>
24     <div class="d2">2</div>
25     <div class="d3">3</div>
26     <div class="d4">4</div>
27     <div class="d5">5</div>
28   </div>

```

The browser's developer tools are visible at the top, showing tabs for CSS, JS, and other resources. The page content area displays a 5x2 grid of colored boxes (light blue, light green, light red, light orange, light purple) arranged in two rows of five columns each. The first row contains boxes 1 through 5, and the second row contains boxes 6 through 10. The code editor has several sections: 'Diseño web (receptor heredado)', 'Cuadrícula CSS', 'Añadir filas con grid-template-rows', 'Ejecutar las pruebas (Ctrl + Enter)', 'Restablecer todo el código', and 'Consigue ayuda -'. A note at the bottom left says 'container debe tener una grid-template-rows propiedad con dos unidades de 50px.'

The screenshot shows a browser window with several tabs open, including 'outil.', 'My x', 'Form x', 'CSS x', 'G�� x', 'Learn x', 'Com x', 'Instr x', 'Atrib x', 'Perfi x', 'pepi x', 'Escu x', 'U CURS x', 'Nue x', and 'Actualizar'. The main content area displays a freeCodeCamp tutorial on responsive web design. The title is 'freeCodeCamp(A)'. A search bar at the top left says 'Buscar más de 8000 tutoriales' and 'Diseño web: receptivo heredado'. The main content area has a heading 'Cuadrícula CSS' and a sub-section 'Agrega filas con grid-template-rows'. It contains the following text: 'La cuadrícula que creó en el último desafío establecerá el número de filas automáticamente. Para ajustar las filas manualmente, use la grid-template-rows propiedad de la misma manera que usó grid-template-columns en el desafío anterior.' Below this is a note: 'Agregue más filas a la cuadrícula que sean 100px altas cada una.' A button labeled 'Ejecutar las pruebas (Ctrl + Enter)' is visible. A link 'Restablecer todo el código' is present. A 'Consigue ayuda -' button is also shown. The right side of the screen displays a code editor with the following CSS code:

```
1  /*style*/
2  #grid {
3     background: lightblue;
4     #grid {
5        background: pink;
6        #grid {
7            background: palegreen;
8        }
9        .container {
10           font-size: 40px;
11           #grid {
12              background: lightgray;
13              #grid {
14                 grid-template-columns: 100px 100px 100px;
15                 /* Only change code below this line */
16                 #grid {
17                    grid-template-rows: 100px 100px 100px;
18                    /* Only change code above this line */
19                }
20            }
21            <div class="container">
22              <div class="d1"></div>
23              <div class="d2"></div>
24              <div class="d3"></div>
25              <div class="d4"></div>
26              <div class="d5"></div>
27            </div>
28          
```

At the bottom of the code editor, there is a note: '/* Su sección de prueba irá aquí */'. The status bar at the bottom right shows '931' and '26/09/2022'.

The screenshot shows a browser window with multiple tabs open, including 'outl.', 'My /', 'Form x', 'CSS x', 'Gra...', 'Learn x', 'Goo...', 'Instr...', 'Atrib...', 'Perf...', 'pepi x', 'Escr...', 'Cur...', 'Nue...', and 'Actualizar...'. The main content area is a FreeCodeCamp exercise titled 'Cuadricula CSS' (Grid CSS). The task is to add rows to a grid template. A modal window titled '¡Está activado como Donkey Kong!' is displayed, featuring a large black circle with a white checkmark icon. Below the icon, the text 'Cuadricula CSS' is visible, along with a progress bar indicating '14% completo'. At the bottom of the modal, there are two buttons: 'Enviar e ir al siguiente desafío (Ctrl + Enter)' and 'Descarga mi solución'. The main exercise interface includes sections for 'Ejecutar las pruebas (Ctrl + Enter)', 'Restablecer todo el código', 'Consigue ayuda', and 'Probar'. A note at the bottom states: 'ContainerLa clase debe tener una grid-template-rows propiedad con dos unidades de 30px.' The status bar at the bottom right shows '9:31'.

```
<style>  
.d1{background:LightSkyBlue;}  
.d2{background:LightSalmon;}  
.d3{background:PaleTurquoise;}  
.d4{background:LightPink;}  
.d5{background:PaleGreen;}
```

```

.container {
    font-size: 40px;
    width: 100%;
    background: LightGray;
    display: grid;
    grid-template-columns: 100px 100px 100px;
    /* Only change code below this line */

grid-template-rows: 50px 50px;

    /* Only change code above this line */
}

</style>

<div class="container">
    <div class="d1">1</div>
    <div class="d2">2</div>
    <div class="d3">3</div>
    <div class="d4">4</div>
    <div class="d5">5</div>
</div>

```

4. Use unidades de cuadricula CSS para cambiar el tamaño de columnas y filas

Puede usar unidades absolutas y relativas como **px** y **em** en CSS Grid para definir el tamaño de filas y columnas. Puedes usar estos también:

fr: establece la columna o fila en una fracción del espacio disponible,

auto: establece la columna o fila al ancho o alto de su contenido automáticamente,

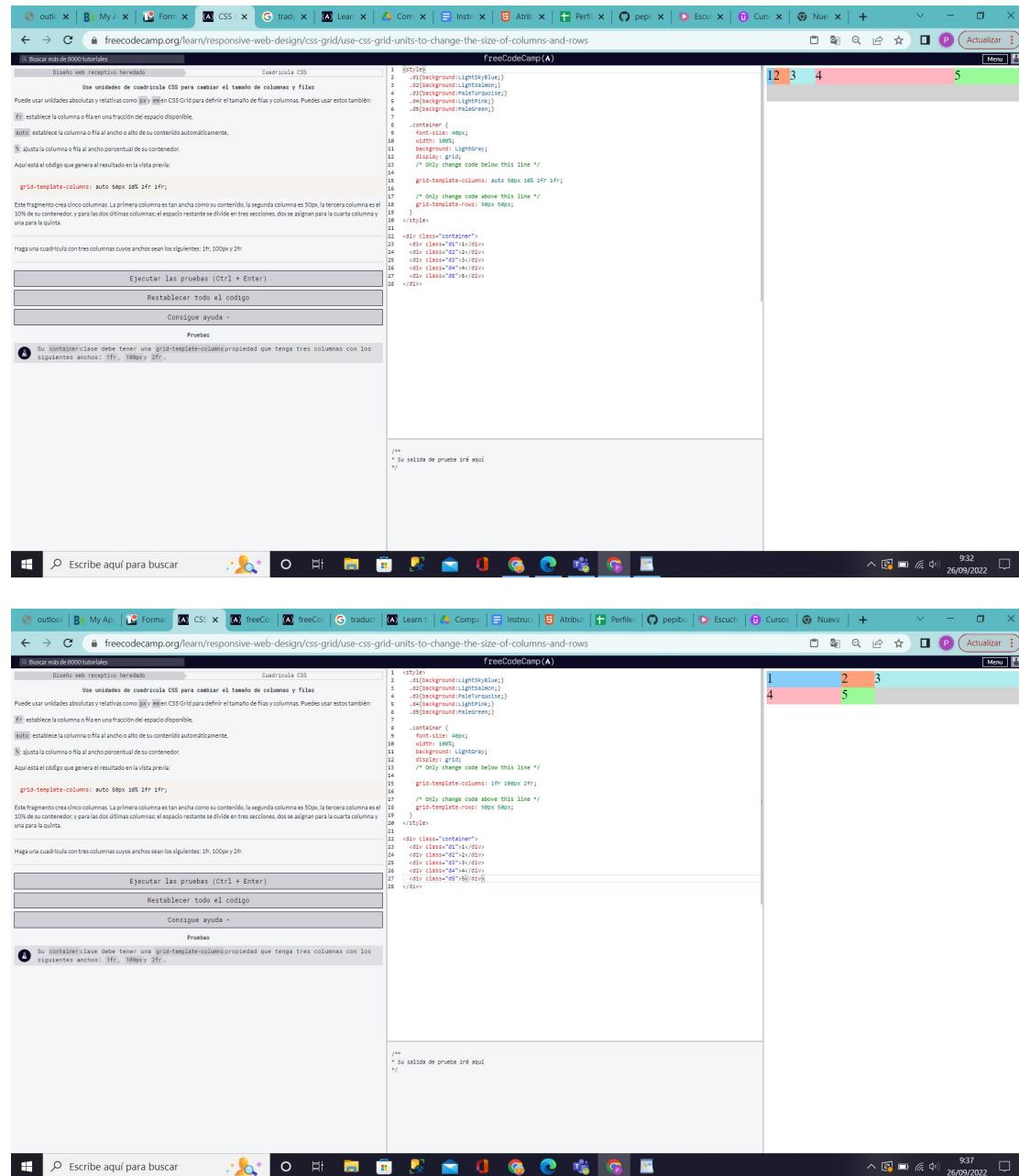
%: ajusta la columna o fila al ancho porcentual de su contenedor.

Aquí está el código que genera el resultado en la vista previa:

grid-template-columns: auto 50px 10% 2fr 1fr;

Este fragmento crea cinco columnas. La primera columna es tan ancha como su contenido, la segunda columna es 50px, la tercera columna es el 10% de su contenedor, y para las dos últimas columnas; el espacio restante se divide en tres secciones, dos se asignan para la cuarta columna y una para la quinta.

Haga una cuadrícula con tres columnas cuyos anchos sean los siguientes: 1fr, 100px y 2fr.



```

1  style="background: lightblue;">
2  <div> background: lightblue;
3  </div>
4  <div> background: pink;
5  </div>
6  <div> background: lightgreen;
7  </div>
8  <div> background: lightgrey;
9  </div>
10 <div> background: lightpink;
11 </div>
12 <div> display: grid;
13 /* Only change code below this line */
14
15 <grid-template-columns: auto 50px 10% 1fr 100px 2fr; */
16 /* Only change code above this line */
17 <grid-template-rows: 10px 50px; */
18
19 </div>
20 </div>
21 <div> class="container">
22 <div> class="col-1"></div>
23 <div> class="col-2"></div>
24 <div> class="col-3"></div>
25 <div> class="col-4"></div>
26 <div> class="col-5"></div>
27 </div>
28 </div>

```

Ejecutar las pruebas (Ctrl + Enter)

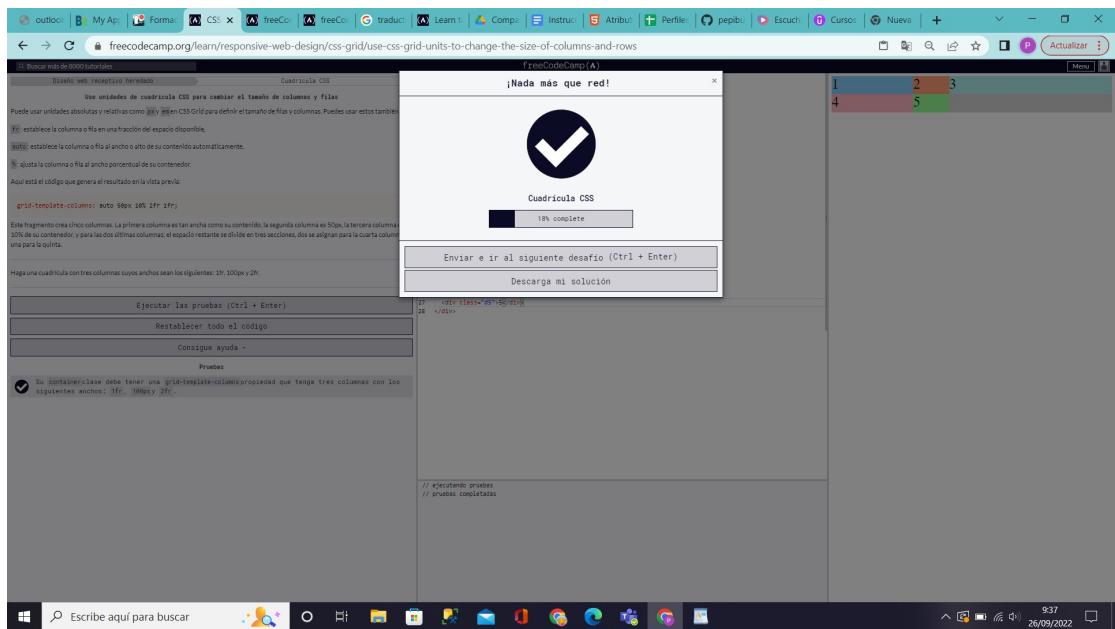
Restablecer todo el código

Consigue ayuda -

Pruebas

/* Su container debe tener una grid-template-columns propiedad que tenga tres columnas con los siguientes anchos: 1fr, 100px y 2fr.

/*
* Su salida de prueba irá aquí
*/



Dado que el desafío requiere que establezca el ancho en 1fr, 100px y 2fr, cambie la css grid-template-columns propiedad de .container a:

grid-template-columns: 1fr 100px 2fr;

```
<style>
.d1{background:LightSkyBlue;}
.d2{background:LightSalmon;}
.d3{background:PaleTurquoise;}
.d4{background:LightPink;}
.d5{background:PaleGreen;}
.container {
    font-size: 40px;
    width: 100%;
    background: LightGray;
    display: grid;
/* Only change code below this line */

```

grid-template-columns: 1fr 100px 2fr;

```
/* Only change code above this line */
```

```

grid-template-rows: 50px 50px;

}

</style>

<div class="container">

<div class="d1">1</div>

<div class="d2">2</div>

<div class="d3">3</div>

<div class="d4">4</div>

<div class="d5">5</div>

</div>

```

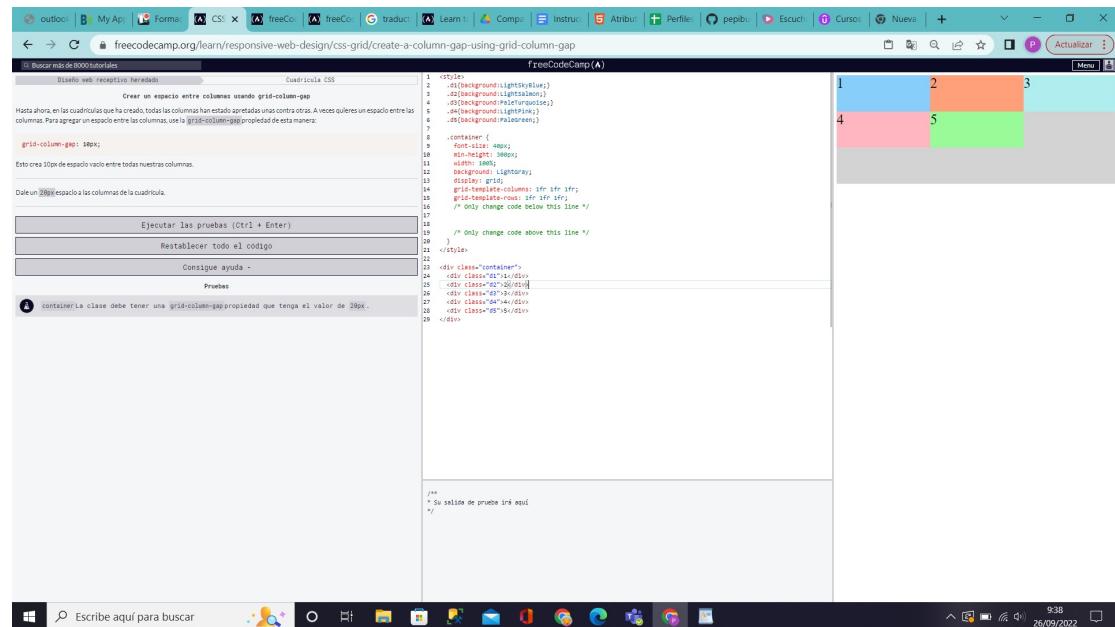
5. Crear un espacio entre columnas usando grid-column-gap

Hasta ahora, en las cuadrículas que ha creado, todas las columnas han estado apretadas unas contra otras. A veces quieras un espacio entre las columnas. Para agregar un espacio entre las columnas, use la `grid-column-gap` propiedad de esta manera:

`grid-column-gap: 10px;`

Esto crea 10px de espacio vacío entre todas nuestras columnas.

Dale un 20px espacio a las columnas de la cuadrícula.



The screenshot shows a browser window with a code editor on the left and a preview on the right. The code editor contains the following CSS:

```

.grid {
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    grid-template-rows: 50px 50px;
    grid-column-gap: 20px;
}

.container {
    width: 100px;
    height: 300px;
    background-color: lightblue;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr;
    grid-column-gap: 20px;
}

```

The preview shows a 3x3 grid of colored cells. The columns have a gap of 20px between them. The first row has three columns (light blue, orange, teal). The second row has three columns (pink, green, grey). The third row has three columns (grey, grey, grey).

[outlook](#) [B My Ap.](#) [Forma.](#) [CSS](#) [freeCodeCamp](#) [freeCodeCamp](#) [Traduc.](#) [Learn](#) [Amp](#) [Comp.](#) [Instruc.](#) [Atribui.](#) [Perfile](#) [pepib](#) [Escuchar](#) [Cursos](#) [Nueva](#) [+](#)

Buscar más de 8000 tutoriales

Diseño web: recepción heredado Cuadricula CSS Crear un espacio entre columnas usando grid-column-gap

Hasta ahora, en las cuadrículas que has creado, todas las columnas han estado apretadas una contra otra. A veces querrás un espacio entre las columnas. Para agregar un espacio entre las columnas, usa la `grid-column-gap` propiedad de esta manera:

```
grid-column-gap: 10px;
```

Esto crea 10px de espacio vacío entre todas nuestras columnas.

Dale un `20px` espacio a las columnas de la cuadricula.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigne ayuda -

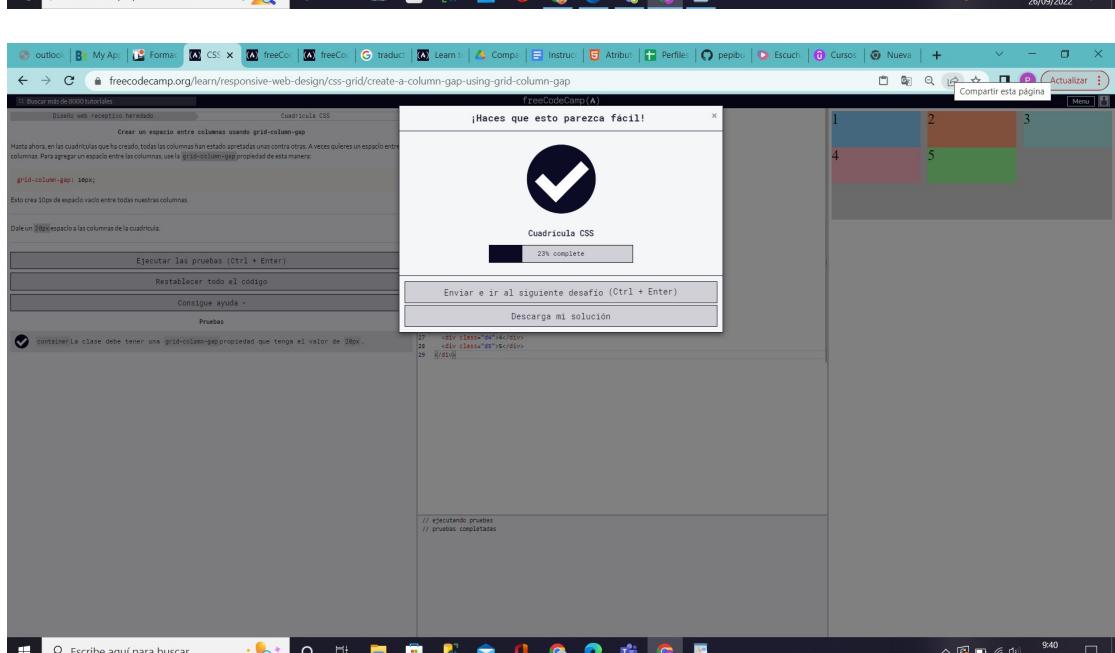
Pruebas

container La clase debe tener una `grid-column-gap` propiedad que tenga el valor de `20px`.

1 `/*style*/`
2 `.grid{background:lightblue;}`
3 `.grid2{background:#ff9966;}`
4 `.grid3{background:#99cc66;}`
5 `.grid4{background:#cc6666;}`
6 `.grid5{background:#339999;}`
7 `/*Container*/`
8 `.container {`
9 `font-size: 40px;`
10 `padding: 10px;`
11 `width: 100%;`
12 `background-color: lightgray;`
13 `border: 1px solid black;`
14 `grid-template-columns: 1fr 1fr 1fr 1fr;`
15 `grid-template-rows: 1fr 1fr 1fr 1fr;`
16 `/* Only change code below this line */`
17 `grid-column-gap: 20px;`

18 `/* Only change code above this line */`
19 `justify-content: space-around;`
20 `}`
21 `</style>`
22 `</head>`
23 `<div class="container">`
24 `<div class="grid1">1</div>`
25 `<div class="grid2">2</div>`
26 `<div class="grid3">3</div>`
27 `<div class="grid4">4</div>`
28 `<div class="grid5">5</div>`
29 `</div>`

/* Su salida de prueba irá aquí



```
<style>  
  
.d1{background:LightSkyBlue;}  
  
.d2{background:LightSalmon;}  
  
.d3{background:PaleTurquoise;}  
  
.d4{background:LightPink;}  
  
.d5{background:PaleGreen;}
```

```

.container {
    font-size: 40px;
    min-height: 300px;
    width: 100%;
    background: LightGray;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr;
    /* Only change code below this line */

grid-column-gap: 20px;

    /* Only change code above this line */
}

</style>

<div class="container">
    <div class="d1">1</div>
    <div class="d2">2</div>
    <div class="d3">3</div>
    <div class="d4">4</div>
    <div class="d5">5</div>
</div>

```

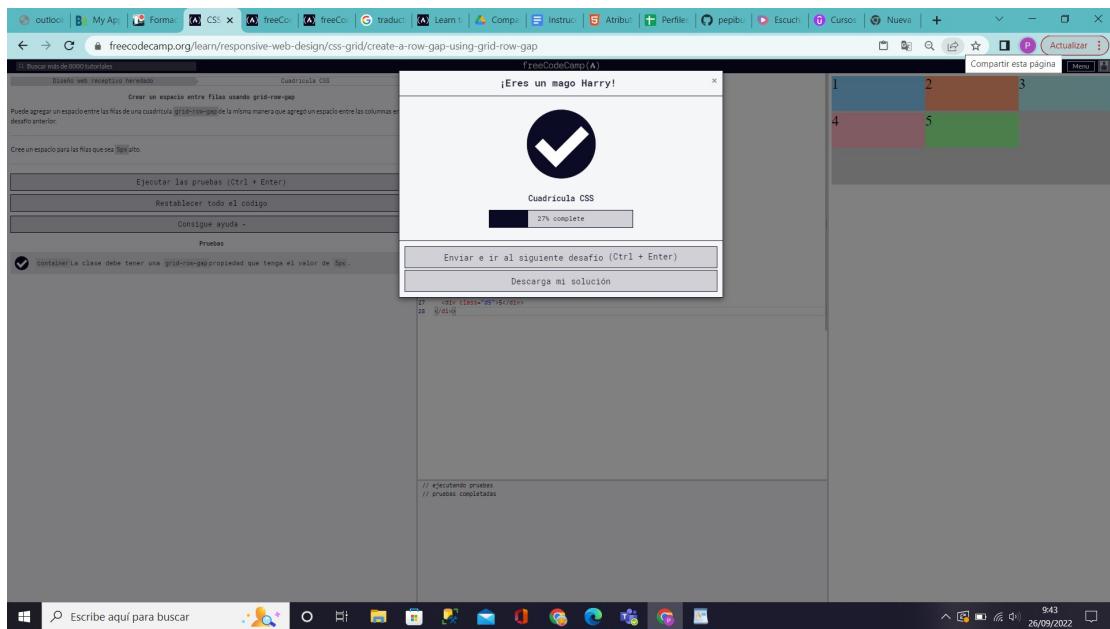
6. Crear un espacio entre filas usando grid-row-gap

Puede agregar un espacio entre las filas de una cuadrícula **grid-row-gap** de la misma manera que agregó un espacio entre las columnas en el desafío anterior.

Cree un espacio para las filas que sea **5px** alto.

Crear un espacio entre filas usando grid-row-gap

```
1 .style{background-color:#f0f0f0}
2 .grid{background-color:#f0f0f0}
3 .grid>div{background-color:#fff}
4 .grid>div:nth-child(1n){background-color:#007bff}
5 .grid>div:nth-child(2n){background-color:#dc3545}
6 .grid>div:nth-child(3n){background-color:#28a745}
7 .grid>div:nth-child(4n){background-color:#ffc107}
8 .grid>div:nth-child(5n){background-color:#6c757d}
9 .container{font-size:40px;
10 width:100px;
11 height:30px;
12 background-color:#f0f0f0;
13 border:1px solid #ccc;
14 grid-template-columns:1fr 1fr 1fr;
15 grid-template-rows:1fr 1fr 1fr;
16 /* Only change code below this line */
17 grid-row-gap:10px;
18 /* Only change code above this line */
19 }
20 /* Only change code above this line */
21 }
22 /* Only change code above this line */
23 <div class="grid">
24 <div>1</div>
25 <div>2</div>
26 <div>3</div>
27 <div>4</div>
28 <div>5</div>
29 </div>
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
22
```



```

<style>
    .d1{background:LightSkyBlue;}
    .d2{background:LightSalmon;}
    .d3{background:PaleTurquoise;}
    .d4{background:LightPink;}
    .d5{background:PaleGreen;}
    .container {
        font-size: 40px;
        min-height: 300px;
        width: 100%;
        background: LightGray;
        display: grid;
        grid-template-columns: 1fr 1fr 1fr;
        grid-template-rows: 1fr 1fr 1fr;
    }
    /* Only change code below this line */
    .container {
        grid-row-gap: 5px;
    }
    /* Only change code above this line */

```

```
    }  
  
</style>  
  
<div class="container">  
  
    <div class="d1">1</div>  
  
    <div class="d2">2</div>  
  
    <div class="d3">3</div>  
  
    <div class="d4">4</div>  
  
    <div class="d5">5</div>  
  
</div>
```

7. Agregue brechas más rápido con grid-gap

grid-gap es una propiedad abreviada para **grid-row-gap** y **grid-column-gap** de los dos desafíos anteriores que es más conveniente de usar. Si grid-gap tiene un valor, creará un espacio entre todas las filas y columnas. Sin embargo, si hay **dos valores**, utilizará el **primero** para establecer el espacio entre las **filas** y el segundo valor para las columnas.

Úselo **grid-gap** para introducir un **10px** espacio entre las filas y un **20px** espacio entre las columnas.

Buscar más de 8000 tutoriales

Diseño web heredado Cuadricula CSS freeCodeCamp(A)

Agregue brechas más rápida con grid-gap

grid-gap es una propiedad sobrepara grid-column-gap y grid-row-gap. Los dos decautos anteriores son más convenientes de usar. Si grid-gap tiene un valor, creará un espacio entre todas las filas y columnas. Sin embargo, si hay dos valores, utilizará el primero para establecer el espacio entre las filas y el segundo valor para las columnas.

Usted grid-gap para introducir un 10pxespacio entre las filas y un 20pxespacio entre las columnas.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

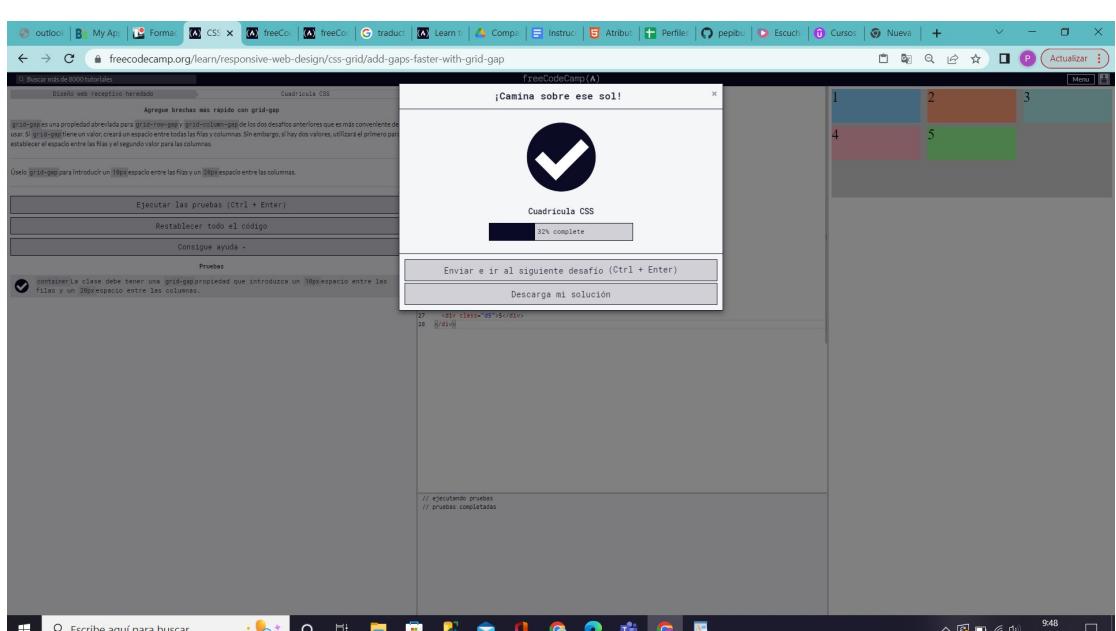
Pruebas

contieneLa clase debe tener una grid-gappropiedad que introduce un 10pxespacio entre las filas y un 20pxespacio entre las columnas.

```
1 <ht>1<br>
2 <div>grid-gap:10px</div>
3 <div>grid-gap:10px 20px</div>
4 <div>grid-template-columns:1fr 1fr 1fr 1fr</div>
5 <div>grid-template-rows:1fr 1fr 1fr</div>
6 <div>grid-template-columns:1fr 1fr 1fr 1fr</div>
7 <div>grid-template-rows:1fr 1fr 1fr</div>
8 <div>grid-template-columns:1fr 1fr 1fr 1fr</div>
9 <div>grid-template-rows:1fr 1fr 1fr</div>
10 <div>grid-template-columns:1fr 1fr 1fr 1fr</div>
11 <div>grid-template-rows:1fr 1fr 1fr</div>
12 <div>grid-template-columns:1fr 1fr 1fr 1fr</div>
13 <div>grid-template-rows:1fr 1fr 1fr</div>
14 <div>grid</div>
15 <div>grid-template-columns:1fr 1fr 1fr 1fr</div>
16 <div>grid-template-rows:1fr 1fr 1fr</div>
17 <div>grid</div>
18 <div>grid-template-columns:1fr 1fr 1fr 1fr</div>
19 <div>grid</div>
20 <div>grid</div>
21 </style>
22 <div><div>grid</div></div>
23 <div><div>grid</div></div>
24 <div><div>grid</div></div>
25 <div><div>grid</div></div>
26 <div><div>grid</div></div>
27 <div><div>grid</div></div>
28 </div>
```

/**
* Su salida de prueba irá aquí
*/

1 2 3
4 5



```
<style>  
.d1{background:LightSkyBlue;}  
.d2{background:LightSalmon;}  
.d3{background:PaleTurquoise;}  
.d4{background:LightPink;}  
.d5{background:PaleGreen;}
```

```

.container {
    font-size: 40px;
    min-height: 300px;
    width: 100%;
    background: LightGray;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr;
    /* Only change code below this line */
grid-gap: 10px 20px;
    /* Only change code above this line */
}
</style>
<div class="container">
    <div class="d1">1</div>
    <div class="d2">2</div>
    <div class="d3">3</div>
    <div class="d4">4</div>
    <div class="d5">5</div>
</div>

```

8. Use grid-column para controlar el espaciado

Hasta este punto, todas las propiedades que se han discutido son para contenedores de cuadrícula. La `grid-column` propiedad es la primera que se utiliza en los propios elementos de la cuadrícula.

Las líneas horizontales y verticales hipotéticas que crean la cuadrícula se denominan líneas . Estas líneas están numeradas comenzando con 1 en la esquina superior izquierda de la cuadrícula y se mueven hacia la derecha para las columnas y hacia abajo para las filas, contando hacia arriba.

Así es como se ven las líneas para una cuadrícula de 3x3:

Líneas de columna1234líneas de fila1234

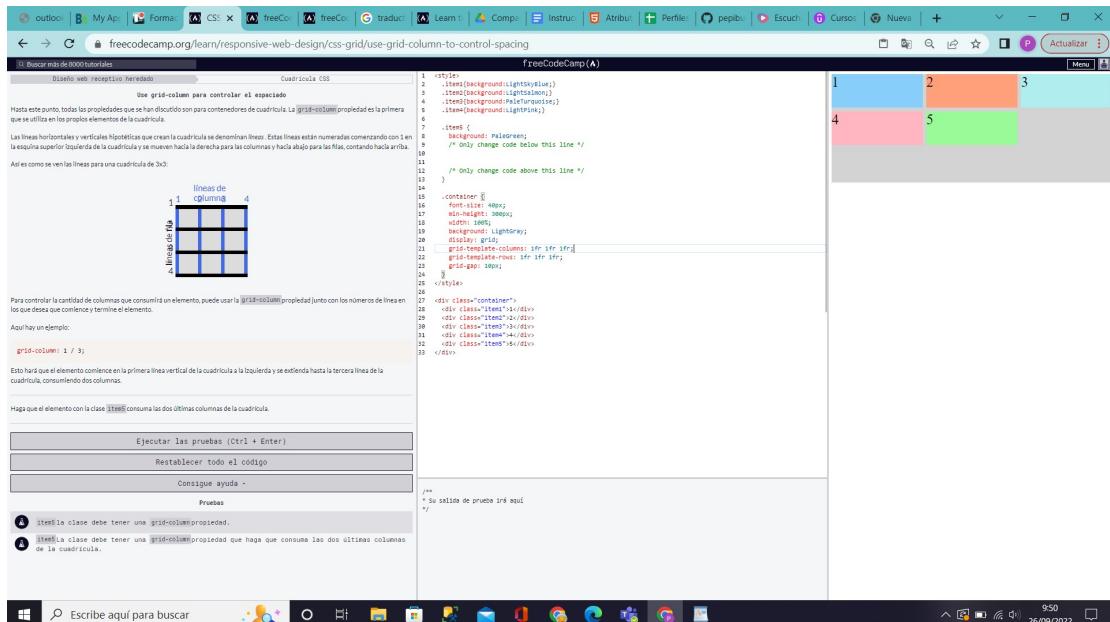
Para controlar la cantidad de columnas que consumirá un elemento, puede usar la `grid-column` propiedad junto con los números de línea en los que desea que comience y termine el elemento.

Aquí hay un ejemplo:

`grid-column: 1 / 3;`

Esto hará que el elemento comience en la primera línea vertical de la cuadrícula a la izquierda y se extienda hasta la tercera línea de la cuadrícula, consumiendo dos columnas.

Haga que el elemento con la clase `item5` consuma las dos últimas columnas de la cuadrícula.



The screenshot shows a browser window with a CSS grid example. The grid has 4 columns and 3 rows. Column 1 is light blue, Column 2 is orange, Column 3 is green, and Column 4 is pink. Row 1 contains items 1, 2, and 3. Row 2 contains item 4. Row 3 contains item 5. A code editor on the left shows CSS for creating this grid and a diagram below it illustrating the grid structure with numbered lines.

grid-column: 1 / 3;

Este hará que el elemento comience en la primera línea vertical de la cuadrícula a la izquierda y se extienda hasta la tercera línea de la cuadrícula, consumiendo dos columnas.

Haga que el elemento con la clase `item5` consuma las dos últimas columnas de la cuadrícula.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

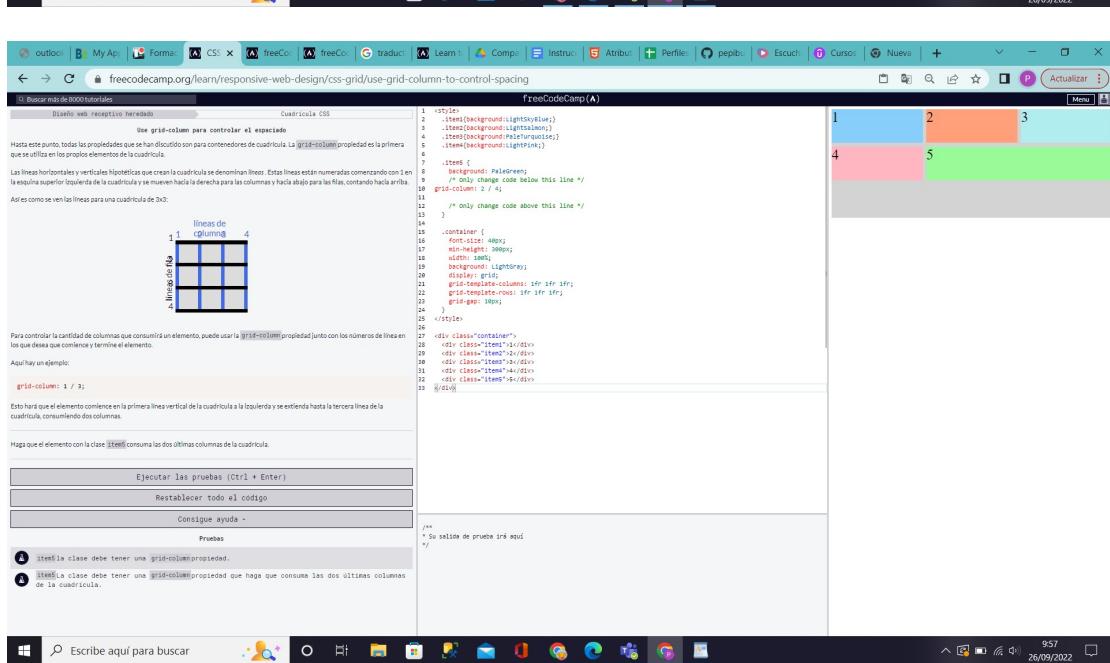
Consigue ayuda -

Pruebas

/* Su salida de prueba irá aquí

item5 La clase debe tener una `grid-column` propiedad.

item5 La clase debe tener una `grid-column` propiedad que haga que consuma las dos últimas columnas de la cuadrícula.



The screenshot shows a second browser window with a similar CSS grid example. The grid has 4 columns and 3 rows. Column 1 is light blue, Column 2 is orange, Column 3 is green, and Column 4 is pink. Row 1 contains items 1, 2, and 3. Row 2 contains item 4. Row 3 contains item 5. A code editor on the left shows the same CSS as the first screenshot.

grid-column: 1 / 3;

Este hará que el elemento comience en la primera línea vertical de la cuadrícula a la izquierda y se extienda hasta la tercera línea de la cuadrícula, consumiendo dos columnas.

Haga que el elemento con la clase `item5` consuma las dos últimas columnas de la cuadrícula.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

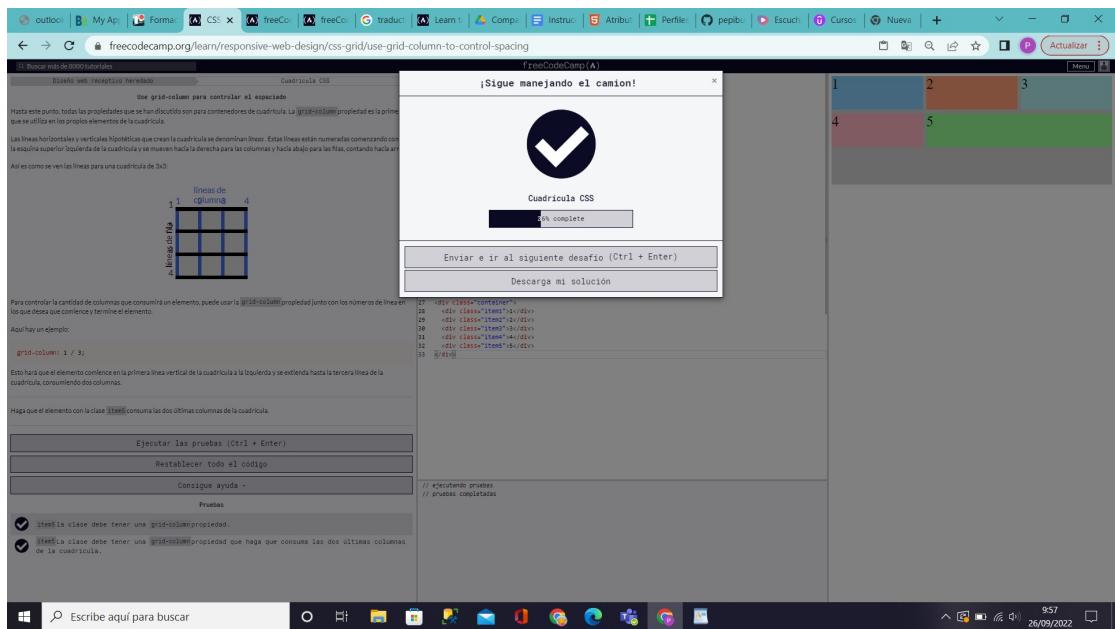
Consigue ayuda -

Pruebas

/* Su salida de prueba irá aquí

item5 La clase debe tener una `grid-column` propiedad.

item5 La clase debe tener una `grid-column` propiedad que haga que consuma las dos últimas columnas de la cuadrícula.



```

<style>

.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5 {

    background: PaleGreen;

/* add your code below this line */

grid-column: 2 / 4;

/* add your code above this line */

}

.container {

    font-size: 40px;

    min-height: 300px;

    width: 100%;

    background: LightGray;

    display: grid;

```

```

grid-template-columns: 1fr 1fr 1fr;
grid-template-rows: 1fr 1fr 1fr;
grid-gap: 10px;
}

</style>

<div class="container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
</div>

```

SOLUCIÓN ALTERNATIVA:

```

.item5 {
  background: PaleGreen;
  /* add your code below this line */

  grid-column: 2;
  grid-column-end: 4;
  /* add your code above this line */
}

```

9. Use grid-row para controlar el espaciado

Por supuesto, puede hacer que los elementos consuman varias filas al igual que con las columnas. Usted define las líneas horizontales en las que desea que un elemento comience y termine usando la `grid-row` propiedad en un elemento de cuadrícula.

Haz que el elemento con la `item5` clase consuma las dos últimas filas.

```
/*styling*/
.item1{background:lightblue;}
.item2{background:lightorange;}
.item3{background:palegreen;}
.item4{background:lightpink;}
.item5{background:lightgray;}
```

Use grid-row para controlar el espacio.

Por supuesto, puede hacer que los elementos consuman varias filas al igual que con las columnas. Usted define las líneas horizontales en las que desea que un elemento comience y termine usando la `grid-row` propiedad en un elemento de cuadrícula.

Haz que el elemento con la `.item5` clase consuma las dos últimas filas.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

item5La clase debe tener una `grid-row` propiedad.

item5La clase debe tener una `grid-row` propiedad que haga que consume las dos últimas filas de la cuadrícula.

/* Only change code above this line */

```
1 <style>
2   .item1{background:lightblue;}
3   .item2{background:lightorange;}
4   .item3{background:palegreen;}
5   .item4{background:lightpink;}
6   .item5{background:lightgray;}
```

/* Only change code below this line */

```
7   .item5 {
8     grid-column: 2 / 4;
9     grid-row: 1fr 1fr 1fr;
10    /* Only change code above this line */
11  }
12  .container {
13   font-size: 40px;
14   min-height: 200px;
15   padding: 10px;
16   background: lightgrey;
17   display: grid;
18   grid-template-columns: 1fr 1fr 1fr;
19   grid-template-rows: 1fr 1fr 1fr;
20   grid-gap: 10px;
21 }
22 </style>
```

/* Su salida de prueba irá aquí

```
/*styling*/
.item1{background:lightblue;}
.item2{background:lightorange;}
.item3{background:palegreen;}
.item4{background:lightpink;}
.item5{background:lightgray;}
```

Use grid-row para controlar el espacio.

Por supuesto, puede hacer que los elementos consuman varias filas al igual que con las columnas. Usted define las líneas horizontales en las que desea que un elemento comience y termine usando la `grid-row` propiedad en un elemento de cuadrícula.

Haz que el elemento con la `.item5` clase consuma las dos últimas filas.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

item5La clase debe tener una `grid-row` propiedad.

item5La clase debe tener una `grid-row` propiedad que haga que consume las dos últimas filas de la cuadrícula.

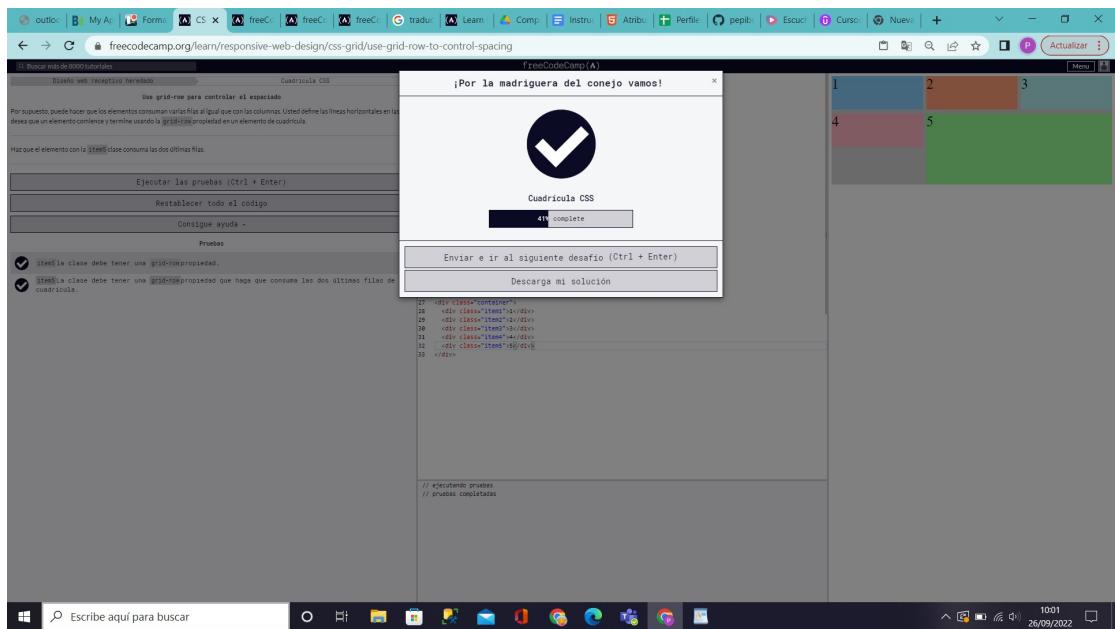
/* Only change code above this line */

```
1 <style>
2   .item1{background:lightblue;}
3   .item2{background:lightorange;}
4   .item3{background:palegreen;}
5   .item4{background:lightpink;}
6   .item5{background:lightgray;}
```

/* Only change code below this line */

```
7   .item5 {
8     grid-column: 2 / 4;
9     grid-row: 1fr 1fr 1fr;
10    /* Only change code above this line */
11  }
12  .container {
13   font-size: 40px;
14   min-height: 200px;
15   padding: 10px;
16   background: lightgrey;
17   display: grid;
18   grid-template-columns: 1fr 1fr 1fr;
19   grid-template-rows: 1fr 1fr 1fr;
20   grid-gap: 10px;
21 }
22 </style>
```

/* Su salida de prueba irá aquí



```

<style>
.item1{background:LightSkyBlue;}
.item2{background:LightSalmon;}
.item3{background:PaleTurquoise;}
.item4{background:LightPink;}
.item5 {
    background: PaleGreen;
    grid-column: 2 / 4;
    /* Only change code below this line */
grid-row: 2 / 4;
    /* Only change code above this line */
}
.container {
    font-size: 40px;
    min-height: 300px;
    width: 100%;
    background: LightGray;
}

```

```

        display: grid;
        grid-template-columns: 1fr 1fr 1fr;
        grid-template-rows: 1fr 1fr 1fr;
        grid-gap: 10px;
    }

</style>

<div class="container">
    <div class="item1">1</div>
    <div class="item2">2</div>
    <div class="item3">3</div>
    <div class="item4">4</div>
    <div class="item5">5</div>
</div>

```

10. Alinee un elemento horizontalmente usando la justificación propia

En CSS Grid, el contenido de cada elemento se encuentra en un cuadro que se denomina celda . Puede alinear la posición del contenido dentro de su celda horizontalmente utilizando la justify-selfpropiedad en un elemento de cuadrícula. Por defecto, esta propiedad tiene un valor de stretch, lo que hará que el contenido ocupe todo el ancho de la celda. Esta propiedad CSS Grid también acepta otros valores:

start: alinea el contenido a la izquierda de la celda,

center: alinea el contenido en el centro de la celda,

end: alinea el contenido a la derecha de la celda.

Utilice la justify-selfpropiedad para centrar el elemento con la clase item2.

```
/* CSS Grid: el contenido de cada elemento se encuentra en un cuadro que se denomina celda. Puede alinear la posición del contenido dentro de su celda horizontalmente utilizando la justify-self propiedades en un elemento de cuadro. Por defecto, esta propiedad tiene un valor de stretch, lo que hará que el contenido ocupe todo el ancho de la celda. Esta propiedad CSS Grid también acepta otros valores:
```

`start`: alinea el contenido a la izquierda de la celda.

`center`: alinea el contenido en el centro de la celda.

`end`: alinea el contenido a la derecha de la celda.

Utilice la `justify-self` propiedad para centrar el elemento con la clase `item2`.

Ejecutar las pruebas (Ctrl+1 + Enter)

Restablecer todo el código

Consegue ayuda ▾

Pruebas

item2La clase debe tener una `justify-self` propiedad que tenga el valor de `center`.

```
1  .style{  
2    .items{background: lightcyan;}  
3    .item2 {  
4      background: lightblue;  
5      /* Only change code below this line */  
6      justify-self: center;  
7    }  
8    /* Only change code above this line */  
9  }  
10 .container {  
11   font-size: 40px;  
12   grid-template-columns: 1fr 1fr 1fr;  
13   grid-template-rows: 1fr 1fr 1fr;  
14   grid-gap: 10px;  
15 }  
16 .container >  
17   .item1{background: #f0f0f0;}  
18   .item2{background: lightpink;}  
19   .item3{background: lightgreen;}  
20   .item4{background: pink;}  
21   .item5{background: lightblue;}  
22 }  
23 .grid-item {  
24   width: 100px;  
25   height: 100px;  
26 }  
27 /*  
28 * Su salida de prueba irá aquí  
29 */  
30 </div>
```

```
/* CSS Grid: el contenido de cada elemento se encuentra en un cuadro que se denomina celda. Puede alinear la posición del contenido dentro de su celda horizontalmente utilizando la justify-self propiedades en un elemento de cuadro. Por defecto, esta propiedad tiene un valor de stretch, lo que hará que el contenido ocupe todo el ancho de la celda. Esta propiedad CSS Grid también acepta otros valores:
```

`start`: alinea el contenido a la izquierda de la celda.

`center`: alinea el contenido en el centro de la celda.

`end`: alinea el contenido a la derecha de la celda.

Utilice la `justify-self` propiedad para centrar el elemento con la clase `item2`.

Ejecutar las pruebas (Ctrl+1 + Enter)

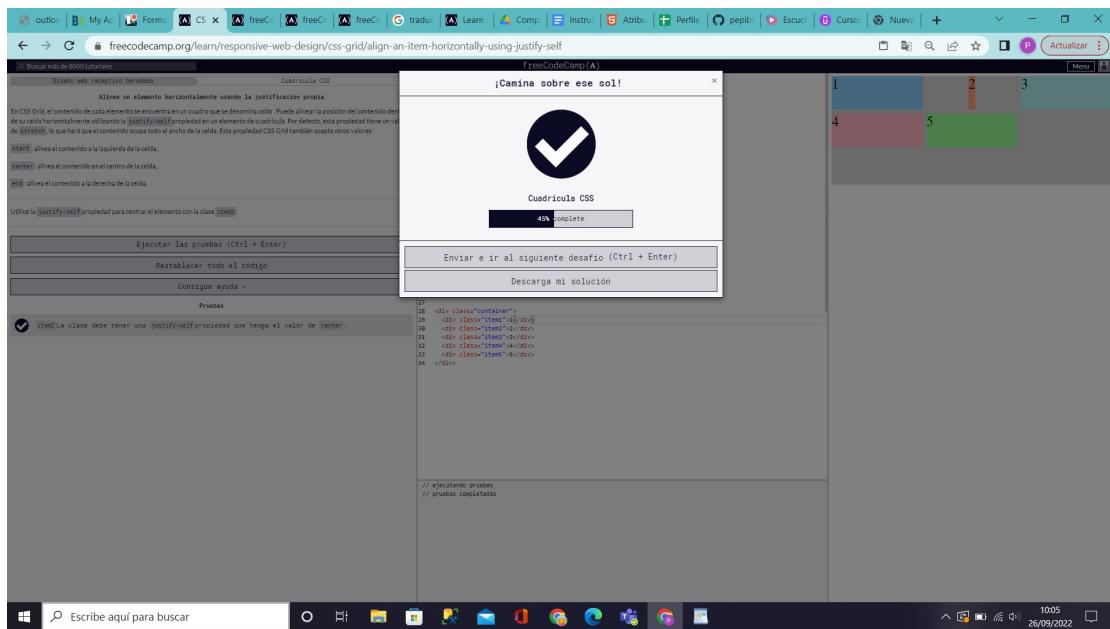
Restablecer todo el código

Consegue ayuda ▾

Pruebas

item2La clase debe tener una `justify-self` propiedad que tenga el valor de `center`.

```
1  .style{  
2    .items{background: lightcyan;}  
3    .item2 {  
4      background: lightblue;  
5      /* Only change code below this line */  
6      justify-self: center;  
7    }  
8    /* Only change code above this line */  
9  }  
10 .container {  
11   font-size: 40px;  
12   grid-template-columns: 1fr 1fr 1fr;  
13   grid-template-rows: 1fr 1fr 1fr;  
14   grid-gap: 10px;  
15 }  
16 .container >  
17   .item1{background: #f0f0f0;}  
18   .item2{background: lightblue;}  
19   .item3{background: lightgreen;}  
20   .item4{background: pink;}  
21   .item5{background: lightblue;}  
22 }  
23 .grid-item {  
24   width: 100px;  
25   height: 100px;  
26 }  
27 /*  
28 * Su salida de prueba irá aquí  
29 */  
30 </div>
```



```

<style>

.item1{background: LightSkyBlue; }

.item2 {
    background: LightSalmon;
    /* Only change code below this line */

justify-self: center;

    /* Only change code above this line */

}

.item3{background:PaleTurquoise; }

.item4{background:LightPink; }

.item5{background:PaleGreen; }

.container {

    font-size: 40px;

    min-height: 300px;

    width: 100%;

    background: LightGray;

    display: grid;

```

```
grid-template-columns: 1fr 1fr 1fr;  
grid-template-rows: 1fr 1fr 1fr;  
grid-gap: 10px;  
}  
</style>  


1



2



3



4



5


```

11. Alinear un elemento verticalmente usando align-self

Así como puede alinear un elemento horizontalmente, también hay una manera de alinear un elemento verticalmente. Para ello, utiliza la **align-self** propiedad en un elemento. Esta propiedad acepta todos los mismos valores justify-self del último desafío.

Alinee el elemento con la clase **item3** verticalmente en el **end**.

The screenshot shows a browser window with the URL <https://freecodecamp.org/learn/responsive-web-design/css-grid/align-an-item-vertically-using-align-self>. The page title is "freeCodeCamp(A)". The content area displays a CSS snippet and its visual output.

HTML:

```
<div>
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</div>
```

CSS:

```
1  <style>
2    <item>1</item>
3    <item>2</item>
4    <item>3</item>
5    <item>4</item>
6    <item>5</item>
7  </style>
8
9  .grid {
10   background-color: lightgray;
11   display: grid;
12   grid-template-columns: 1fr 1fr 1fr;
13   grid-template-rows: 1fr 1fr 1fr;
14   grid-gap: 10px;
15   justify-items: center;
16   align-items: center;
17   font-size: 40px;
18   margin: auto;
19   width: 300px;
20 }
21
22 .item {
23   background-color: lightgray;
24   border-radius: 10px;
25   color: black;
26   padding: 10px;
27 }
28
29 <div class="grid">
30   <div class="item">1</div>
31   <div class="item">2</div>
32   <div class="item">3</div>
33   <div class="item">4</div>
34   <div class="item">5</div>
35 </div>
```

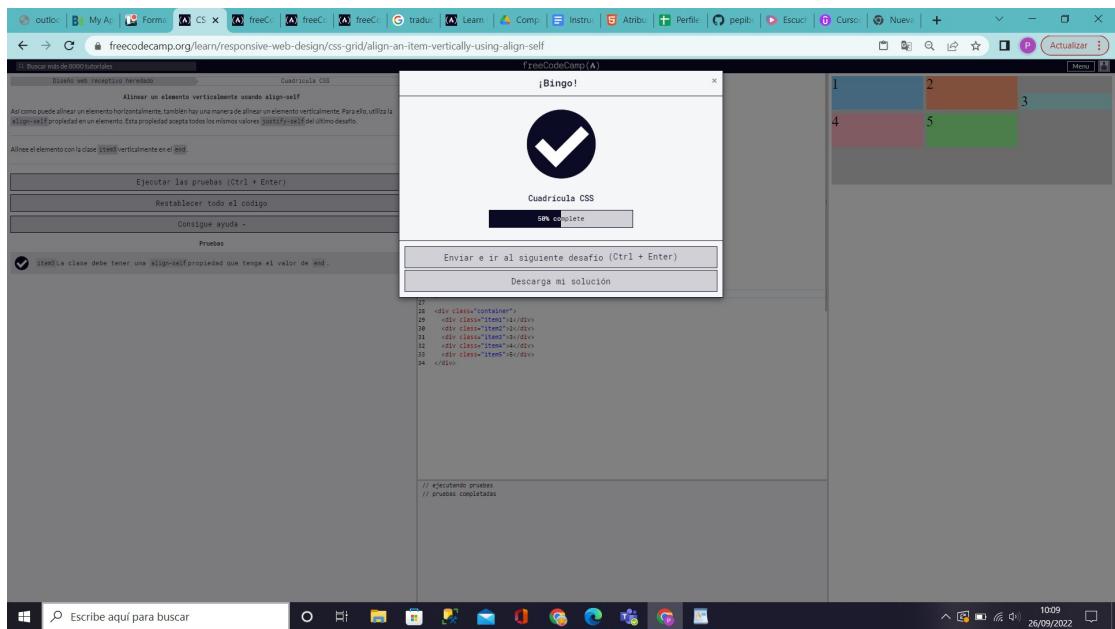
Notes:

- Alinear un elemento verticalmente usando align-self
- Así como puede alinear un elemento horizontalmente, también hay una manera de alinear un elemento verticalmente. Para ello, utiliza la align-self propiedad en un elemento. Esta propiedad acepta todos los mismos valores que justify-self del último desafío.
- Alinee el elemento con la clase .item verticalmente en el grid.
- Ejecutar las pruebas (Ctrl + Enter)
- Restablecer todo el código
- Consigue ayuda -
- Pruebas

Help:

itemBla clase debe tener una align-self propiedad que tenga el valor de end.

This screenshot shows the same browser window as the first one, displaying the same CSS grid example and code. The visual output shows five colored items (1-5) arranged in a 3x2 grid with gaps between them.



```

<style>

.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3 {
    background: PaleTurquoise;
    /* Only change code below this line */

align-self: end;

    /* Only change code above this line */
}

.item4{background:LightPink;}

.item5{background:PaleGreen;}

.container {
    font-size: 40px;
    min-height: 300px;
    width: 100%;

    background: LightGray;
    display: grid;
}

```

```
grid-template-columns: 1fr 1fr 1fr;  
grid-template-rows: 1fr 1fr 1fr;  
grid-gap: 10px;  
}  
</style>  


1



2



3



4



5


```

12. Alinear todos los elementos horizontalmente usando justificar elementos

A veces desea que todos los elementos de su cuadrícula CSS compartan la misma alineación. Puede usar las propiedades aprendidas previamente y alinearlas individualmente, o puede alinearlas todas a la vez horizontalmente usando justify-items en su contenedor de cuadrícula. Esta propiedad puede aceptar los mismos valores que aprendiste en los dos desafíos anteriores, con la diferencia de que moverá todos los elementos de nuestra cuadrícula a la alineación deseada.

Utilice esta propiedad para centrar todos nuestros elementos horizontalmente.

A veces deseas que todos los elementos de tu cuadro CSS comparten la misma alineación. Puedes usar las propiedades aportadas previamente y alinearlos individualmente, o puedes alinearlos todos a la vez horizontalmente usando `justify-items` en su contenedor de cuadro. Esta propiedad puede aceptar los mismos valores que aprendiste en los dos desafíos anteriores, con la diferencia de que moverá todos los elementos de nuestra cuadrícula a la alineación deseada.

Utilice esta propiedad para centrar todos nuestros elementos horizontalmente.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

containerLa clase debe tener una `justify-items` propiedad que tenga el valor de `center`.

```
1 <style>
2   .container {background: lightblue; 
3   .item1 {background: lightgreen; 
4   .item2 {background: lightorange; 
5   .item3 {background: paleyellow; 
6   .item4 {background: pink; 
7   .item5 {background: lightpink; 
8 
9   .container {
10    font-size: 40px;
11    width: 100px;
12    display: grid;
13    grid-template-columns: 1fr 1fr 1fr;
14    grid-template-rows: 1fr 1fr 1fr;
15    grid-gap: 10px;
16    /* Only change code below this line */
17    justify-items: center;
18    /* Only change code above this line */
19  }
20 
21 </style>
22 
23 <div class="container">
24   <div class="item1"></div>
25   <div class="item2"></div>
26   <div class="item3"></div>
27   <div class="item4"></div>
28   <div class="item5"></div>
29 </div>
30 
```

/* Su salida de prueba irá aquí

A veces deseas que todos los elementos de tu cuadro CSS comparten la misma alineación. Puedes usar las propiedades aportadas previamente y alinearlos individualmente, o puedes alinearlos todos a la vez horizontalmente usando `justify-items` en su contenedor de cuadro. Esta propiedad puede aceptar los mismos valores que aprendiste en los dos desafíos anteriores, con la diferencia de que moverá todos los elementos de nuestra cuadrícula a la alineación deseada.

Utilice esta propiedad para centrar todos nuestros elementos horizontalmente.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

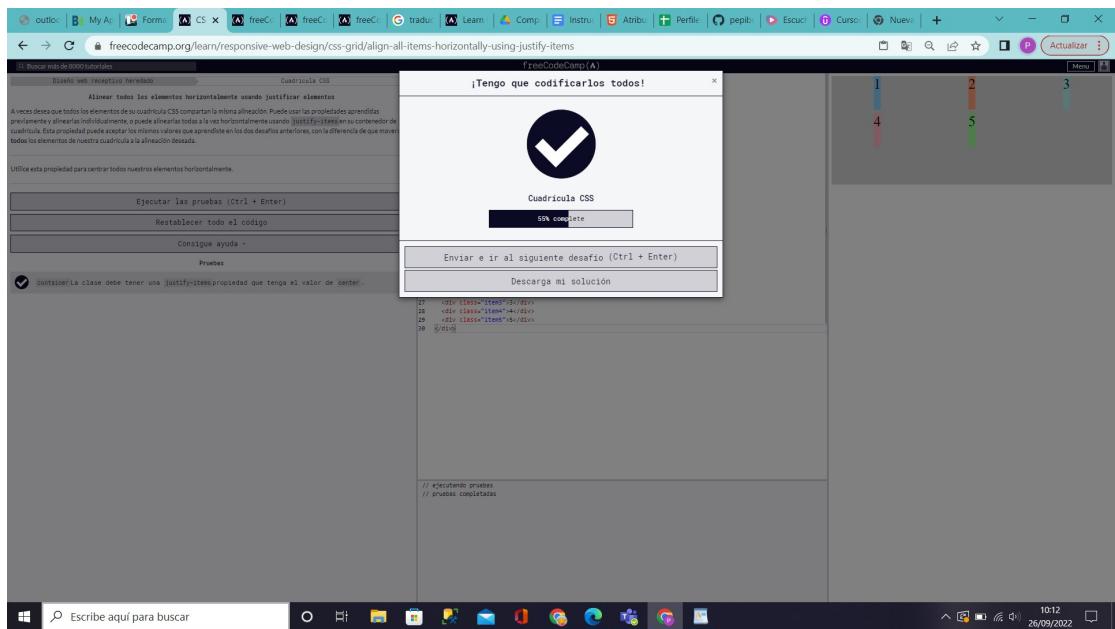
Consigue ayuda -

Pruebas

containerLa clase debe tener una `justify-items` propiedad que tenga el valor de `center`.

```
1 <style>
2   .container {background: lightblue; 
3   .item1 {background: lightgreen; 
4   .item2 {background: lightorange; 
5   .item3 {background: paleyellow; 
6   .item4 {background: pink; 
7   .item5 {background: lightpink; 
8 
9   .container {
10    font-size: 40px;
11    width: 100px;
12    display: grid;
13    grid-template-columns: 1fr 1fr 1fr;
14    grid-template-rows: 1fr 1fr 1fr;
15    grid-gap: 10px;
16    /* Only change code below this line */
17    justify-items: center;
18    /* Only change code above this line */
19  }
20 
21 </style>
22 
23 <div class="container">
24   <div class="item1"></div>
25   <div class="item2"></div>
26   <div class="item3"></div>
27   <div class="item4"></div>
28   <div class="item5"></div>
29 </div>
30 
```

/* Su salida de prueba irá aquí



```

<style>

.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5{background:PaleGreen;}

.container {

font-size: 40px;

min-height: 300px;

width: 100%;

background: LightGray;

display: grid;

grid-template-columns: 1fr 1fr 1fr;

grid-template-rows: 1fr 1fr 1fr;

grid-gap: 10px;

/* Only change code below this line */

justify-items: center;

```

```

/* Only change code above this line */

}

</style>

<div class="container">

<div class="item1">1</div>

<div class="item2">2</div>

<div class="item3">3</div>

<div class="item4">4</div>

<div class="item5">5</div>

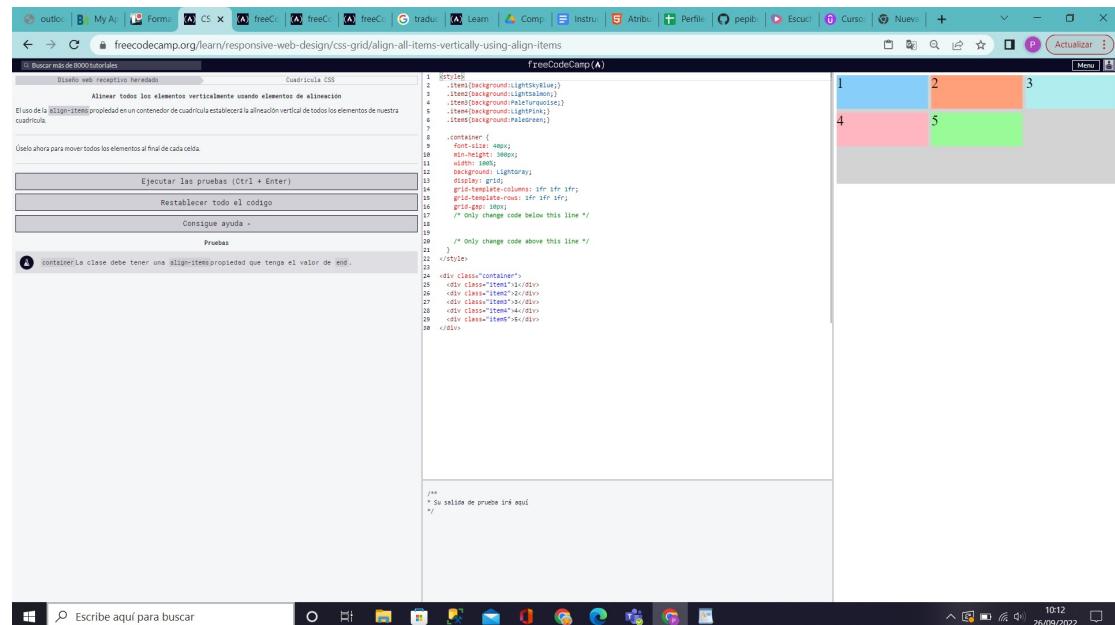
</div>

```

13. Alinear todos los elementos verticalmente usando elementos de alineación

El uso de la **align-items** propiedad en un contenedor de cuadrícula establecerá la alineación vertical de todos los elementos de nuestra cuadrícula.

Úsalo ahora para mover todos los elementos al final de cada celda.



[outlook](#) [My A:](#) [Form](#) [CS](#) [freeC](#) [freeC](#) [freeC](#) [traduc](#) [Learn](#) [Comp](#) [Instru](#) [Attrib](#) [Perfil](#) [pepit](#) [Escuci](#) [Curso](#) [Nuevo](#) + [Actualizar](#)

freeCodeCamp(A)

Alinear todos los elementos verticalmente usando elementos de alineación

El uso de la `align-items` propiedad en un contenedor de cuadrícula establecerá la alineación vertical de todos los elementos de nuestra cuadrícula.

Úsalo ahora para mover todos los elementos al final de cada celda.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

Container CSS

```

1  <style>
2    .container {
3      display: grid;
4      grid-template-columns: 1fr 1fr 1fr;
5      grid-template-rows: 1fr 1fr 1fr;
6      align-items: end; /* Only change code above this line */
7    }
8
9    .item1 {background: LightSkyBlue;}
10   .item2 {background: LightSalmon;}
11   .item3 {background: PaleTurquoise;}
12   .item4 {background: LightPink;}
13   .item5 {background: PaleGreen;}
14
15   .item1, .item2, .item3, .item4, .item5 {
16     width: 100px;
17     height: 30px;
18     border: 1px solid black;
19     padding: 5px;
20     margin: 5px;
21   }
22 </style>
23
24 <div class="container">
25   <div class="item1"></div>
26   <div class="item2"></div>
27   <div class="item3"></div>
28   <div class="item4"></div>
29   <div class="item5"></div>
30 </div>
31
32 /**
33 * Salida de prueba irá aquí
34 */

```

Windows 10 10.15 26/09/2022

[outlook](#) [My A:](#) [Form](#) [CS](#) [freeC](#) [freeC](#) [freeC](#) [traduc](#) [Learn](#) [Comp](#) [Instru](#) [Attrib](#) [Perfil](#) [pepit](#) [Escuci](#) [Curso](#) [Nuevo](#) + [Actualizar](#)

freeCodeCamp(A)

Alinear todos los elementos verticalmente usando elementos de alineación

El uso de la `align-items` propiedad en un contenedor de cuadrícula establecerá la alineación vertical de todos los elementos de nuestra cuadrícula.

Úsalo ahora para mover todos los elementos al final de cada celda.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

Container CSS

Tienes esto!



5% complete

Enviar e ir al siguiente desafío (Ctrl + Enter)

Descarga mi solución

```

17   <div class="item1"></div>
18   <div class="item2"></div>
19   <div class="item3"></div>
20   <div class="item4"></div>
21   <div class="item5"></div>
22
23 /**
24 * Ejecutando pruebas
25 * Pruebas completadas
26 */

```

Windows 10 10.15 26/09/2022

<style>

```

.item1{background:LightSkyBlue;}
.item2{background:LightSalmon;}
.item3{background:PaleTurquoise;}
.item4{background:LightPink;}
.item5{background:PaleGreen;}

```

```

.container {
    font-size: 40px;
    min-height: 300px;
    width: 100%;
    background: LightGray;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr;
    grid-gap: 10px;
    /* Only change code below this line */

align-items: end;

    /* Only change code above this line */
}

</style>

<div class="container">
    <div class="item1">1</div>
    <div class="item2">2</div>
    <div class="item3">3</div>
    <div class="item4">4</div>
    <div class="item5">5</div>
</div>

```

14. Divide la cuadrícula en una plantilla de área

Puede agrupar las celdas de su cuadrícula en un área y darle un nombre personalizado. Haga esto usando `grid-template-areas`en el contenedor de esta manera:

`grid-template-areas:`

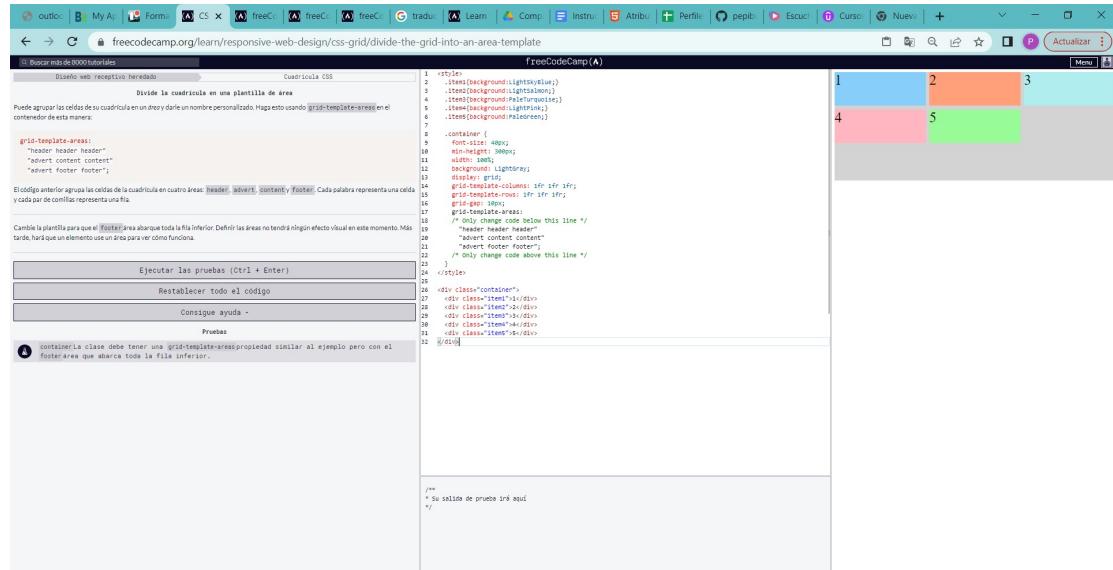
`"header header header"`

`"advert content content"`

"advert footer footer";

El código anterior agrupa las celdas de la cuadrícula en cuatro áreas: header, advert, contenty footer. Cada palabra representa una celda y cada par de comillas representa una fila.

Cambie la plantilla para que el footerárea abarque toda la fila inferior. Definir las áreas no tendrá ningún efecto visual en este momento. Más tarde, hará que un elemento use un área para ver cómo funciona.



```
/* Buscar más de 8000 tutoriales */


freeCodeCamp(A)



Diseño web heredado Cuadrícula CSS



Divide la cuadrícula en una plantilla de área



Puede agrupar las celdas de su cuadrícula en un área y darle un nombre personalizado. Hágalo usando grid-template-areas en el contenedor de esta manera:



```
grid-template-areas:
 "header header header"
 "advert content content"
 "advert footer footer";
```



El código anterior agrupa las celdas de la cuadrícula en cuatro áreas: header, advert, content y footer. Cada palabra representa una celda y cada par de comillas representa una fila.



Cambie la plantilla para que el grid-area abarque toda la fila inferior. Definir las áreas no tendrá ningún efecto visual en este momento. Más tarde, hará que un elemento use un área para ver cómo funciona.



Ejecutar las pruebas (Ctrl + Enter)



Restablecer todo el código



Consigue ayuda -

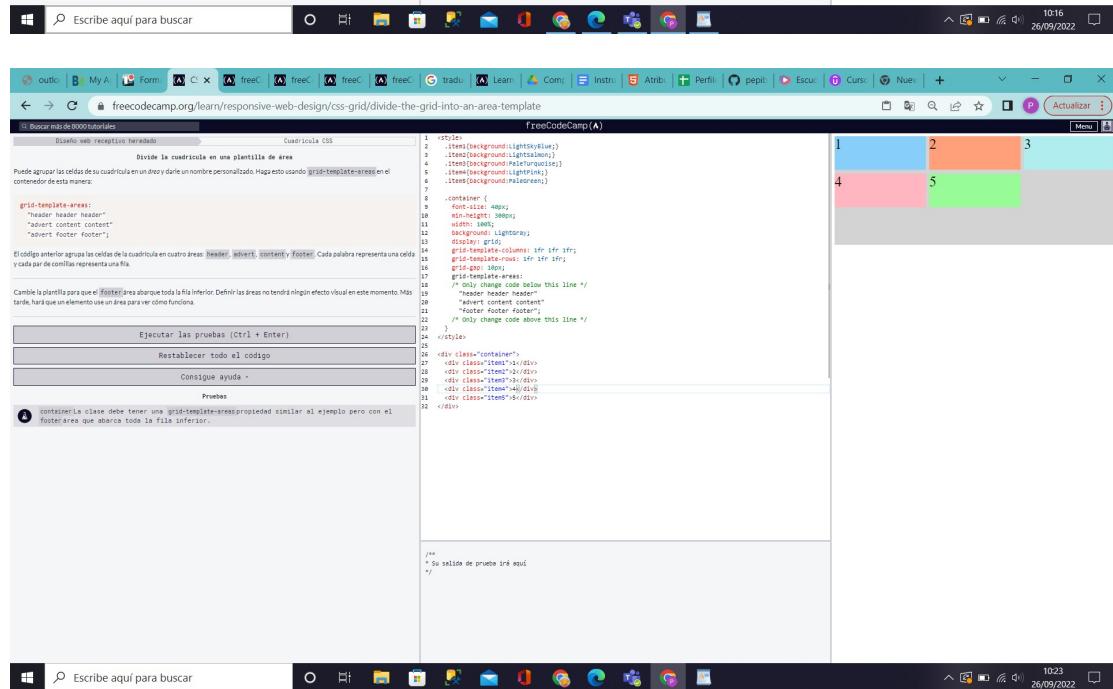


Pruebas



⚠️ containerLa clase debe tener una grid-template-areas propiedad similar al ejemplo pero con el footer área que abarca toda la fila inferior.


```



```
/* Buscar más de 8000 tutoriales */


freeCodeCamp(A)



Diseño web heredado Cuadrícula CSS



Divide la cuadrícula en una plantilla de área



Puede agrupar las celdas de su cuadrícula en un área y darle un nombre personalizado. Hágalo usando grid-template-areas en el contenedor de esta manera:



```
grid-template-areas:
 "header header header"
 "advert content content"
 "advert footer footer";
```



El código anterior agrupa las celdas de la cuadrícula en cuatro áreas: header, advert, content y footer. Cada palabra representa una celda y cada par de comillas representa una fila.



Cambie la plantilla para que el grid-area abarque toda la fila inferior. Definir las áreas no tendrá ningún efecto visual en este momento. Más tarde, hará que un elemento use un área para ver cómo funciona.



Ejecutar las pruebas (Ctrl + Enter)



Restablecer todo el código



Consigue ayuda -

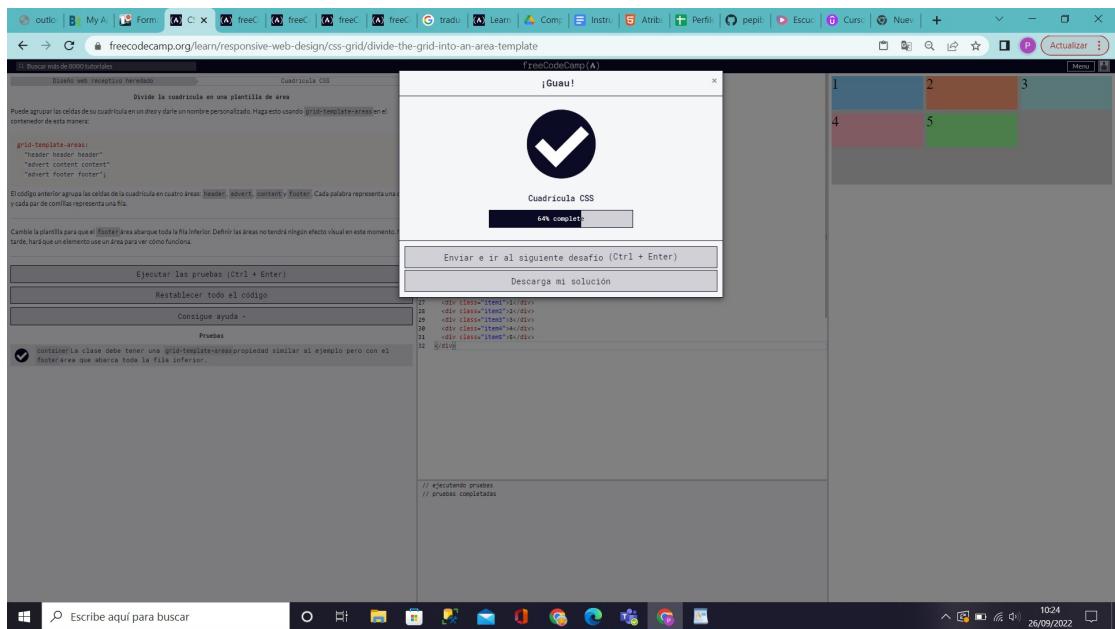


Pruebas



⚠️ containerLa clase debe tener una grid-template-areas propiedad similar al ejemplo pero con el footer área que abarca toda la fila inferior.


```



```

<style>

.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5{background:PaleGreen;}

.container {

font-size: 40px;

min-height: 300px;

width: 100%;

background: LightGray;

display: grid;

grid-template-columns: 1fr 1fr 1fr;

grid-template-rows: 1fr 1fr 1fr;

grid-gap: 10px;

grid-template-areas:

/* Only change code below this line */

```

```

"header header header"

"advert content content"

"footer footer footer";

/* Only change code above this line */

}

</style>

<div class="container">

<div class="item1">1</div>

<div class="item2">2</div>

<div class="item3">3</div>

<div class="item4">4</div>

<div class="item5">5</div>

</div>

```

15. Coloque elementos en áreas de cuadrícula usando la propiedad grid-area

Después de crear una plantilla de área para su contenedor de cuadrícula, como se muestra en el desafío anterior, puede colocar un elemento en su área personalizada haciendo referencia al nombre que le dio. Para hacer esto, usa la `grid-area` propiedad en un elemento como este:

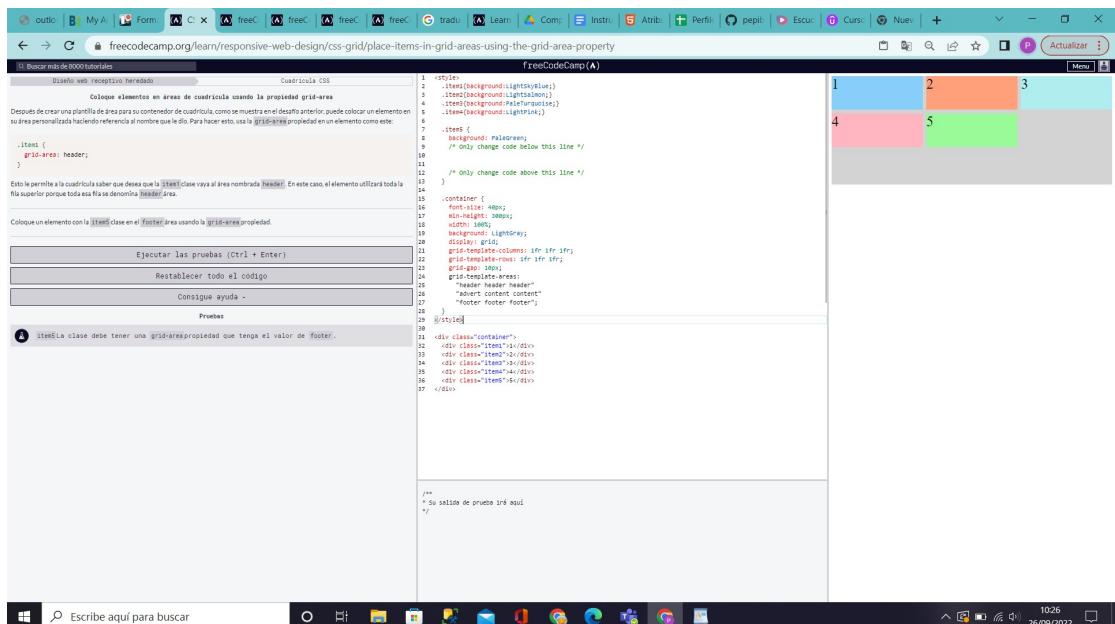
```
.item1 {

  grid-area: header;

}
```

Esto le permite a la cuadrícula saber que desea que la `item1` clase vaya al área nombrada `header`. En este caso, el elemento utilizará toda la fila superior porque toda esa fila se denomina `header` área.

Coloque un elemento con la `item5` clase en el `footer` área usando la `grid-area` propiedad.



```
1  #style{background:lightblue;}
```

```
2  .item1{background:lightblue;}
```

```
3  .item2{background:lightorange;}
```

```
4  .item3{background:paleturquoise;}
```

```
5  .item4{background:lightpink;}
```

```
6  .item5{background:palegreen;}
```

```
7  /* Only change code below this line */
```

```
8  .grid>.item1{grid-area:1fr 1fr 1fr;}
```

```
9  .grid>.item2{grid-area:1fr 1fr 1fr;}
```

```
10 .grid>.item3{grid-area:1fr 1fr 1fr;}
```

```
11 .grid>.item4{grid-area:1fr 1fr 1fr;}
```

```
12 .grid>.item5{grid-area:1fr 1fr 1fr;}
```

```
13 }
```

```
14 .container {
```

```
15   font-size: 40px;
```

```
16   min-height: 300px;
```

```
17   background-color:
```

```
18   background: lightgrey;
```

```
19   display: grid;
```

```
20   grid-template-columns: 1fr 1fr 1fr;
```

```
21   grid-template-rows: 1fr 1fr 1fr;
```

```
22 }
```

```
23 .grid>.item1{grid-area:1fr 1fr 1fr;}
```

```
24 .grid>.item2{grid-area:1fr 1fr 1fr;}
```

```
25 .grid>.item3{grid-area:1fr 1fr 1fr;}
```

```
26 .grid>.item4{grid-area:1fr 1fr 1fr;}
```

```
27 .grid>.item5{grid-area:1fr 1fr 1fr;}
```

```
28 }
```

```
29 
```

```
30 .grid>.item1{grid-area:1fr 1fr 1fr;}
```

```
31 .grid>.item2{grid-area:1fr 1fr 1fr;}
```

```
32 .grid>.item3{grid-area:1fr 1fr 1fr;}
```

```
33 .grid>.item4{grid-area:1fr 1fr 1fr;}
```

```
34 .grid>.item5{grid-area:1fr 1fr 1fr;}
```

```
35 }
```

```
36 }
```

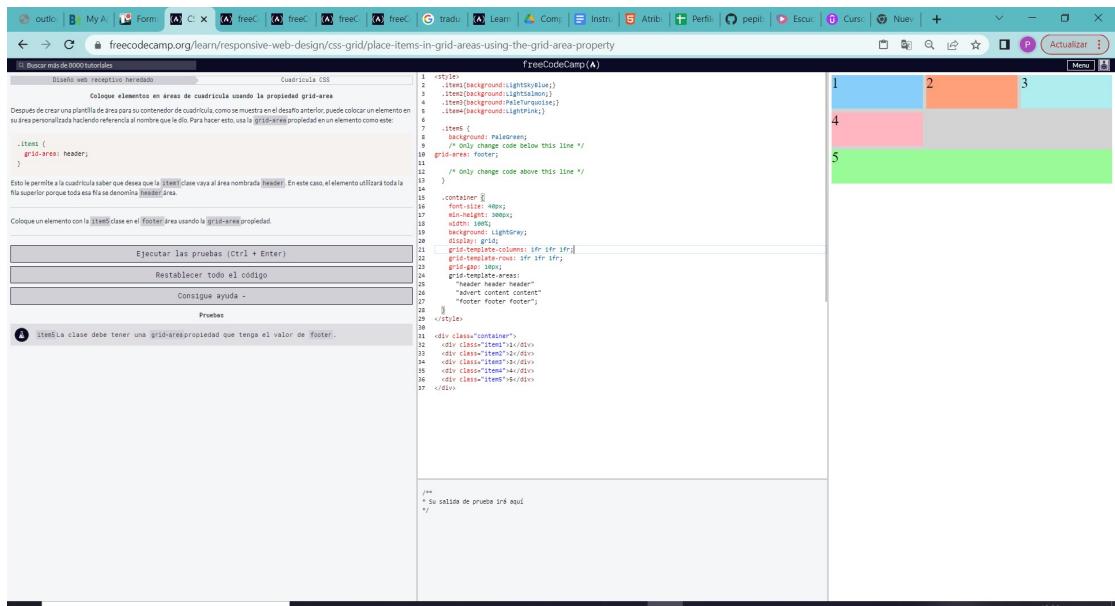
```
37 
```

```
38 /*
```

```
39 * Su salida de prueba irá aquí
```

```
40 */
```

Windows taskbar: Escribe aquí para buscar 1026 26/09/2022



```
1  #style{background:lightblue;}
```

```
2  .item1{background:lightblue;}
```

```
3  .item2{background:lightorange;}
```

```
4  .item3{background:paleturquoise;}
```

```
5  .item4{background:lightpink;}
```

```
6  .item5{background:palegreen;}
```

```
7  /* Only change code below this line */
```

```
8  .grid>.item1{grid-area:1fr 1fr 1fr;}
```

```
9  .grid>.item2{grid-area:1fr 1fr 1fr;}
```

```
10 .grid>.item3{grid-area:1fr 1fr 1fr;}
```

```
11 .grid>.item4{grid-area:1fr 1fr 1fr;}
```

```
12 .grid>.item5{grid-area:1fr 1fr 1fr;}
```

```
13 }
```

```
14 .container {
```

```
15   font-size: 40px;
```

```
16   min-height: 300px;
```

```
17   background-color:
```

```
18   background: lightgrey;
```

```
19   display: grid;
```

```
20   grid-template-columns: 1fr 1fr 1fr;
```

```
21   grid-template-rows: 1fr 1fr 1fr;
```

```
22 }
```

```
23 .grid>.item1{grid-area:1fr 1fr 1fr;}
```

```
24 .grid>.item2{grid-area:1fr 1fr 1fr;}
```

```
25 .grid>.item3{grid-area:1fr 1fr 1fr;}
```

```
26 .grid>.item4{grid-area:1fr 1fr 1fr;}
```

```
27 .grid>.item5{grid-area:1fr 1fr 1fr;}
```

```
28 }
```

```
29 
```

```
30 .grid>.item1{grid-area:1fr 1fr 1fr;}
```

```
31 .grid>.item2{grid-area:1fr 1fr 1fr;}
```

```
32 .grid>.item3{grid-area:1fr 1fr 1fr;}
```

```
33 .grid>.item4{grid-area:1fr 1fr 1fr;}
```

```
34 .grid>.item5{grid-area:1fr 1fr 1fr;}
```

```
35 }
```

```
36 }
```

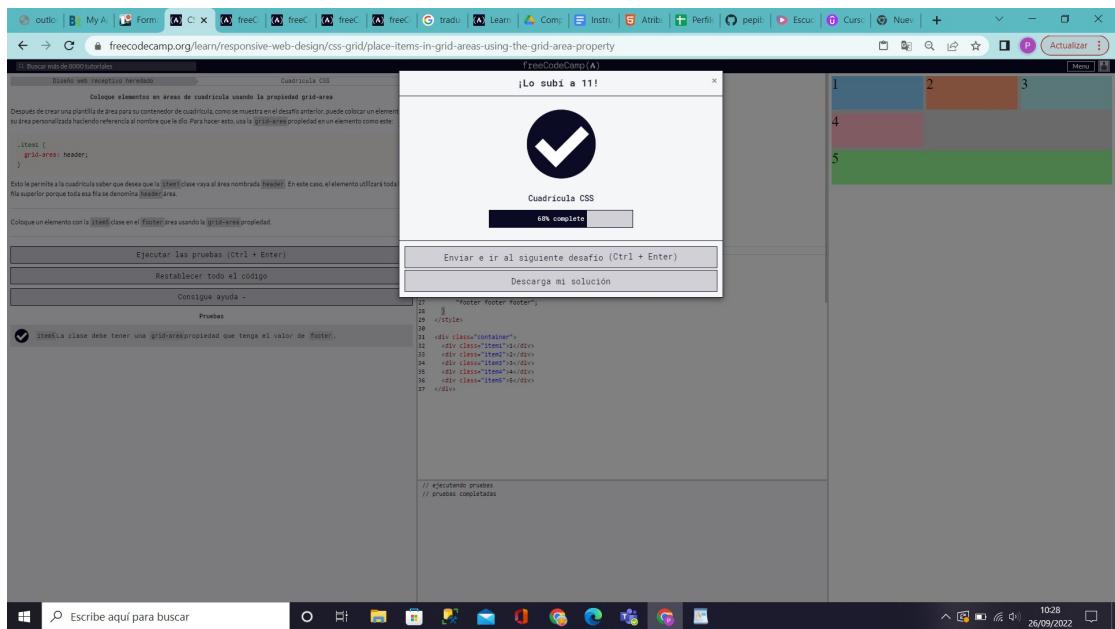
```
37 
```

```
38 /*
```

```
39 * Su salida de prueba irá aquí
```

```
40 */
```

Windows taskbar: Escribe aquí para buscar 1028 26/09/2022



<style>

```
.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5 {

    background: PaleGreen;

/* Only change code below this line */


```

grid-area: footer;

```
/* Only change code above this line */
```

```
}
```

```
.container {
```

```
font-size: 40px;

min-height: 300px;

width: 100%;

background: LightGray;

display: grid;
```

```

grid-template-columns: 1fr 1fr 1fr;
grid-template-rows: 1fr 1fr 1fr;
grid-gap: 10px;
grid-template-areas:
  "header header header"
  "advert content content"
  "footer footer footer";
}

</style>

<div class="container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
</div>

```

16. Usar grid-area sin crear una plantilla de áreas

La grid-area propiedad que aprendiste en el último desafío se puede usar de otra manera. Si su cuadrícula no tiene una plantilla de áreas a la que hacer referencia, puede crear un área sobre la marcha para colocar un elemento de esta manera:

```
item1 { grid-area: 1/1/2/4; }
```

Esto está utilizando los números de línea que aprendió anteriormente para definir dónde estará el área para este artículo. Los números en el ejemplo anterior representan estos valores:

grid-area: horizontal line to start at / vertical line to start at / horizontal line to end at / vertical line to end at;

Entonces, el elemento del ejemplo consumirá las filas entre las líneas 1 y 2, y las columnas entre las líneas 1 y 4.

Usando la grid-area propiedad, coloque el elemento con item5 clase entre la tercera y cuarta

Línea horizontal y entre la primera y cuarta línea vertical.

The screenshot shows a browser window displaying a CSS grid example from freeCodeCamp. The grid consists of 5 items labeled 1 through 5. Item 1 is in the top-left cell (row 1, column 1), item 2 is in the top-right cell (row 1, column 2), item 3 is in the bottom-left cell (row 2, column 1), item 4 is in the bottom-right cell (row 2, column 2), and item 5 is in the middle cell (row 1, column 2). The grid has 2 columns and 4 rows. The CSS code in the editor includes styling for the grid container and its items, such as background colors and grid-template-rows/columns.

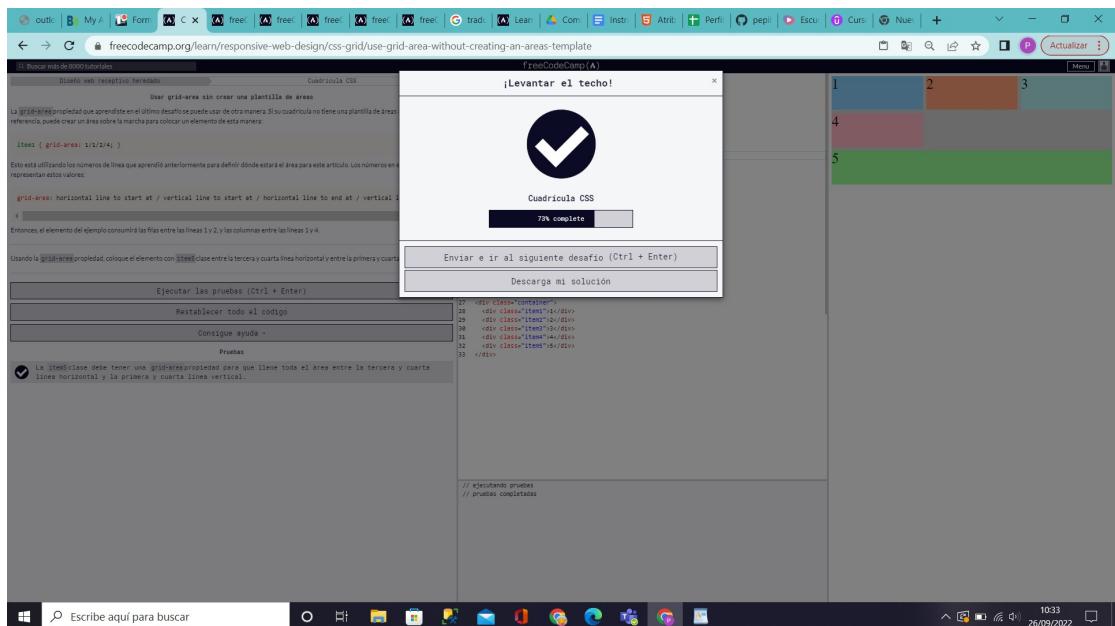
```
1 <div>
2   <item>background:lightblue;</item>
3   <item>background:lightorange;</item>
4   <item>background:lightcyan;</item>
5   <item>background:lightpink;</item>
6 
7   <item>
8     background: palegreen;
9     /* Only change code below this line */
10    ...
11  }
12 
13 .container {
14   font-size: 40px;
15   min-height: 100px;
16   width: 100px;
17   border: 1px solid black;
18   display: grid;
19   grid-template-columns: 1fr 1fr 1fr;
20   grid-template-rows: 1fr 1fr 1fr;
21   grid-gap: 10px;
22 }
23 
24 </style>
25 
26 <div>
27   <div><item></item></div>
28   <div><item></item></div>
29   <div><item></item></div>
30   <div><item></item></div>
31   <div><item></item></div>
32 </div>
33 
34 /**
35 * Su salida de prueba irá aquí
36 */
37
```

1030
26/09/2022

The screenshot shows a browser window displaying the same CSS grid example as the first one, but with a different layout. Item 1 is in the top-left cell (row 1, column 1), item 2 is in the top-right cell (row 1, column 2), item 3 is in the bottom-left cell (row 2, column 1), item 4 is in the bottom-right cell (row 2, column 2), and item 5 is in the middle cell (row 1, column 2). The grid has 2 columns and 4 rows. The CSS code in the editor is identical to the first screenshot.

```
1 <div>
2   <item>background:lightblue;</item>
3   <item>background:lightorange;</item>
4   <item>background:lightcyan;</item>
5   <item>background:lightpink;</item>
6 
7   <item>
8     background: palegreen;
9     /* Only change code below this line */
10    ...
11  }
12 
13 .container {
14   font-size: 40px;
15   min-height: 100px;
16   width: 100px;
17   border: 1px solid black;
18   display: grid;
19   grid-template-columns: 1fr 1fr 1fr;
20   grid-template-rows: 1fr 1fr 1fr;
21   grid-gap: 10px;
22 }
23 
24 </style>
25 
26 <div>
27   <div><item></item></div>
28   <div><item></item></div>
29   <div><item></item></div>
30   <div><item></item></div>
31   <div><item></item></div>
32 </div>
33 
34 /**
35 * Su salida de prueba irá aquí
36 */
37
```

1033
26/09/2022



style>

```
.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5 {

    background: PaleGreen;

grid-area: 3/1/4/4;

}

.container {

    font-size: 40px;

    min-height: 300px;

    width: 100%;

    background: LightGray;

    display: grid;

    grid-template-columns: 1fr 1fr 1fr;

    grid-template-rows: 1fr 1fr 1fr;
```

```

        grid-gap: 10px;
    }

</style>

<div class="container">

    <div class="item1">1</div>
    <div class="item2">2</div>
    <div class="item3">3</div>
    <div class="item4">4</div>
    <div class="item5">5</div>
</div>

```

17. Reduzca la repetición utilizando la función de repetición

Cuando usó `grid-template-columns` y `grid-template-rows` para definir la estructura de una cuadrícula, ingresó un valor para cada fila o columna que creó.

Digamos que quiere una cuadrícula con 100 filas de la misma altura. No es muy práctico insertar 100 valores individualmente. Afortunadamente, hay una manera mejor: usar la **`repeat`** función para especificar la cantidad de veces que desea que se repita su columna o fila, seguida de una coma y el valor que desea repetir.

Aquí hay un ejemplo que crearía la cuadrícula de 100 filas, cada fila con una altura de 50 px.

`grid-template-rows: repeat(100, 50px);`

También puede repetir varios valores con la función de repetición e insertar la función entre otros valores al definir una estructura de cuadrícula. Esto es lo que parece:

`grid-template-columns: repeat(2, 1fr 50px) 20px;`

Esto se traduce a:

`grid-template-columns: 1fr 50px 1fr 50px 20px;`

Nota: El `1fr 50px` se repite dos veces seguido de `20px`.

Úselo **`repeat`** para eliminar la repetición de la `grid-template-columns` propiedad.

The screenshot shows a browser window with the URL freeCodeCamp.org/learn/responsive-web-design/css-grid/reduce-repetition-using-the-repeat-function. The page content is about using the `grid-template-columns` and `grid-template-rows` properties to define a grid structure. It includes code snippets, explanations, and a preview of the grid layout.

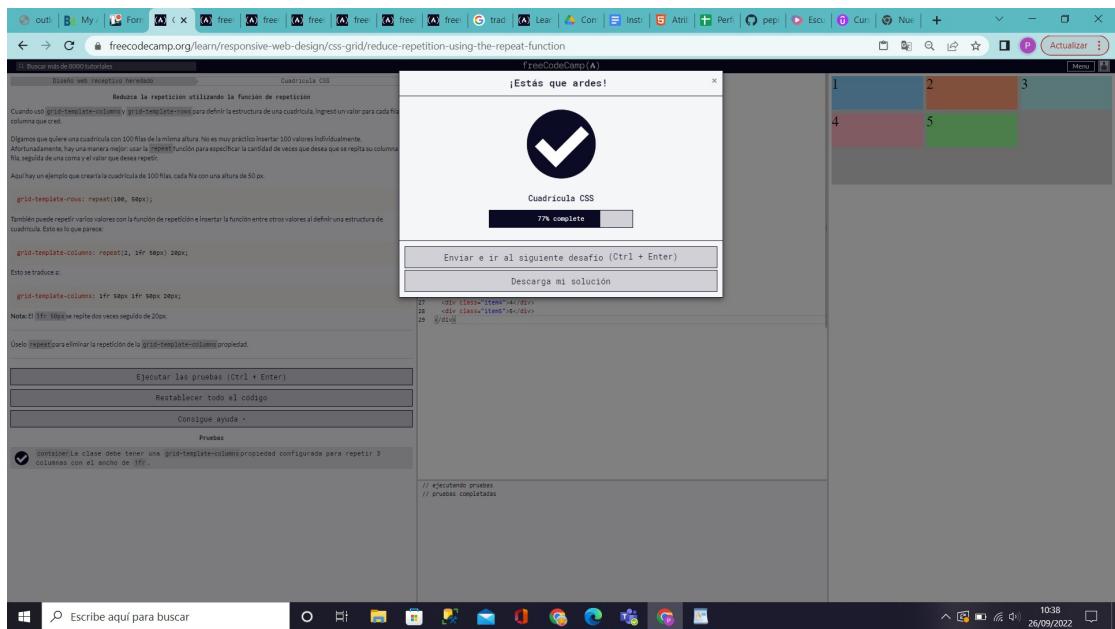
Container CSS:

```
1  :style{background:#f0f0f0}
2  2:item{background:#lightblue; width:100px; height:30px; margin:10px; border:1px solid black; font-size:40px; text-align:center; line-height:30px; color:#fff; }
3  3:item{background:#orange; width:100px; height:30px; margin:10px; border:1px solid black; font-size:40px; text-align:center; line-height:30px; color:#fff; }
4  4:item{background:#red; width:100px; height:30px; margin:10px; border:1px solid black; font-size:40px; text-align:center; line-height:30px; color:#fff; }
5  5:item{background:#green; width:100px; height:30px; margin:10px; border:1px solid black; font-size:40px; text-align:center; line-height:30px; color:#fff; }
```

Preview:

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | |

This screenshot is identical to the one above, showing the same CSS code and grid preview on the freeCodeCamp page.



```
<style>

.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5{background:PaleGreen;}

.container {

font-size: 40px;

min-height: 300px;

width: 100%;

background: LightGray;

display: grid;

/* Only change code below this line */

grid-template-columns: repeat(3, 1fr);

/* Only change code above this line */

grid-template-rows: 1fr 1fr 1fr;

grid-gap: 10px;
```

```
}

</style>

<div class="container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
</div>
```

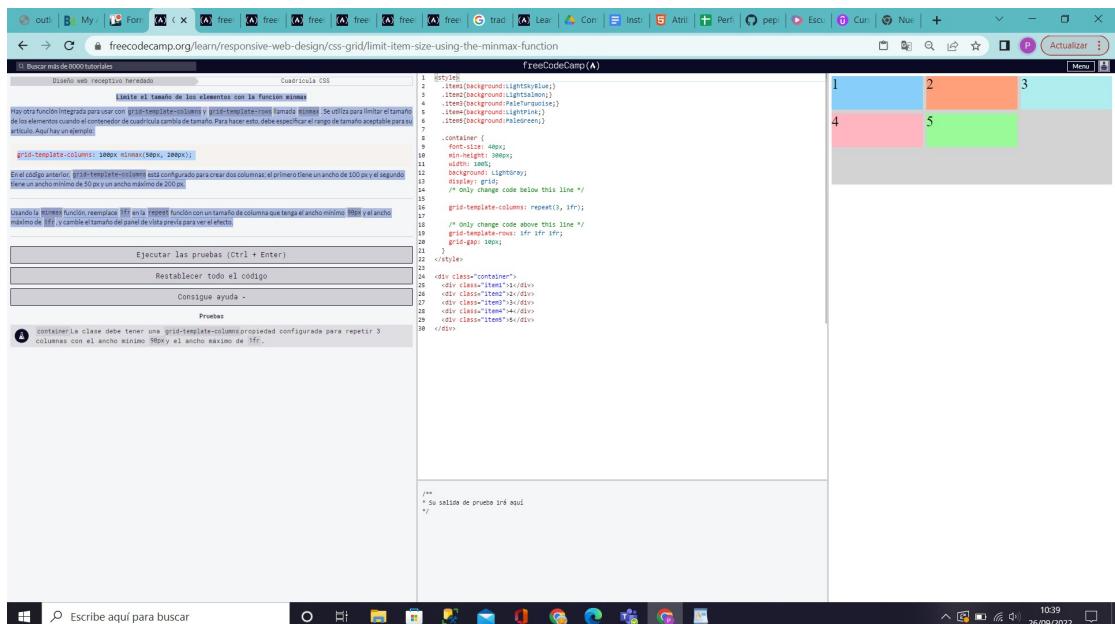
18. Limite el tamaño de los elementos con la función minmax

Hay otra función integrada para usar con grid-template-columns y grid-template-rows llamada minmax. Se utiliza para limitar el tamaño de los elementos cuando el contenedor de cuadrícula cambia de tamaño. Para hacer esto, debe especificar el rango de tamaño aceptable para su artículo. Aquí hay un ejemplo:

```
grid-template-columns: 100px minmax(50px, 200px);
```

En el código anterior, grid-template-columns está configurado para crear dos columnas; el primero tiene un ancho de 100 px y el segundo tiene un ancho mínimo de 50 px y un ancho máximo de 200 px.

Usando la minmax función, reemplace 1fr en la repeat función con un tamaño de columna que tenga el ancho mínimo 90px y el ancho máximo de 1fr, y cambie el tamaño del panel de vista previa para ver el efecto.



freeCodeCamp(A)

Buscador: freeCodeCamp.org/learn/responsive-web-design/css-grid/limit-item-size-using-the-minmax-function

Diseño web: heredado

Codificación CSS

```
1 1stChild {background: lightblue; }
2 2ndChild {background: lightorange; }
3 3rdChild {background: paleturquoise; }
4 4thChild {background: pink; }
5 5thChild {background: lightgreen; }
```

Límite el tamaño de los elementos con la función minmax

Hay otra función integrada para usar con `grid-template-columns` y `grid-template-rows`: `minmax()`. Se utiliza para limitar el tamaño de los elementos cuando el contenido de cuadrícula cambia de tamaño. Para hacer esto, debe especificar el rango de tamaño aceptable para su artículo. Aquí hay un ejemplo:

```
grid-template-columns: 100px; minmax(50px, 200px);
```

En el código anterior, `grid-template-columns` está configurado para crear dos columnas: el primero tiene un ancho de 100 px y el segundo tiene un ancho mínimo de 50 px y un ancho máximo de 200 px.

Usando la `minmax` función, reemplaza `ifr` en la `repeat` función con un tamaño de columna que tenga el ancho mínimo `50px` y el ancho máximo de `1fr`, y cambia el tamaño del panel de vista previa para ver el efecto.

```
grid-template-columns: repeat(3, 1fr);
```

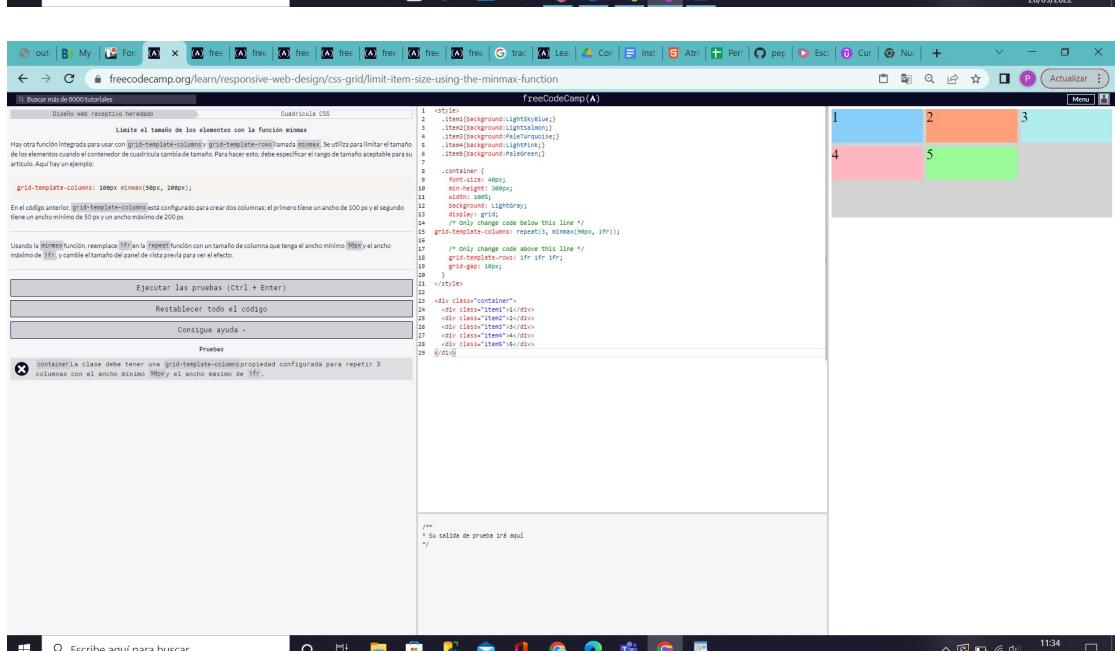
Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

containerLa clase debe tener una `grid-template-columns` propiedad configurada para repetir 3 columnas con el ancho mínimo `50px` y el ancho máximo de `1fr`.



freeCodeCamp(A)

Buscador: freeCodeCamp.org/learn/responsive-web-design/css-grid/limit-item-size-using-the-minmax-function

Diseño web: heredado

Codificación CSS

```
1 1stChild {background: lightblue; }
2 2ndChild {background: lightorange; }
3 3rdChild {background: paleturquoise; }
4 4thChild {background: pink; }
5 5thChild {background: lightgreen; }
```

Límite el tamaño de los elementos con la función minmax

Hay otra función integrada para usar con `grid-template-columns` y `grid-template-rows`: `minmax()`. Se utiliza para limitar el tamaño de los elementos cuando el contenido de cuadrícula cambia de tamaño. Para hacer esto, debe especificar el rango de tamaño aceptable para su artículo. Aquí hay un ejemplo:

```
grid-template-columns: 100px; minmax(50px, 200px);
```

En el código anterior, `grid-template-columns` está configurado para crear dos columnas: el primero tiene un ancho de 100 px y el segundo tiene un ancho mínimo de 50 px y un ancho máximo de 200 px.

Usando la `minmax` función, reemplaza `ifr` en la `repeat` función con un tamaño de columna que tenga el ancho mínimo `50px` y el ancho máximo de `1fr`, y cambia el tamaño del panel de vista previa para ver el efecto.

```
grid-template-columns: repeat(3, minmax(50px, 1fr));
```

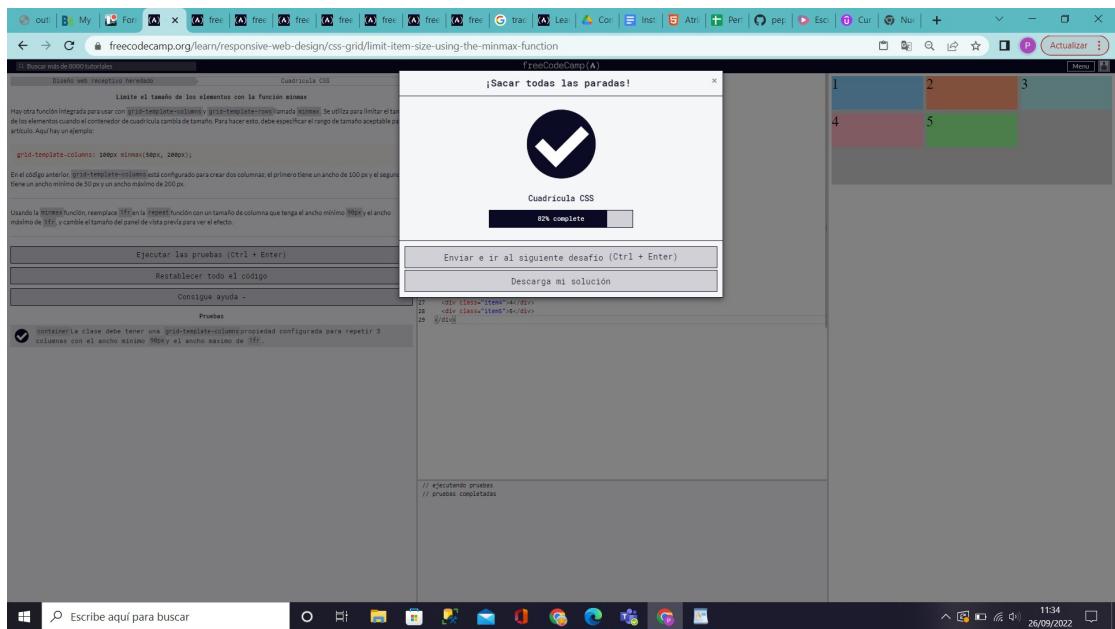
Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

containerLa clase debe tener una `grid-template-columns` propiedad configurada para repetir 3 columnas con el ancho mínimo `50px` y el ancho máximo de `1fr`.



```
<style>

.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5{background:PaleGreen;}

.container {

font-size: 40px;

min-height: 300px;

width: 100%;

background: LightGray;

display: grid;

/* Only change code below this line */

grid-template-columns: minmax(90px, 1fr);

/* Only change code above this line */

grid-template-rows: 1fr 1fr 1fr;

grid-gap: 10px;
```

```
}

</style>

<div class="container">
  <div class="item1">1</div>
  <div class="item2">2</div>
  <div class="item3">3</div>
  <div class="item4">4</div>
  <div class="item5">5</div>
</div>
```

19. Cree diseños flexibles usando autocompletar

La función de repetición viene con una opción llamada autocompletar . Esto le permite insertar automáticamente tantas filas o columnas del tamaño deseado como sea posible dependiendo del tamaño del contenedor. Puede crear diseños flexibles al combinarlos auto-fill con minmax, como este:

```
repeat(auto-fill, minmax(60px, 1fr));
```

Cuando el contenedor cambia de tamaño, esta configuración sigue insertando columnas de 60px y estirándolas hasta que pueda insertar otra. Nota: si su contenedor no puede acomodar todos sus artículos en una fila, los moverá hacia abajo a uno nuevo.

En la primera cuadrícula, use auto-fillwith repeatpara llenar la cuadrícula con columnas que tengan un ancho mínimo 60pxy máximo de 1fr. Luego cambie el tamaño de la vista previa para ver la acción de autocompletar.

freeCodeCamp(A)

```

    <style>
      .container {
        background-color: #f0f0f0;
        padding: 10px;
      }
      .item {
        width: 100px;
        height: 100px;
        background-color: lightgray;
        border: 1px solid black;
        margin: 5px;
      }
      .item1 {
        background-color: lightblue;
      }
      .item2 {
        background-color: lightorange;
      }
      .item3 {
        background-color: lightgreen;
      }
      .item4 {
        background-color: lightred;
      }
      .item5 {
        background-color: lightcyan;
      }
    </style>
    <div class="container">
      <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>
    </div>
  
```

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

Container la clase debe tener una `grid-template-columns` propieded con `repeat(auto-fit, minmax(100px, 1fr))`.

En la primera cuadrícula, use `grid-template-columns: repeat(3, minmax(100px, 1fr))` para llenar la cuadrícula con columnas que tengan un ancho mínimo de 100px y máximo de 1fr. Luego cambie el tamaño de la lista previa para ver la acción de autocompletar.

1 2 3
4 5

1 2 3
4 5

11:35 26/09/2022

freeCodeCamp(A)

```

    <style>
      .container {
        background-color: #f0f0f0;
        padding: 10px;
      }
      .item {
        width: 100px;
        height: 100px;
        background-color: lightgray;
        border: 1px solid black;
        margin: 5px;
      }
      .item1 {
        background-color: lightblue;
      }
      .item2 {
        background-color: lightorange;
      }
      .item3 {
        background-color: lightgreen;
      }
      .item4 {
        background-color: lightred;
      }
      .item5 {
        background-color: lightcyan;
      }
    </style>
    <div class="container">
      <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>
    </div>
  
```

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

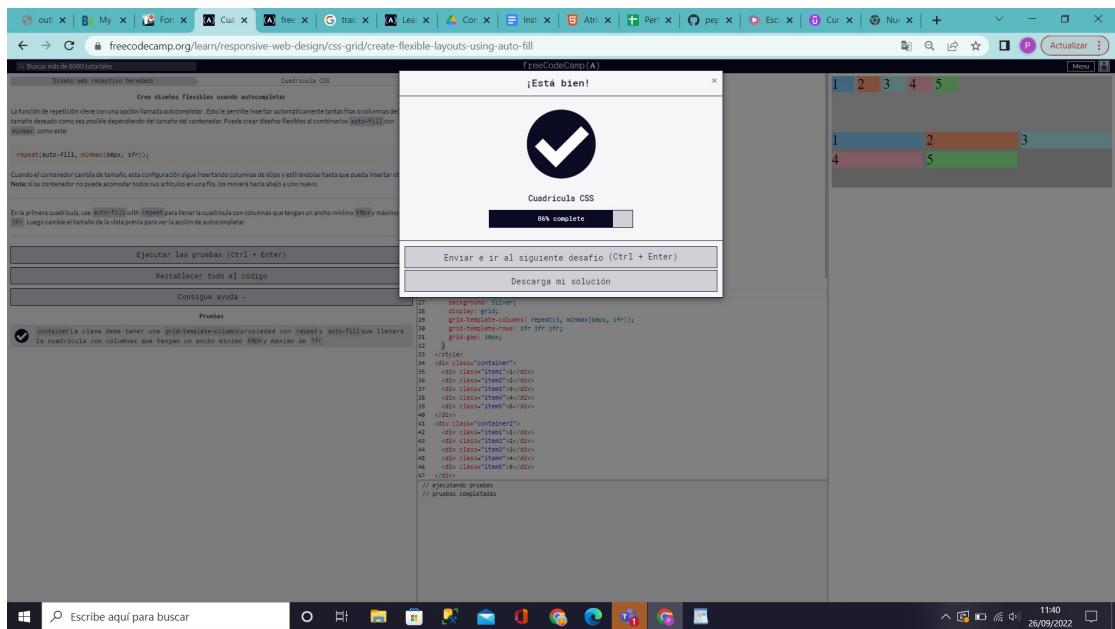
Container la clase debe tener una `grid-template-columns` propieded con `repeat(auto-fit, minmax(100px, 1fr))`.

En la primera cuadrícula, use `grid-template-columns: repeat(3, minmax(100px, 1fr))` para llenar la cuadrícula con columnas que tengan un ancho mínimo de 100px y máximo de 1fr. Luego cambie el tamaño de la lista previa para ver la acción de autocompletar.

1 2 3 4 5
1 2 3
4 5

1 2 3 4 5
1 2 3
4 5

11:39 26/09/2022



Antes

grid-template-columns: repeat(3, minmax(60px, 1fr));

Después

grid-template-columns: repeat(auto-fill, minmax(60px, 1fr));

```
<style>
.item1{background:LightSkyBlue;}
.item2{background:LightSalmon;}
.item3{background:PaleTurquoise;}
.item4{background:LightPink;}
.item5{background:PaleGreen;}
.container {
    font-size: 40px;
    min-height: 100px;
    width: 100%;
    background: LightGray;
```

```

display: grid;

/* Only change code below this line */

grid-template-columns: repeat(auto-fill, minmax(60px, 1fr));

/* Only change code above this line */

grid-template-rows: 1fr 1fr 1fr;
grid-gap: 10px;
}

.container2 {
font-size: 40px;
min-height: 100px;
width: 100%;
background: Silver;
display: grid;
grid-template-columns: repeat(3, minmax(60px, 1fr));
grid-template-rows: 1fr 1fr 1fr;
grid-gap: 10px;
}

</style>

<div class="container">
<div class="item1">1</div>
<div class="item2">2</div>
<div class="item3">3</div>
<div class="item4">4</div>
<div class="item5">5</div>
</div>

<div class="container2">
<div class="item1">1</div>
<div class="item2">2</div>

```

```

<div class="item3">3</div>
<div class="item4">4</div>
<div class="item5">5</div>
</div>

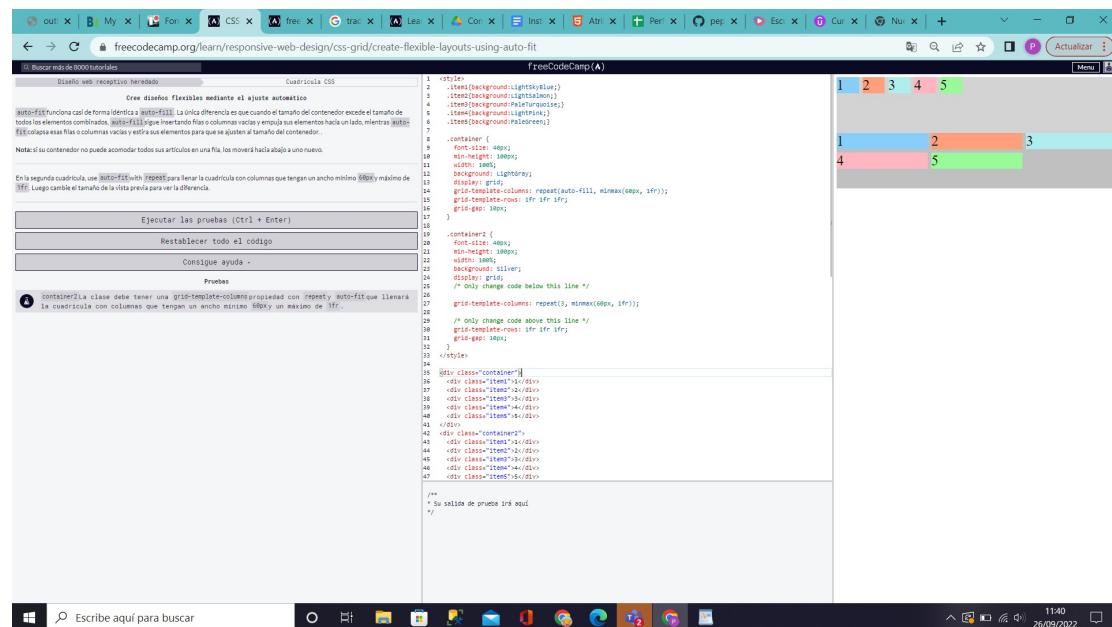
```

20. Cree diseños flexibles mediante el ajuste automático

auto-fitfunciona casi de forma idéntica a auto-fill. La única diferencia es que cuando el tamaño del contenedor excede el tamaño de todos los elementos combinados, auto-fillsigue insertando filas o columnas vacías y empuja sus elementos hacia un lado, mientras auto-fitcolapsa esas filas o columnas vacías y estira sus elementos para que se ajusten al tamaño del contenedor. .

Nota: si su contenedor no puede acomodar todos sus artículos en una fila, los moverá hacia abajo a uno nuevo.

En la segunda cuadrícula, use auto-fitwith repeatpara llenar la cuadrícula con columnas que tengan un ancho mínimo 60px y máximo de 1fr. Luego cambie el tamaño de la vista previa para ver la diferencia.



```

.out x | B: My x | F: x | CSS x | freecamp x | G: tra x | Lea x | Cor x | Inst x | ATri x | Per x | pep x | Esc x | Cui x | Nu x | + | - | □ | Actualizar



Diseño web: reactividad heredada
Cuadrícula CSS



```

<div>
 <div class="item1">1</div>
 <div class="item2">2</div>
 <div class="item3">3</div>
 <div class="item4">4</div>
 <div class="item5">5</div>
</div>

```



Cuadrícula CSS



Cree diseños flexibles mediante el ajuste automático



auto-fitfunciona casi de forma idéntica a auto-fill. La única diferencia es que cuando el tamaño del contenedor excede el tamaño de todos los elementos combinados, auto-fitsigue insertando filas o columnas vacías y empuja sus elementos hacia un lado, mientras auto-fitcolapsa esas filas o columnas vacías y estira sus elementos para que se ajusten al tamaño del contenedor.



Notas: si su contenedor no puede acomodar todos sus artículos en una fila, los moverá hacia abajo a uno nuevo.



En la segunda cuadrícula, use grid-template-columns: repeat(3, minmax(60px, 1fr))para llenar la cuadrícula con columnas que tengan un ancho mínimo 60px y máximo de 1fr. Luego cambie el tamaño de la vista previa para ver la diferencia.



Ejecutar las pruebas (Ctrl + Enter)



Restablecer todo el código



Consigue ayuda -



Pruebas



containerLa clase debe tener una grid-template-columns propiedad con repeat(3, minmax(60px, 1fr)) para llenar la cuadrícula con columnas que tengan un ancho mínimo 60px y un máximo de 1fr.



```

1 1px solid black;
2 .item1{background:lightskyblue;}
3 .item2{background:lightblueyellow;}
4 .item3{background:lightpink;}
5 .item4{background:lightgreen;}
6 .item5{background:lightcoral;}
7
8 .container {
9 font-size: 4em;
10 min-height: 100vh;
11 background-color: #f0f0f0;
12 background: repeating-linear-gradient(45deg, transparent, transparent 2px, #f0f0f0 2px, #f0f0f0 4px);
13 display: grid;
14 grid-template-columns: repeat(auto-fit, minmax(60px, 1fr));
15 grid-template-rows: 1fr 1fr 1fr;
16 grid-gap: 10px;
17}
18
19 .container {
20 font-size: 4em;
21 min-height: 100vh;
22 background-color: #f0f0f0;
23 background: repeating-linear-gradient(45deg, transparent, transparent 2px, #f0f0f0 2px, #f0f0f0 4px);
24 display: grid;
25 grid-template-columns: repeat(3, minmax(60px, 1fr));
26 grid-template-rows: repeat(2, minmax(60px, 1fr));
27 grid-gap: 10px;
28}
29 /* Only change code above this line */
30 /* Only change code above this line */
31 /* Only change code above this line */
32 /* Only change code above this line */
33 /* Only change code above this line */
34
35 <div>
36 <div>1</div>
37 <div>2</div>
38 <div>3</div>
39 <div>4</div>
40 <div>5</div>
41 </div>
42 <div><!--Container--></div>
43 <div><!--Container--></div>
44 <div><!--Container--></div>
45 <div><!--Container--></div>
46 <div><!--Container--></div>
47 <div><!--Container--></div>
48
49 /**
50 * Su salida de prueba irá aquí
51 */

```


```

outlook | B My API | Formas | CSS x freeCodeCamp | traducir | Learn | Comp | Instruc | Atribu | Perfil | pepi | Escuch | Cursos | Nueva | +

Buscador de código heredado Cuadrícula CSS

Crea cuadrículas flexibles mediante el ajuste automático

Nota: si su contenedor no puede acomodar todos sus artículos en una fila, los moverá hacia abajo a un nuevo.

En la segunda cuadrícula, use `grid-template-columns: repeat(3, minmax(60px, 1fr))` para llenar la cuadrícula con columnas que tengan un ancho mínimo 60px y máximo de 1fr. Luego cambie el tamaño de la vista previa para ver la diferencia.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consegue ayuda -

Proches

containerLa clase debe tener una `grid-template-columns:repeat(3, auto-fit, auto-fit)` que llenará la cuadrícula con columnas que tengan un ancho mínimo 60px y un máximo de 1fr.

```

1 <style>
2   .cuadricula {
3     background-color: #f0f0f0;
4     grid-template-columns: 1fr 1fr 1fr;
5     grid-template-rows: 1fr 1fr 1fr;
6     grid-template-rows: repeat(auto-fit, minmax(60px, 1fr));
7     grid-template-columns: repeat(auto-fit, minmax(60px, 1fr));
8   }
9   .cuadricula .contenedores {
10    font-size: 40px;
11    height: 100px;
12    width: 100px;
13    background-color: lightgray;
14    border: 1px solid black;
15    display: grid;
16    justify-content: center;
17    align-items: center;
18  }
19  .cuadricula .contenedores .item1 {
20    background-color: red;
21  }
22  .cuadricula .contenedores .item2 {
23    background-color: green;
24  }
25  /* Only change code above this line */
26  .cuadricula .contenedores .item3 {
27    background-color: blue;
28  }
29  .cuadricula .contenedores .item4 {
30    background-color: orange;
31  }
32  .cuadricula .contenedores .item5 {
33    background-color: yellow;
34  }
35  .cuadricula .contenedores .item6 {
36    background-color: purple;
37  }
38  .cuadricula .contenedores .item7 {
39    background-color: pink;
40  }
41  .cuadricula .contenedores .item8 {
42    background-color: teal;
43  }
44  .cuadricula .contenedores .item9 {
45    background-color: lightblue;
46  }
47  .cuadricula .contenedores .item10 {
48    background-color: lightgreen;
49  }
50  .cuadricula .contenedores .item11 {
51    background-color: lightpurple;
52  }
53  .cuadricula .contenedores .item12 {
54    background-color: lightpink;
55  }
56  /* Salida de prueba irá aquí
57 */
</style>
```

1 2 3 4 5

1 2 3 4 5

11:57 26/09/2022

Escribe aquí para buscar

outlook | B My API | Formas | CSS x freeCodeCamp | traducir | Learn | Comp | Instruc | Atribu | Perfil | pepi | Escuch | Cursos | Nueva | +

Buscador de código heredado Cuadrícula CSS

Crea cuadrículas flexibles mediante el ajuste automático

Nota: si su contenedor no puede acomodar todos sus artículos en una fila, los moverá hacia abajo a un nuevo.

En la segunda cuadrícula, use `grid-template-columns: repeat(3, minmax(60px, 1fr))` para llenar la cuadrícula con columnas que tengan un ancho mínimo 60px y máximo de 1fr. Luego cambie el tamaño de la vista previa para ver la diferencia.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consegue ayuda -

Proches

containerLa clase debe tener una `grid-template-columns:repeat(3, auto-fit, auto-fit)` que llenará la cuadrícula con columnas que tengan un ancho mínimo 60px y un máximo de 1fr.

Solución completa

Enviar e ir al siguiente desafío (Ctrl + Enter)

Descarga mi solución

```

1 <style>
2   .cuadricula {
3     background-color: #f0f0f0;
4     grid-template-columns: 1fr 1fr 1fr;
5     grid-template-rows: 1fr 1fr 1fr;
6     grid-template-rows: repeat(auto-fit, minmax(60px, 1fr));
7     grid-template-columns: repeat(auto-fit, minmax(60px, 1fr));
8   }
9   .cuadricula .contenedores {
10    font-size: 40px;
11    height: 100px;
12    width: 100px;
13    background-color: lightgray;
14    border: 1px solid black;
15    display: grid;
16    justify-content: center;
17    align-items: center;
18  }
19  .cuadricula .contenedores .item1 {
20    background-color: red;
21  }
22  .cuadricula .contenedores .item2 {
23    background-color: green;
24  }
25  /* Only change code above this line */
26  .cuadricula .contenedores .item3 {
27    background-color: blue;
28  }
29  .cuadricula .contenedores .item4 {
30    background-color: orange;
31  }
32  .cuadricula .contenedores .item5 {
33    background-color: yellow;
34  }
35  .cuadricula .contenedores .item6 {
36    background-color: purple;
37  }
38  .cuadricula .contenedores .item7 {
39    background-color: pink;
40  }
41  .cuadricula .contenedores .item8 {
42    background-color: teal;
43  }
44  .cuadricula .contenedores .item9 {
45    background-color: lightblue;
46  }
47  .cuadricula .contenedores .item10 {
48    background-color: lightgreen;
49  }
50  .cuadricula .contenedores .item11 {
51    background-color: lightpurple;
52  }
53  .cuadricula .contenedores .item12 {
54    background-color: lightpink;
55  }
56  /* Salida de prueba irá aquí
57 */
</style>
```

1 2 3 4 5

1 2 3 4 5

11:57 26/09/2022

Antes:

`grid-template-columns: repeat(3, minmax(60px, 1fr));`

Después:

`grid-template-columns: repeat(auto-fit, minmax(60px, 1fr));`

<style>

```

.item1{background:LightSkyBlue;}

.item2{background:LightSalmon;}

.item3{background:PaleTurquoise;}

.item4{background:LightPink;}

.item5{background:PaleGreen;}

.container {
    font-size: 40px;
    min-height: 100px;
    width: 100%;
    background: LightGray;
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(60px, 1fr));
    grid-template-rows: 1fr 1fr 1fr;
    grid-gap: 10px;
}

.container2 {
    font-size: 40px;
    min-height: 100px;
    width: 100%;
    background: Silver;
    display: grid;
    /* Only change code below this line */
grid-template-columns: repeat(auto-fit, minmax(60px, 1fr));
    /* Only change code above this line */
    grid-template-rows: 1fr 1fr 1fr;
    grid-gap: 10px;
}

</style>

```

```
<div class="container">  
  <div class="item1">1</div>  
  <div class="item2">2</div>  
  <div class="item3">3</div>  
  <div class="item4">4</div>  
  <div class="item5">5</div>  
</div>  
  
<div class="container2">  
  <div class="item1">1</div>  
  <div class="item2">2</div>  
  <div class="item3">3</div>  
  <div class="item4">4</div>  
  <div class="item5">5</div>  
</div>
```

21. Use consultas de medios para crear diseños receptivos

CSS Grid puede ser una manera fácil de hacer que su sitio sea más receptivo mediante el uso de consultas de medios para reorganizar las áreas de la cuadrícula, cambiar las dimensiones de una cuadrícula y reorganizar la ubicación de los elementos.

En la vista previa, cuando el ancho de la ventana gráfica es de 300 px o más, el número de columnas cambia de 1 a 2. El área de publicidad ocupa la columna izquierda por completo.

Cuando el ancho de la ventana gráfica sea 400pxo más, haga que el área del encabezado ocupe la fila superior por completo y el área del pie de página ocupe la fila inferior por completo.

```
freeCodeCamp(A)
```

```
Diseño web / responsive heredado Cuestionario CSS
```

```
Use consultas de medios para crear diseños receptivos
```

```
CSS Grid puede ser una manera fácil de hacer que su sitio sea más receptivo mediante el uso de consultas de medios para reorganizar los áreas de la cuadrícula, cambiar las dimensiones de una cuadrícula y reorganizar la ubicación de los elementos.
```

```
En la vista previa, cuando el ancho de la ventana gráfica es de 300 px o más, el número de columnas cambia de 1 a 2. El área de publicidad ocupa la columna izquierda por completo.
```

```
Cuando el ancho de la ventana gráfica sea 480px o más, haga que el área del encabezado ocupe la fila superior por completo y el área del pie de página ocupe la fila inferior por completo.
```

```
Ejecutar las pruebas (Ctrl + Enter)
```

```
Restablecer todo el código
```

```
Consigue ayuda -
```

```
Problemas
```

```
Cuando la ventana gráfica es 480px o más, contiene la clase debe tener una grid-template-columns propiedad en la que las áreas de encabezado y pie de página ocupen las filas superior e inferior respectivamente y el anuncio y el contenido ocupen las columnas izquierda y derecha de la fila central.
```

```
.header { background-color: orange; height: 50px; }
```

```
.content { background-color: lightblue; min-height: 300px; }
```

```
.advert { background-color: lightcoral; height: 50px; }
```

```
.footer { background-color: pink; height: 50px; }
```

```
.container { font-size: 1.5em; width: 100%; background-color: lightgray; display: grid; grid-template-columns: 1fr; grid-template-rows: 100px auto 100px auto 100px auto 1fr auto; grid-gap: 10px; grid-template-areas: "header" "advert" "content" "advert" "content" "footer"; }
```

```
/* @media (min-width: 300px){ .container { grid-template-columns: auto 1.5em; grid-template-rows: auto 1fr auto; grid-template-areas: "header" "advert header" "advert content" "advert footer"; } }
```

```
/* - salida de prueba irá aquí
```

11:58
26/09/2022

```
freeCodeCamp(A)
```

```
Diseño web / responsive heredado Cuestionario CSS
```

```
Use consultas de medios para crear diseños receptivos
```

```
CSS Grid puede ser una manera fácil de hacer que su sitio sea más receptivo mediante el uso de consultas de medios para reorganizar los áreas de la cuadrícula, cambiar las dimensiones de una cuadrícula y reorganizar la ubicación de los elementos.
```

```
En la vista previa, cuando el ancho de la ventana gráfica es de 300 px o más, el número de columnas cambia de 1 a 2. El área de publicidad ocupa la columna izquierda por completo.
```

```
Cuando el ancho de la ventana gráfica sea 480px o más, haga que el área del encabezado ocupe la fila superior por completo y el área del pie de página ocupe la fila inferior por completo.
```

```
Ejecutar las pruebas (Ctrl + Enter)
```

```
Restablecer todo el código
```

```
Consigue ayuda -
```

```
Problemas
```

```
Cuando la ventana gráfica es 480px o más, contiene la clase debe tener una grid-template-columns propiedad en la que las áreas de encabezado y pie de página ocupen las filas superior e inferior respectivamente y el anuncio y el contenido ocupen las columnas izquierda y derecha de la fila central.
```

```
.header { background-color: orange; height: 50px; }
```

```
.content { background-color: lightblue; min-height: 300px; }
```

```
.advert { background-color: lightcoral; height: 50px; }
```

```
.footer { background-color: pink; height: 50px; }
```

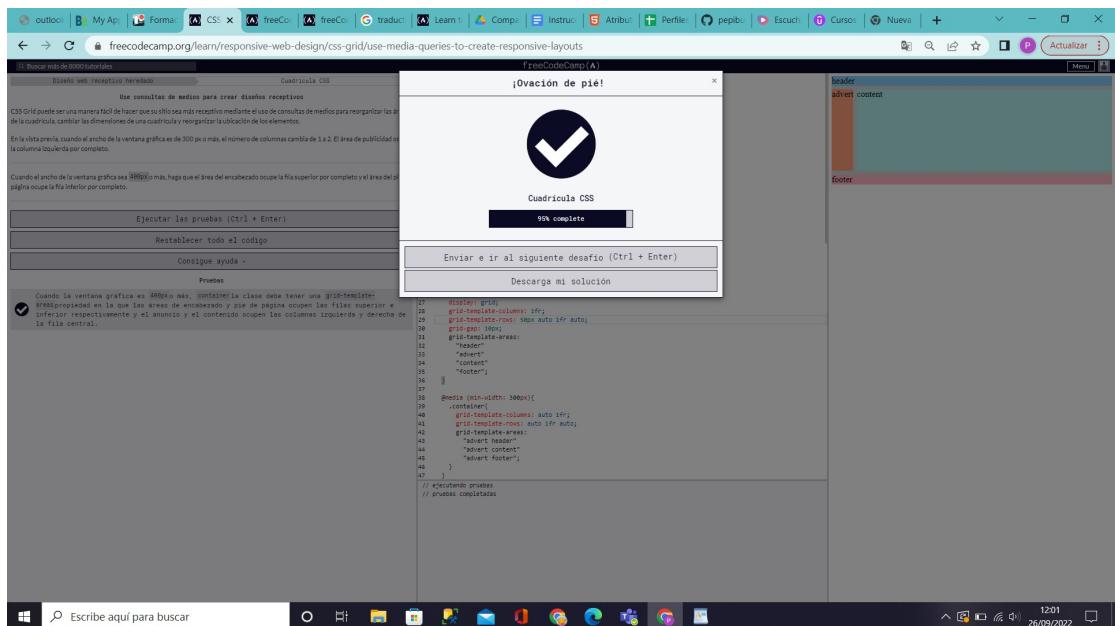
```
.container { font-size: 1.5em; width: 100%; background-color: lightgray; display: grid; grid-template-columns: 1fr; grid-template-rows: 100px auto 100px auto 100px auto 1fr auto; grid-gap: 10px; grid-template-areas: "header" "advert" "content" "advert" "content" "footer"; }
```

```
/* @media (min-width: 300px){ .container { grid-template-columns: auto 1.5em; grid-template-rows: auto 1fr auto; grid-template-areas: "header" "advert header" "advert content" "advert footer"; } }
```

```
// ejecutando pruebas
```

```
// pruebas completadas
```

12:01
26/09/2022



```

<style>

.item1 {
    background: LightSkyBlue;
    grid-area: header;
}

.item2 {
    background: LightSalmon;
    grid-area: advert;
}

.item3 {
    background: PaleTurquoise;
    grid-area: content;
}

.item4 {
    background: lightpink;
    grid-area: footer;
}

```

```
.container {  
    font-size: 1.5em;  
    min-height: 300px;  
    width: 100%;  
    background: LightGray;  
    display: grid;  
    grid-template-columns: 1fr;  
    grid-template-rows: 50px auto 1fr auto;  
    grid-gap: 10px;  
    grid-template-areas:  
        "header"  
        "advert"  
        "content"  
        "footer";  
}  
  
@media (min-width: 300px){  
    .container{  
        grid-template-columns: auto 1fr;  
        grid-template-rows: auto 1fr auto;  
        grid-template-areas:  
            "advert header"  
            "advert content"  
            "advert footer";  
    }  
}  
  
@media (min-width: 400px){  
    .container{  
        grid-template-areas:  
    }  
}
```

```

/* Only change code below this line */

"header header"

"advert content"

"footer footer";

/* Only change code above this line */

}

}

</style>

<div class="container">

<div class="item1">header</div>

<div class="item2">advert</div>

<div class="item3">content</div>

<div class="item4">footer</div>

</div>

```

22. Crear cuadrículas dentro de cuadrículas

Convertir un elemento en una grilla solo afecta el comportamiento de sus descendientes directos. Entonces, al convertir un descendiente directo en una cuadrícula, tienes una cuadrícula dentro de una cuadrícula.

Por ejemplo, al establecer las propiedades `display` y `grid-template-columns` del elemento con la `item3` clase, crea una cuadrícula dentro de su cuadrícula.

Convierta el elemento con la **item3** clase en una **cuadrícula** con **dos columnas** con un ancho de **auto** y **1 fr** usando **display** y **grid-template-columns**.

outlook My API Formas CSS freeCo freeCo traduc Learn Comp Instru Atribu Perfil pepi Escuch Cursos Nueva Actualizar

Buscador de resultados heredados

Cuadrícula CSS

Crear cuadrículas dentro de cuadrículas

Convertir un elemento en una grilla solo afecta el comportamiento de sus descendientes directos. Entonces, al convertir un descendiente directo en una cuadrícula, tienes una cuadrícula dentro de una cuadrícula.

Por ejemplo, al establecer las propiedades `display: grid; grid-template-columns: auto 1fr;` y `grid-template-columns: auto 1fr;` del elemento con la `.item` clase, crea una cuadrícula dentro de su cuadrícula.

Convierte el elemento con la `.item` clase en una cuadrícula con dos columnas con un ancho de `1fr` usando `grid-template-columns: auto 1fr;` y `grid-template-columns: auto 1fr;`.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

Pruebas

item La clase debe tener una `grid-template-columns` propiedad con `auto 1fr` como valores.

item La clase debe tener una `display` propiedad con el valor de `grid`.

```
1 style
2 .cuadricula {
3   font-size: 1.5em;
4   min-height: 300px;
5   background-color: lightgray;
6   background: lightgray;
7   display: grid;
8   grid-template-columns: auto 1fr;
9   grid-template-rows: auto 1fr;
10  grid-template-areas:
11    "advert header"
12    "advert content"
13    "advert footer";
14 }
15 .item {
16   background: lightskyblue;
17   grid-area: header;
18 }
19 .item {
20   background: lightsalmon;
21   grid-area: content;
22 }
23 .item {
24   background: lightblue;
25   grid-area: footer;
26 }
27 /* Only change code above this line */
28 }
29 .item {
30   background: lightpink;
31   grid-area: footer;
32 }
33 .itemone {
34   background: palegreen;
35 }
36 .itemtwo {
37   background: blanchedalmond;
38 }
39 .itemthree {
40   background: palegreen;
41 }
42 .itemfour {
43   background: blanchedalmond;
44 }
45 .itemfive {
46   background: blanchedalmond;
47 }
48 /* Only change code below this line */
49 }
50 .item {
51   background: lightpink;
52   grid-area: content;
53 }
54 /* Only change code below this line */
55 }
56 display: grid;
57 grid-template-columns: auto 1fr;
58 grid-template-rows: auto 1fr;
59 grid-template-areas:
60   "advert header"
61   "paragraph1 paragraph2"
62   "footer";
63 
```

12:02 26/09/2022

outlook My API Formas CSS freeCo freeCo traduc Learn Comp Instru Atribu Perfil pepi Escuch Cursos Nueva Actualizar

Buscador de resultados heredados

Cuadrícula CSS

Crear cuadrículas dentro de cuadrículas

Convertir un elemento en una grilla solo afecta el comportamiento de sus descendientes directos. Entonces, al convertir un descendiente directo en una cuadrícula, tienes una cuadrícula dentro de una cuadrícula.

Por ejemplo, al establecer las propiedades `display: grid; grid-template-columns: auto 1fr;` y `grid-template-columns: auto 1fr;` del elemento con la `.item` clase, crea una cuadrícula dentro de su cuadrícula.

Convierte el elemento con la `.item` clase en una cuadrícula con dos columnas con un ancho de `1fr` usando `grid-template-columns: auto 1fr;` y `grid-template-columns: auto 1fr;`.

Ejecutar las pruebas (Ctrl + Enter)

Restablecer todo el código

Consigue ayuda -

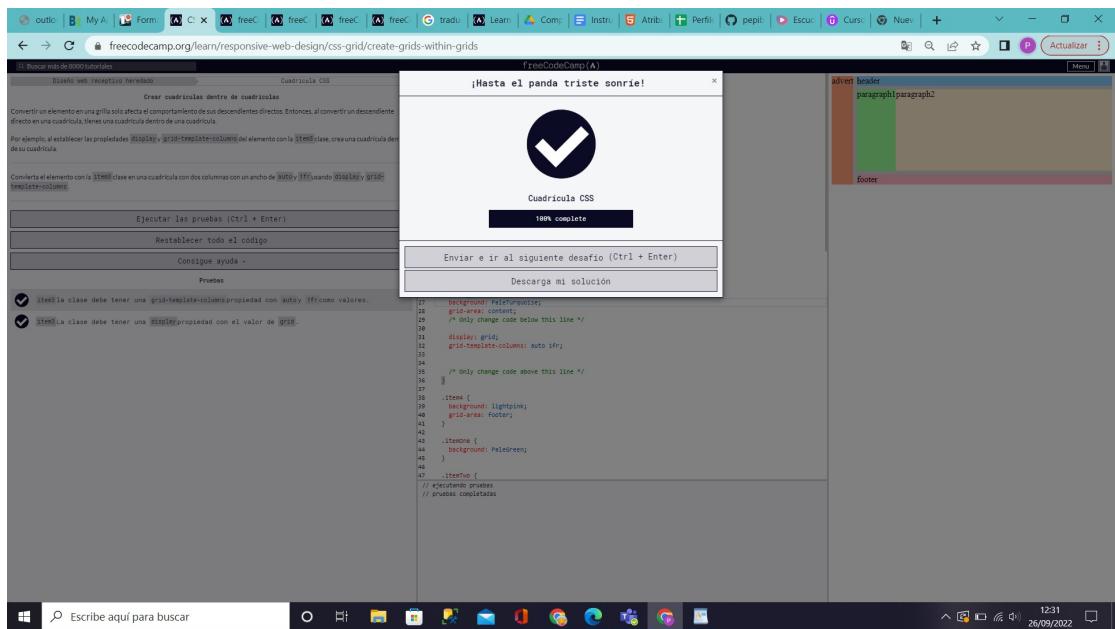
Pruebas

item La clase debe tener una `grid-template-columns` propiedad con `auto 1fr` como valores.

item La clase debe tener una `display` propiedad con el valor de `grid`.

```
1 style
2 .cuadricula {
3   font-size: 1.5em;
4   min-height: 300px;
5   background-color: lightgray;
6   background: lightgray;
7   display: grid;
8   grid-template-columns: auto 1fr;
9   grid-template-rows: auto 1fr;
10  grid-template-areas:
11    "advert header"
12    "advert content"
13    "advert footer";
14 }
15 .item {
16   background: lightskyblue;
17   grid-area: header;
18 }
19 .item {
20   background: lightsalmon;
21   grid-area: content;
22 }
23 .item {
24   background: lightblue;
25   grid-area: footer;
26 }
27 /* Only change code above this line */
28 }
29 .item {
30   background: lightpink;
31   grid-area: footer;
32 }
33 .itemone {
34   background: palegreen;
35 }
36 .itemtwo {
37   background: blanchedalmond;
38 }
39 .itemthree {
40   background: palegreen;
41 }
42 .itemfour {
43   background: blanchedalmond;
44 }
45 .itemfive {
46   background: blanchedalmond;
47 }
48 /* Only change code below this line */
49 }
50 .item {
51   background: lightpink;
52   grid-area: content;
53 }
54 /* Only change code below this line */
55 }
56 display: grid;
57 grid-template-columns: auto 1fr;
58 grid-template-rows: auto 1fr;
59 grid-template-areas:
60   "advert header"
61   "paragraph1 paragraph2"
62   "footer";
63 
```

12:30 26/09/2022



<style>

```
.container {
    font-size: 1.5em;
    min-height: 300px;
    width: 100%;
    background: LightGray;
    display: grid;
    grid-template-columns: auto 1fr;
    grid-template-rows: auto 1fr auto;
    grid-gap: 10px;
    grid-template-areas:
        "advert header"
        "advert content"
        "advert footer";
}

.item1 {
```

```
background: LightSkyBlue;  
grid-area: header;  
}  
  
.item2 {  
background: LightSalmon;  
grid-area: advert;  
}  
  
.item3 {  
background: PaleTurquoise;  
grid-area: content;  
/* Only change code below this line */  
  
display: grid;  
  
grid-template-columns: auto 1fr;  
/* Only change code above this line */  
}  
  
.item4 {  
background: lightpink;  
grid-area: footer;  
}  
  
.itemOne {  
background: PaleGreen;  
}  
  
.itemTwo {  
background: BlanchedAlmond;  
}  
  
</style>
```

```
<div class="container">  
  <div class="item1">header</div>  
  <div class="item2">advert</div>  
  <div class="item3">  
    <div class="itemOne">paragraph1</div>  
    <div class="itemTwo">paragraph2</div>  
  </div>  
  <div class="item4">footer</div>  
</div>
```