

UD3.2 CSS AMB GRID

ÍNDEX

1.	<i>CONCEPTES BÀSICS DE DISPOSICIÓ GRID</i>	2
1.1.	Què és una quadrícula?.....	2
	Carrils fixos i flexibles	2
	Col·locació de elements.....	2
	Creació de pistes addicionals per a albergar continguts	2
	Control d'alignació.....	2
	Control del solapament de continguts	2
2.	<i>CONTENIDOR DE QUADRÍCULA</i>	3
3.	<i>PISTES DE QUADRÍCULA.....</i>	4
3.1.	Exemple bàsic.....	4
3.2.	La unitat fr.....	5
3.3.	Grandàries desiguals.....	6
3.4.	Mescla de grandàries flexibles i absolutes	6
3.5.	Llistats de pistes amb notació repeat().....	7
3.6.	Reixetes implícites i explícites	8
3.7.	Dimensionament de vies i minmax.....	9
4.	<i>LÍNIES DE QUADRÍCULA</i>	10

1. CONCEPTES BÀSICS DE DISPOSICIÓ GRID

El disseny de quadrícula CSS (GRID) introduceix un sistema de quadrícula bidimensional en CSS. Les quadrícules poden utilitzar-se per a distribuir les principals àrees de la pàgina o xicotets elements de la interfície d'usuari. Estos apunts presenten el CSS Grid Layout. Les característiques que es mostren en este resum s'explicaran amb més detall en la resta d'esta guia.

1.1. Què és una quadrícula?

Una quadrícula és un conjunt de línies horizontals i verticals que es creuen i defineixen columnes i files. Els elements poden col·locar-se en la quadrícula dins d'estes línies de columnes i files. El disseny de quadrícula CSS té les següents característiques:

Carrils fixos i flexibles

Pot crear una quadrícula amb grandàries de pista fixos, utilitzant píxels, per exemple. Això estableix la quadrícula en el píxel especificat que s'ajusta al disseny que desitja. També pot crear una quadrícula utilitzant grandàries flexibles amb percentatges o amb la unitat `fr` dissenyada per a este fi.

Col·locació de elements

Pot col·locar elements en un lloc precís de la quadrícula utilitzant números de línia, noms o apuntant a una zona de la quadrícula. Grid també conté un algorisme per a controlar la col·locació d'elements que no tenen una posició explícita en la quadrícula.

Creació de pistes addicionals per a albergar continguts

Pot definir una quadrícula explícita amb el disseny de quadrícula. L'especificació Grid Layout és prou flexible com per a afegir files i columnes addicionals quan siga necessari. S'inclouen funcions com afegir "tantes columnes com còpien en un contenidor".

Control d'alignació

La quadrícula conté funcions d'alignació perquè puguem controlar com s'alignen els elements una vegada col·locats en una àrea de la quadrícula, i com s'aligna tota la quadrícula.

Control del solapament de continguts

Es pot col·locar més d'un element en una cel·la o àrea de la quadrícula i poden superposar-se parcialment entre si. Esta superposició pot controlar-se amb la propietat `z-index`.

Grid és una potent especificació que, quan es combina amb altres parts de CSS com [flexbox \(proper document de teoria\)](#), pot ajudar-te a crear dissenys que abans eren impossibles de construir en CSS. Tot comença amb la creació d'una quadrícula en el seu **contenidor de quadrícula**.

2. CONTENIDOR DE QUADRICULA

Creem un *contenidor de quadricula* declarant `display: grid` o `display: inline-grid` en un element. Quan ho fem, tots els *fils directes* d'aquest element es convertiran en *elements de quadricula*.

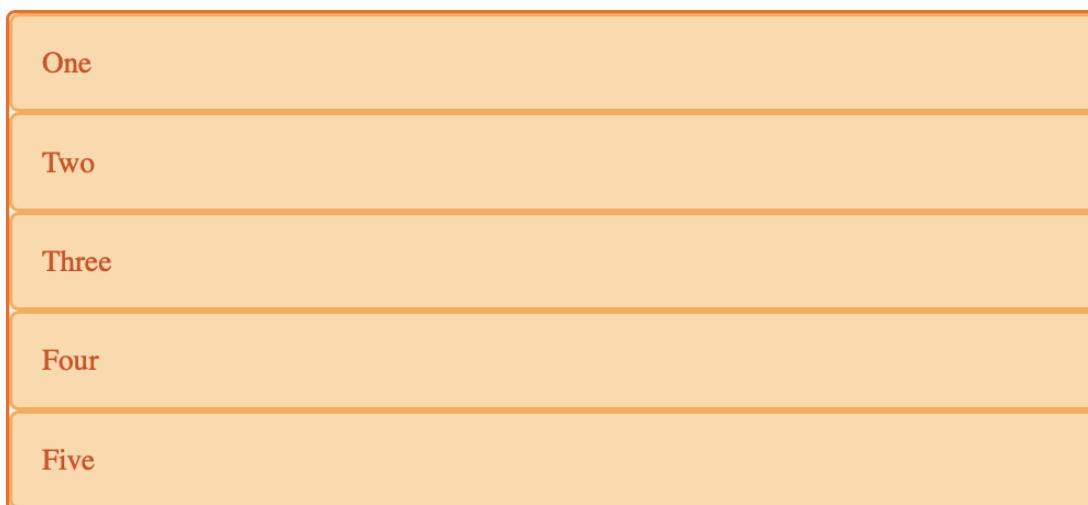
En este exemple, tinc un div contenidor amb una classe 'wrapper' i, dins hi ha cinc elements fill.

```
<div class="wrapper">  
  <div>Un</div>  
  <div>Dos</div>  
  <div>Tres</div>  
  <div>Quatre</div>  
  <div>Cinc</div>  
</div>
```

Faig del .wrapper un contenidor de quadricula.

```
.wrapper {  
  display: grid;  
}
```

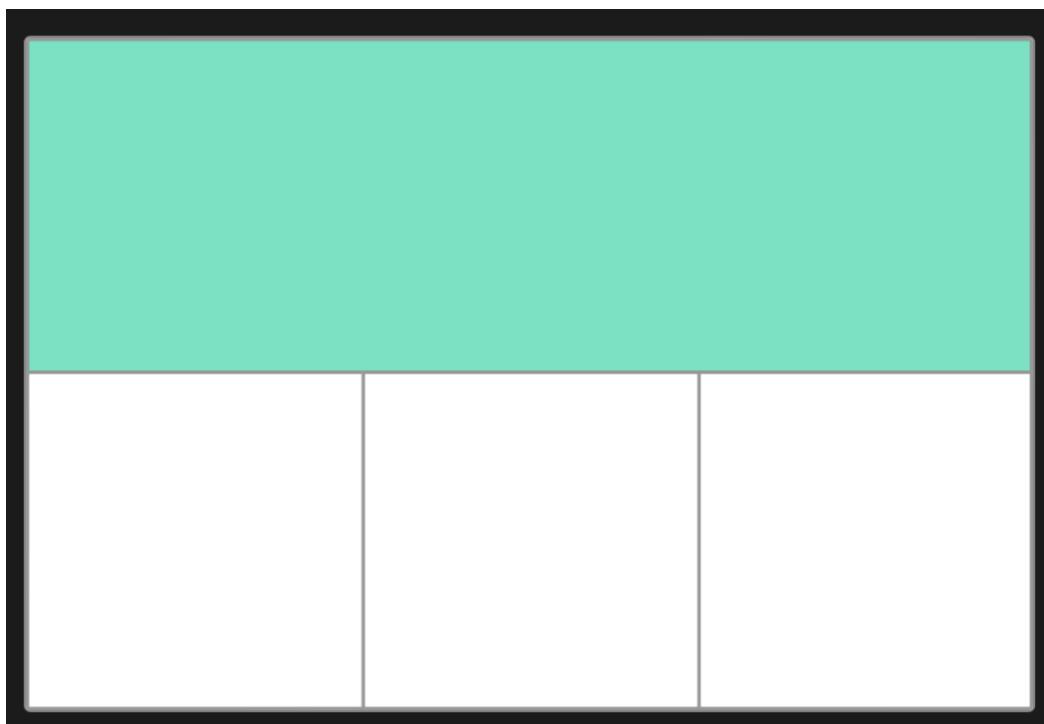
Tots els fills directes són ara elements de la quadricula. En un navegador web, no veuràs cap diferència en com es mostren estos elements abans de convertir-los en una quadricula, ja que grid ha creat una quadricula d'una sola columna per als elements.



Si volem que això se sembla més a una quadrícula, hem d'afegir pistes de columna.

3. PISTES DE QUADRÍCULA

Definim files i columnes en la nostra quadricula amb les propietats grid-template-rows i grid-template-columns. Estes defineixen les línies de la quadrícula. Un traçat de *quadrícula* és l'espai entre dues línies adjacents de la quadrícula. La imatge de baix mostra una pista ressaltada - esta és la pista de la primera fila de la nostra quadricula.



Els traçats de la quadricula es defineixen en la quadricula explícita utilitzant les propietats `grid-template-columns` i `grid-template-rows`.

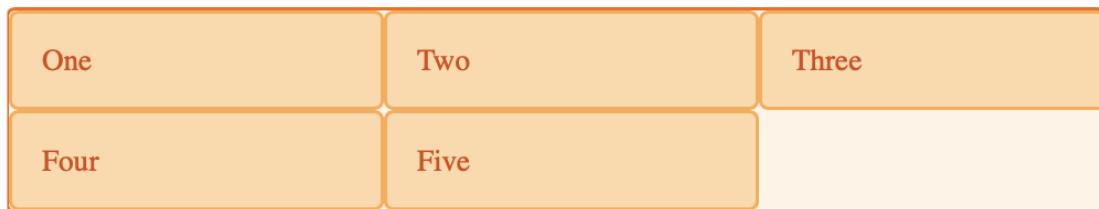
3.1. Exemple bàsic

Puc afegir al nostre exemple anterior mitjançant l'addició de la propietat `grid-template-columns`, a continuació, definir la grandària de les pistes de columna.

Ara he creat una quadricula amb tres pistes de columna de 200 pixels d'ample. Els elements fill es col·locaran en esta quadricula un en cada cel·la de la quadricula.

```
<div class="wrapper">
  <div>Un</div>
  <div>Dos</div>
  <div>Tres</div>
  <div>Quatre</div>
  <div>Cinc</div>
</div>
```

```
.wrapper {  
    display: grid;  
    grid-template-columns: 200px 200px 200px;  
}
```

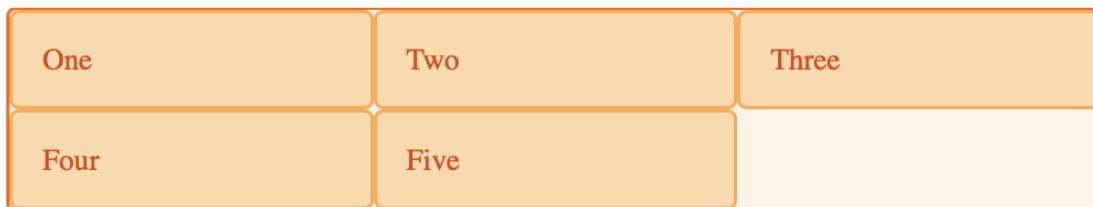


3.2. La unitat fr

Els traçats poden definir-se utilitzant qualsevol unitat de longitud. Grid també introduceix una unitat de longitud addicional per a ajudar-nos a crear traçats de quadrícula flexibles. La nova unitat `fr` representa una fracció de l'espai disponible en el contingidor de la quadricula. La següent definició de quadricula crearia tres tracks d'igual amplària que creixen i decreixen segons l'espai disponible.

```
<div class="wrapper">  
    <div>Un</div>  
    <div>Dos</div>  
    <div>Tres</div>  
    <div>Quatre</div>  
    <div>Cinc</div>  
</div>
```

```
.wrapper {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
}
```

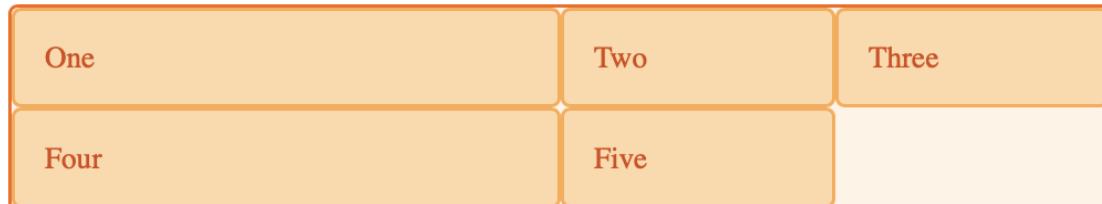


3.3. Grandàries desiguals

En el següent exemple, creem una definició amb una pista de 2fr i després dues pistes de 1fr. L'espai disponible es divideix en quatre. S'assignen dues parts a la primera pista i una part a cadascuna de les dues pistes següents.

```
<div class="wrapper">  
  <div>Un</div>  
  <div>Dos</div>  
  <div>Tres</div>  
  <div>Quatre</div>  
  <div>Cinc</div>  
</div>
```

```
.wrapper {  
  display: grid;  
  grid-template-columns: 2fr 1fr 1fr;  
}
```

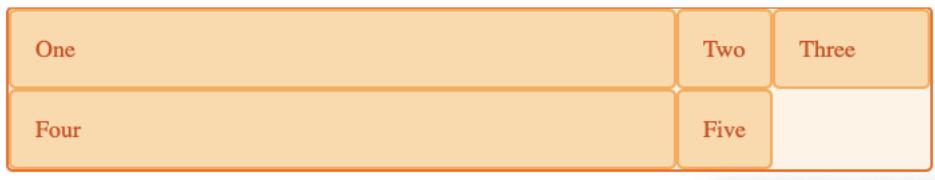


3.4. Mescla de grandàries flexibles i absolutes

En este últim exemple, mesclam pistes de grandària absoluta amb unitats `fr`. La primera pista és de 500 pixels, per la qual cosa l'amplària fixa es resta de l'espai disponible. L'espai restant es divideix en tres i s'assiganya en proporció a les dues pistes flexibles.

```
<div class="wrapper">  
  <div>Un</div>  
  <div>Dos</div>  
  <div>Tres</div>  
  <div>Quatre</div>  
  <div>Cinc</div>  
</div>
```

```
.wrapper {  
    display: grid;  
    grid-template-columns: 500px 1fr 2fr;  
}
```



3.5. Llistats de pistes amb notació repeat()

Les reixetes grans amb moltes pistes poden utilitzar la notació `repeat()`, per a repetir tot o una secció del llistat de pistes. Per exemple, la definició de la quadricula:

```
.wrapper {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
}
```

També es pot escriure com:

```
.wrapper {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
}
```

La notació de repetició es pot utilitzar per a una part de la llista de pistes. En el següent exemple he creat una quadricula amb una pista inicial de 20 píxels, després una secció de repetició de 6 pistes de `1fr` i després una pista final de 20 píxels.

```
.wrapper {  
    display: grid;  
    grid-template-columns: 20px repeat(6, 1fr) 20px;  
}
```

La notació de repetició pren el llistat de pistes i l'utilitza per a crear un patró repetitiu de pistes. En el següent exemple, la meua graella constarà de 10 pistes, una pista de `1fr` i, a continuació, una pista de `2fr`. Este patró es repetirà cinc vegades.

```
.wrapper {  
    display: grid;  
    grid-template-columns: repeat(5, 1fr 2fr);  
}
```

3.6. Reixetes implícites i explícites

En crear la nostra quadricula d'exemple definim específicament les nostres pistes de columnes amb la propietat `grid-template-columns`, però la quadricula també va crear files per si mateixa. Estes files formen part de la quadricula implícita. Mentre que la quadricula explícita consisteix en qualsevol fila i columna definida amb `grid-template-columns` o `grid-template-rows`.

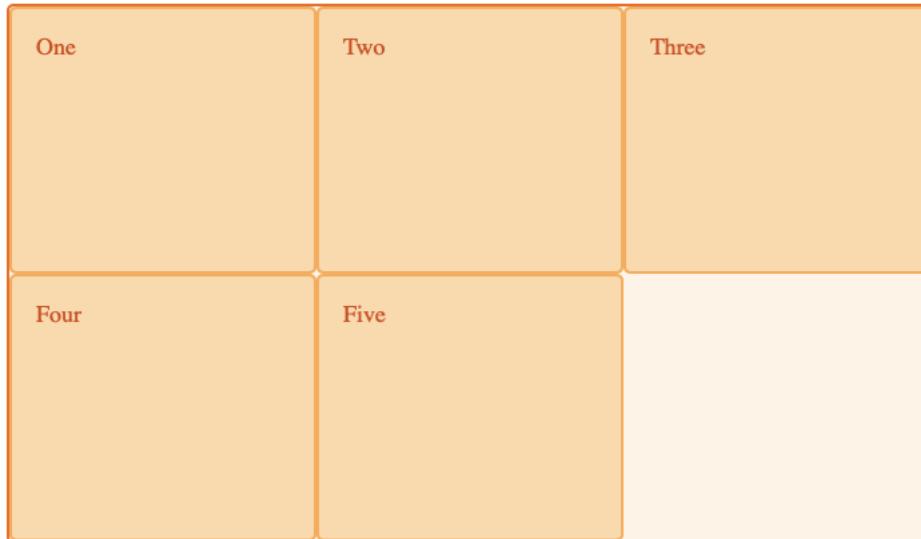
Si col·loca alguna cosa fora de la quadricula definida -o a causa de la quantitat de contingut, es necessiten més pistes de quadricula- llavors la quadricula crea files i columnes en la quadricula implícita. Per defecte, la grandària d'estes files s'ajusta automàticament en funció del contingut que contenen.

També pot definir una grandària fixa per a les pistes creades en la quadricula implícita amb les propietats `grid-auto-rows` i `grid-auto-columns`.

En l'exemple següent, utilitzem `grid-auto-rows` per a garantir que les pistes creades en la quadricula implícita tinguen 200 píxels d'altura.

```
<div class="wrapper">  
    <div>Un</div>  
    <div>Dos</div>  
    <div>Tres</div>  
    <div>Quatre</div>  
    <div>Cinc</div>  
</div>
```

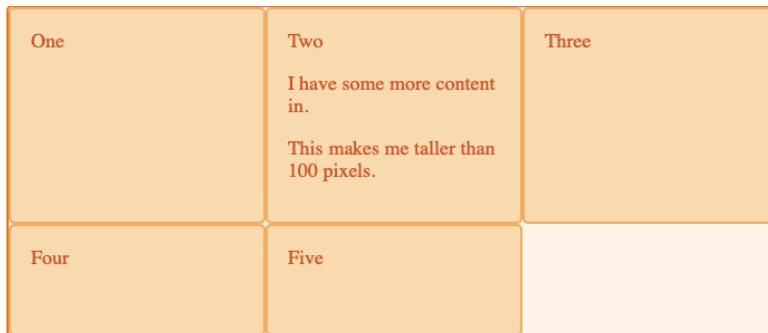
```
.wrapper {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-auto-rows: 200px;  
}
```



3.7. Dimensionament de vies i minmax

En configurar una quadricula explícita o definir la grandària de les files o columnes creades automàticament, pot ser que vulguem donar a les files una grandària mínima, però també assegurar-nos que s'expandeixen per a ajustar-se a qualsevol contingut que s'afija. Per exemple, pot ser que vulga que les meues files mai es contraguen per davall de 100 pixels, però si el meu contingut s'estira fins a 300 pixels d'altura, llavors m'agradaaria que la fila s'estire fins a eixa altura.

Grid té una solució per a això amb la funció `minmax()`. En el següent exemple estic usant `minmax()` en el valor de `grid-auto-rows`. Això significa que les files creades automàticament tindran un mínim de 100 pixels d'alt, i un màxim d'acte . Usant `auto` significa que la grandària es veurà en la grandària del contingut i s'estirarà per a donar espai per a l'element més alt en una cel·la, en esta fila.

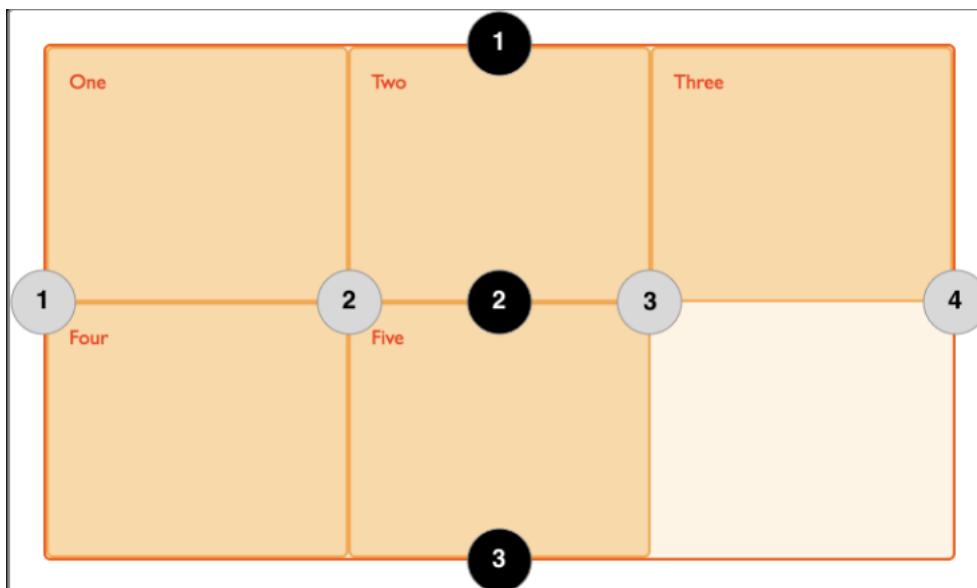


```
.wrapper {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-auto-rows: minmax(100px, auto);
}
```

```
<div class="wrapper">
<div>Un</div>
<div>
    Dos
    <p>Tinc una mica més de contingut en.</p>
    <p>Això em fa més alt de 100 píxels.</p>
</div>
<div>Tres</div>
<div>Quatre</div>
<div>Cinc</div>
</div>
```

4. LÍNIES DE QUADRÍCULA

Cal tindre en compte que quan definim una quadricula definim les pistes de la quadricula, no les línies. La quadricula llavors ens dona línies numerades per a usar quan posicionem elements. En la nostra quadrícula de tres columnes i dues files tenim quatre línies de column.



Les línies es numeren segons el mode d'escriptura del document. En una llengua d'esquerra a dreta, la línia 1 està en el costat esquerre de la quadrícula. En un idioma de dreta a esquerra, està en el costat dret de la quadrícula. Les línies també poden nomenar-se, i veurem com fer-ho en una guia posterior d'esta sèrie.

4.1. Posicionament d'articles contra línies

Explorarem la col·locació basada en línies amb tot detall en un article posterior. El següent exemple mostra com fer-ho de manera senzilla. En col·locar un element, ens dirigim a la línia - en lloc de la pista.

En el següent exemple estic col·locant els dos primers elements en la nostra quadricula de tres columnes, utilitzant les propietats `grid-column-start`, `grid-column-end`, `grid-row-start` i `grid-row-end`. Treballant d'esquerra a dreta, el primer element es col·loca contra la línia de columna 1, i s'estén fins a la línia de columna 4, que en el nostre cas és la línia de l'extrem dret de la quadricula. Comença en la línia de fila 1 i acaba en la línia de fila 3, per la qual cosa abasta dues pistes de fila.

El segon element comença en la línia 1 de la columna de la quadrícula i abasta una pista. Este és el valor per defecte, per la qual cosa no és necessari especificar la línia final. També abasta dues pistes de fila des de la línia 3 fins a la línia 5. Els altres elements es col·locaran en els espais buits de la quadrícula.



```
<div class="wrapper">  
  <div class="box1">Un</div>  
  <div class="box2">Dos</div>  
  <div class="box3">Tres</div>  
  <div class="box4">Quatre</div>  
  <div class="box5">Cinc</div>  
</div>
```

```
.wrapper {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-auto-rows: 100px;  
}  
  
.box1 {  
    grid-column-start: 1;  
    grid-column-end: 4;  
    grid-row-start: 1;  
    grid-row-end: 3;  
}  
  
.box2 {  
    grid-column-start: 1;  
    grid-row-start: 3;  
    grid-row-end: 5;  
}
```

4.2. Abreviatures de posicionament de línies

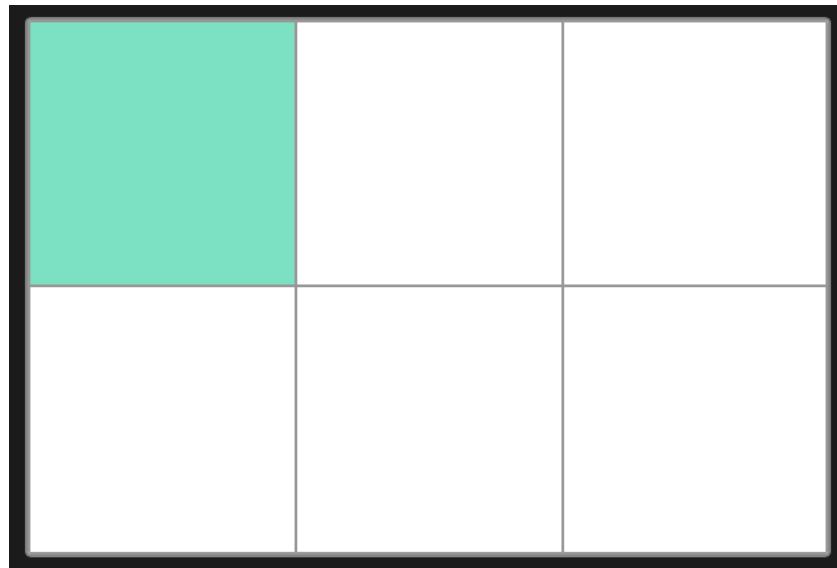
Els valors a mà alçada utilitzats anteriorment poden comprimir-se en una línia per a les columnes amb grid-column, i en una línia per a les files amb grid-row. El següent exemple donaria el mateix posicionament que en el codi anterior, però amb molt menys CSS. El valor abans de la barra obliqua (/) és la línia d'inici, el valor després de la línia final.

Pot ometre el valor final si l'àrea només abasta una pista.

```
.wrapper {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-auto-rows: 100px;  
}  
  
.box1 {  
    grid-column  
: 1 / 4;  
    grid-row: 1 / 3;  
}  
  
.box2 {  
    grid-column: 1;  
    grid-row: 3 / 5;  
}
```

5. CEL·LES DE LA QUADRÍCULA

Una *cel·la de quadrícula* és la unitat més xicoteta d'una quadrícula. Conceptualment és com una cel·la de taula. Com vam veure en els nostres exemples anteriors, una vegada que una quadrícula es defineix com a pare, els elements fills es col·locaran en una cel·la cadascun de la quadrícula definida. En la següent imatge, he ressaltat la primera cel·la de la quadrícula.



5.1. Zones quadriculades

Els elements poden abastar una o més cel·les tant per fila com per columna, i això crea una *àrea de quadrícula*. Les àrees de quadrícula han de ser rectangulars; no és possible crear una àrea en forma de L, per exemple. L'àrea de quadrícula ressaltada abasta dues files i dues columnes.

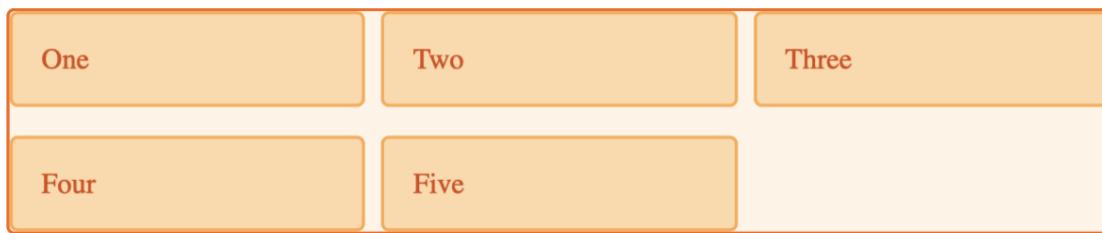


5.2. Canalons

Es poden crear *carrerons* entre les cel·les de la quadrícula utilitzant les propietats `column-gap` i `row-gap`, o l'abreviatura `gap`. En el següent exemple, estic creant un espai de 10 pixels entre columnes i un espai de `1em` entre files.

```
.wrapper {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    column-gap: 10px;  
    row-gap: 1em;  
}
```

```
<div class="wrapper">  
    <div>Un</div>  
    <div>Dos</div>  
    <div>Tres</div>  
    <div>Quatre</div>  
    <div>Cinc</div>  
</div>
```



Qualsevol espai utilitzat pels buits es tindrà en compte abans d'assignar espai a les pistes `fr` de longitud flexible, i els buits actuen a l'efecte de grandària com una pista de quadrícula normal, encara que no es pot col·locar res en un buit. En termes de posicionament basat en línies, el buit actua com una línia gruixuda.

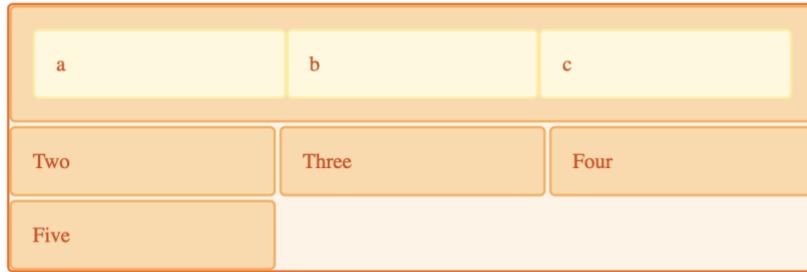
6. REIXETES NIADES

Un element de la quadricula pot convertir-se en un contenidor de la quadricula. En el següent exemple, tinc la quadricula de tres columnes que vaig crear anteriorment, amb els nostres dos elements posicionats. En este cas el primer element té alguns sub-elements. Com estos elements no són fills directes de la quadricula, no participen en el disseny de la quadricula, per la qual cosa es mostren en un flux de document normal.



6.1. Anidamiento sense subplota

Si configure `box1` per a `display:grid`, puc donar-li una definició de pista i també es convertirà en una quadrícula. Els elements es col·locaran en esta nova quadrícula.



```
.box1 {  
    grid-column-start: 1;  
    grid-column-end: 4;  
    grid-row-start: 1;  
    grid-row-end: 3;  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
}
```

```
* {  
    box-sizing: border-box;  
}  
  
.wrapper {  
    border: 2px solid #f76707;  
    border-radius: 5px;  
    gap: 3px;  
    background-color: #fff4e6;  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
}
```

```
.box {  
    border: 2px solid #ffa94d;  
    border-radius: 5px;  
    background-color: #ffd8a8;  
    padding: 1em;  
    color: #d9480f;  
}  
  
.box1 {  
    grid-column: 1 / 4;  
}  
  
.nested {  
    border: 2px solid #ffec99;  
    border-radius: 5px;  
    background-color: #fff9db;  
    padding: 1em;  
}
```

En este cas la quadricula niada no té cap relació amb la quadricula pare. Com pot veure en l'exemple, no ha heretat el gap de la quadricula principal i les línies de la quadricula niada no s'alignen amb les línies de la quadricula principal.

6.2.Subgrid

A més de les quadrícules normals, *les subcuadrícules* ens permeten crear quadrícules niades que utilitzen la definició de pista de la quadrícula pare.

Per a utilitzar-les, editem l'exemple de quadricula niada anterior per a canviar la definició de la pista `grid-template-columns: repeat(3, 1fr)`, per `grid-template-columns: subgrid`. La quadricula niada utilitzarà llavors les pistes de la quadricula pare per a maquetar els elements.

```
.box1 {  
    grid-column-start: 1;  
    grid-column-end: 4;  
    grid-row-start: 1;  
    grid-row-end: 3;  
    display: grid;  
    grid-template-columns: subgrid;  
}
```

7. SUPERPOSICIÓ D'ELEMENTS AMB Z-INDEX

Els elements de la quadrícula poden ocupar la mateixa cel·la, i en este cas podem utilitzar la propietat z-index per a controlar l'ordre en què s'apilen els elements superposats.

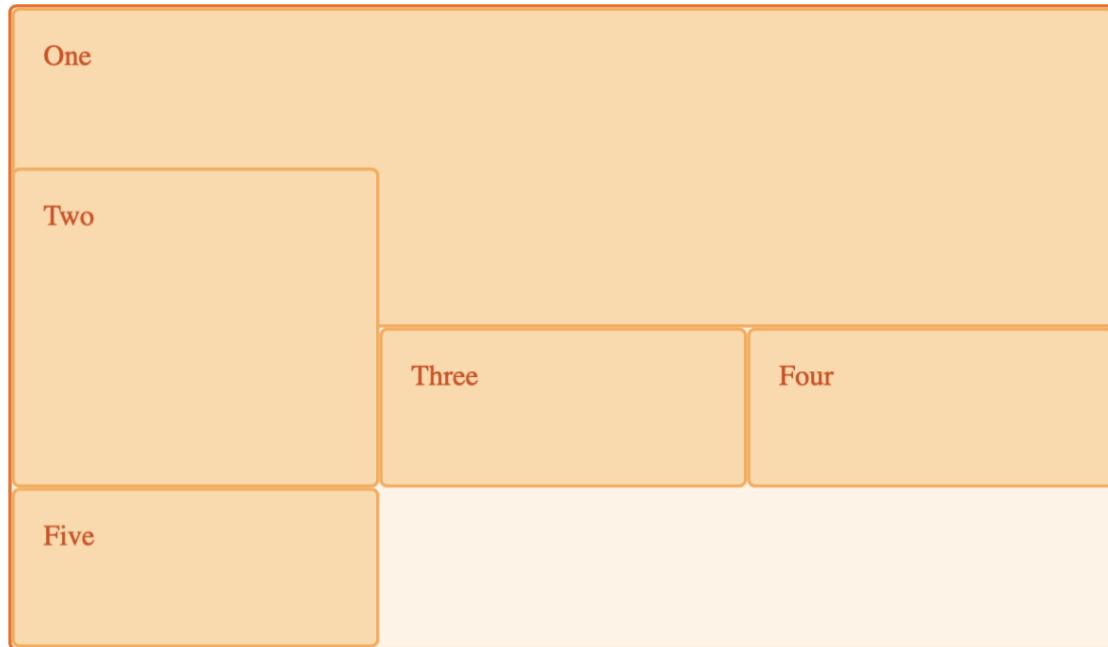
7.1. Superposició sense z-index

Si tornem al nostre exemple amb elements posicionats per número de línia, podem canviar això per a fer que dos elements se superposen.

```
<div class="wrapper">  
  <div class="box box1">Un</div>  
  <div class="box box2">Dos</div>  
  <div class="box box3">Tres</div>  
  <div class="box box4">Quatre</div>  
  <div class="box box5">Cinc</div>  
</div>
```

```
.wrapper {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-auto-rows: 100px;  
}  
  
.box1 {  
  grid-column-start: 1;  
  grid-column-end: 4;  
  grid-row-start: 1;  
  grid-row-end: 3;  
}  
  
.box2 {  
  grid-column-start: 1;  
  grid-row-start: 2;  
  grid-row-end: 4;  
}
```

L'element `box2` està ara superposat a `box1`, es mostra en la part superior ja que ve més tard en l'ordre d'origen.



7.2. Controlar el ordre

Podem controlar l'ordre en el qual els elements s'apilen utilitzant la propietat `z-index` igual que els elements posicionats. Si donem a `box2` un `z-index` inferior al de `box1`, es mostrarà davall de `box1` en la pila.

```
.wrapper {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-auto-rows: 100px;  
}  
  
.box1 {  
    grid-column-start: 1;  
    grid-column-end: 4;  
    grid-row-start: 1;  
    grid-row-end: 3;  
    z-index: 2;  
}  
  
.box2 {  
    grid-column-start: 1;  
    grid-row-start: 2;  
    grid-row-end: 4;  
    z-index: 1;  
}
```



