Hello, subgrid

@rachelandrew at CSSConf EU 2019

Doing things on the web since 1996

Co-founder **Perch CMS** & **Notist**. Editor in Chief **Smashing Magazine**. Writer of many books. **CSS Working Group** Member representing **Fronteers**. Spec editor **Multicol** and Page Floats. **MDN** tech writer.

CSS Grid Layout

#cssconfeu 2014

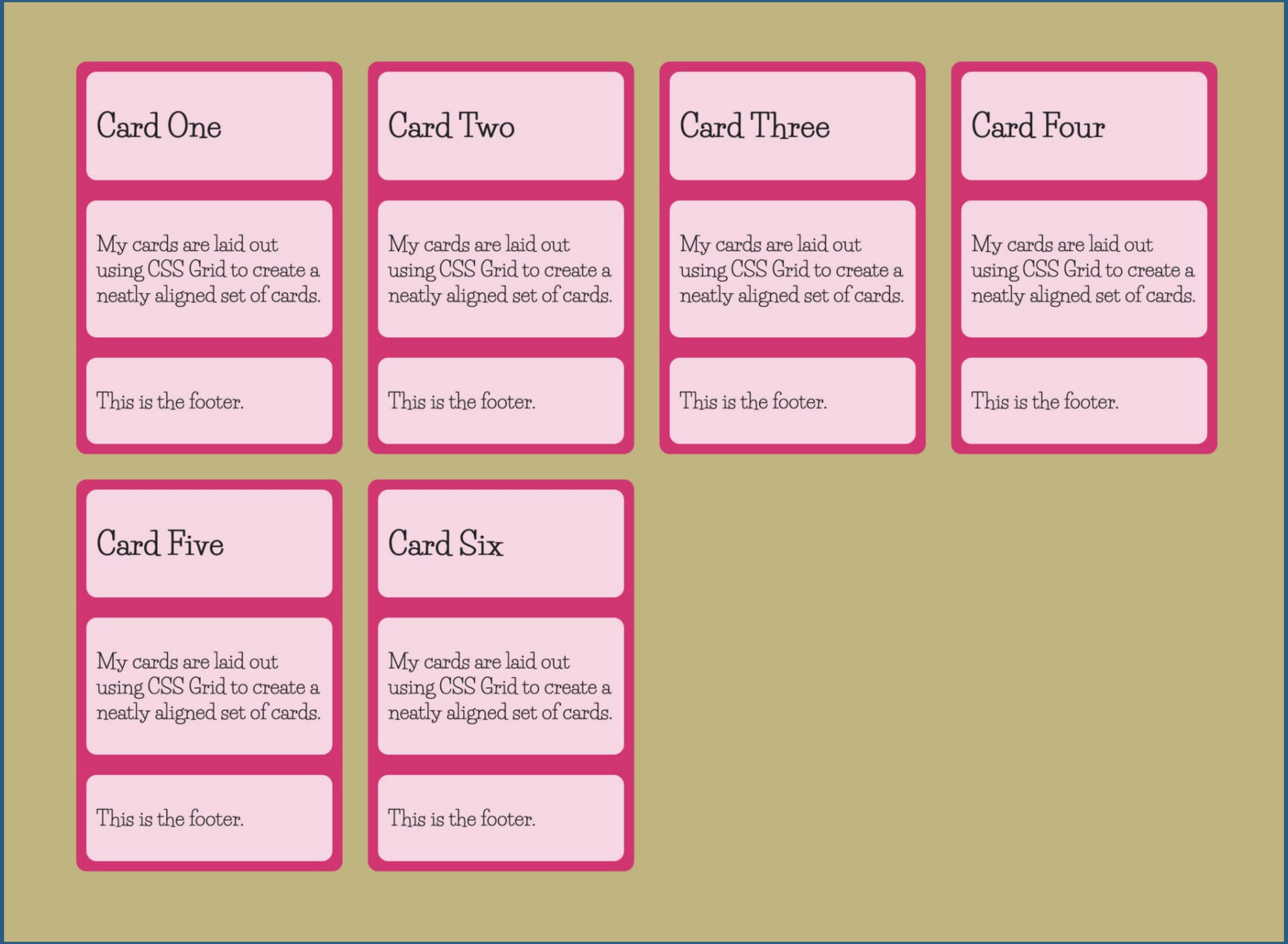
https://www.flickr.com/photos/blachswan/15174207821

Friday, 12 September 14

CSS Grid Layout

Two years on.

So, what’s next?



Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

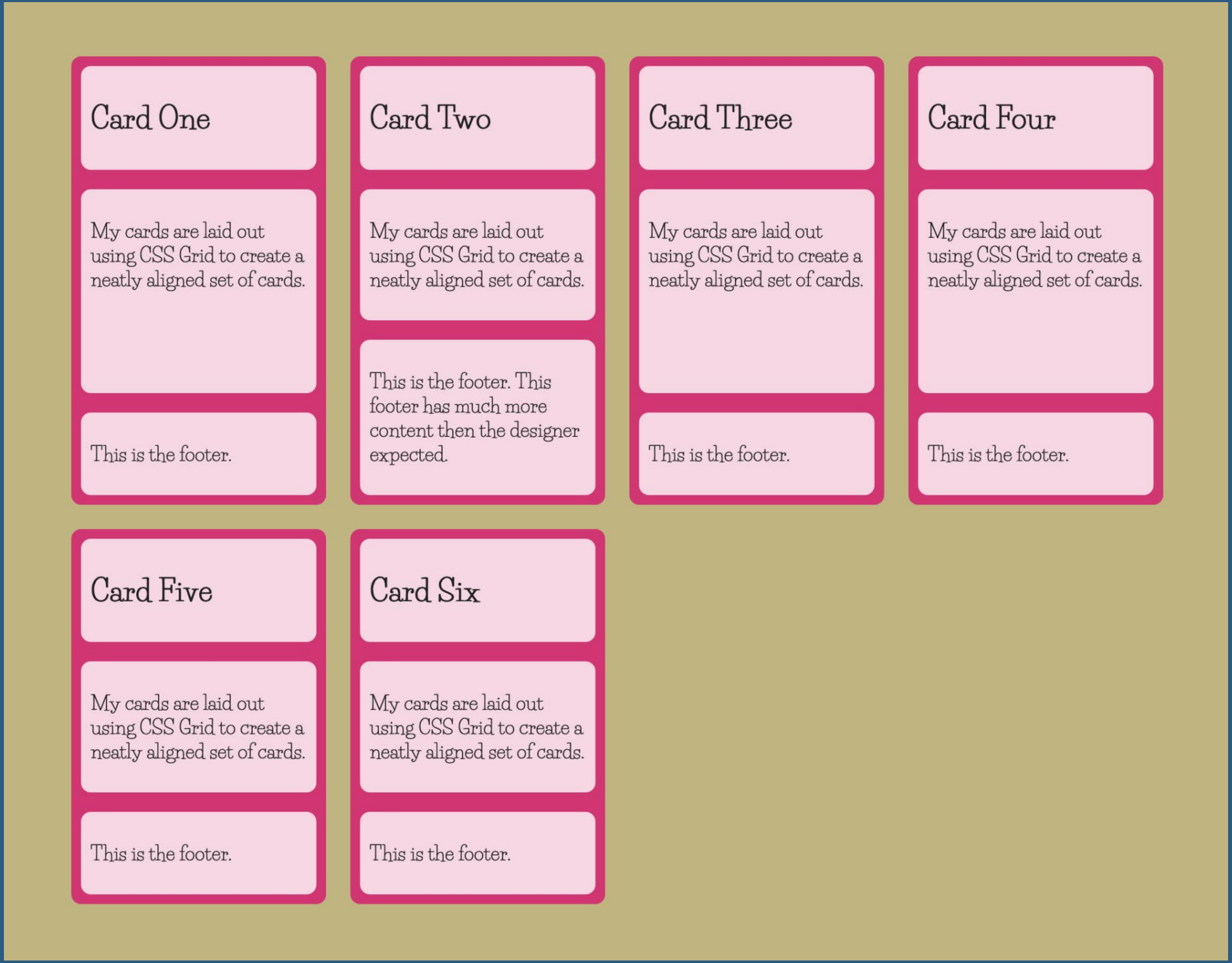
My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.



Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content then the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

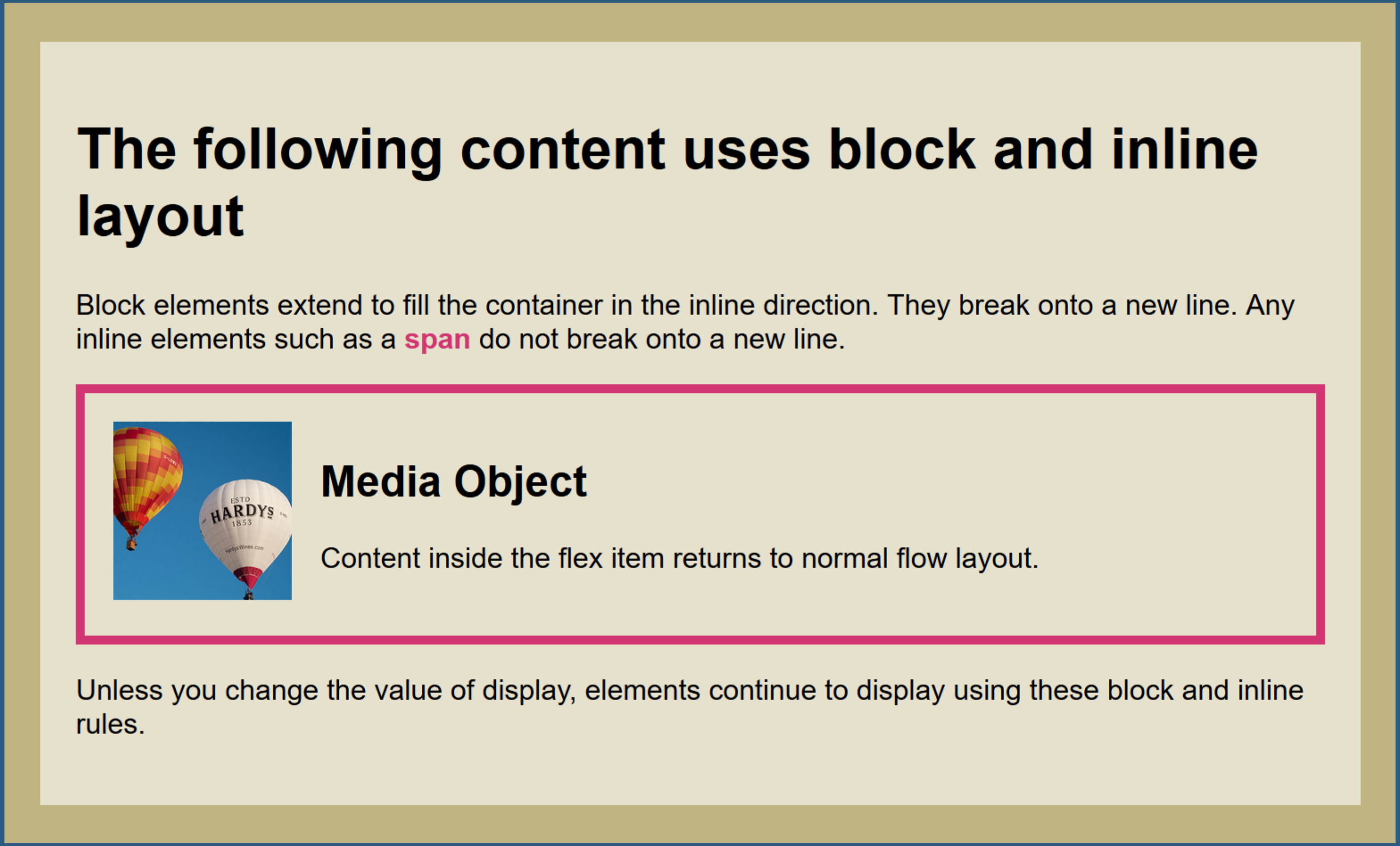
Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Only **direct children** become grid or flex items.

Their children return to normal flow.



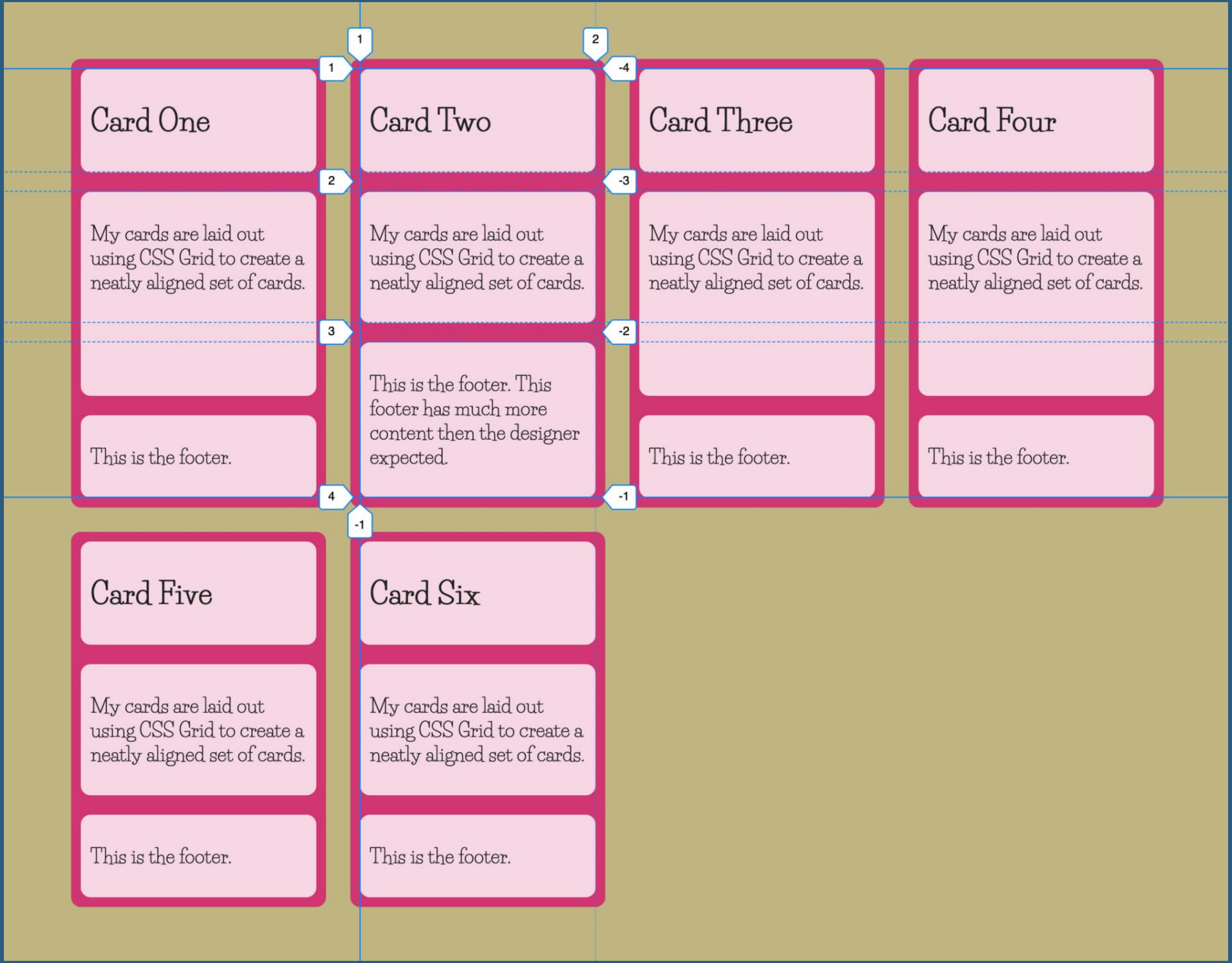
The following content uses block and inline layout

Block elements extend to fill the container in the inline direction. They break onto a new line. Any inline elements such as a **span** do not break onto a new line.

Media Object

Content inside the flex item returns to normal flow layout.

Unless you change the value of display, elements continue to display using these block and inline rules.



Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content then the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

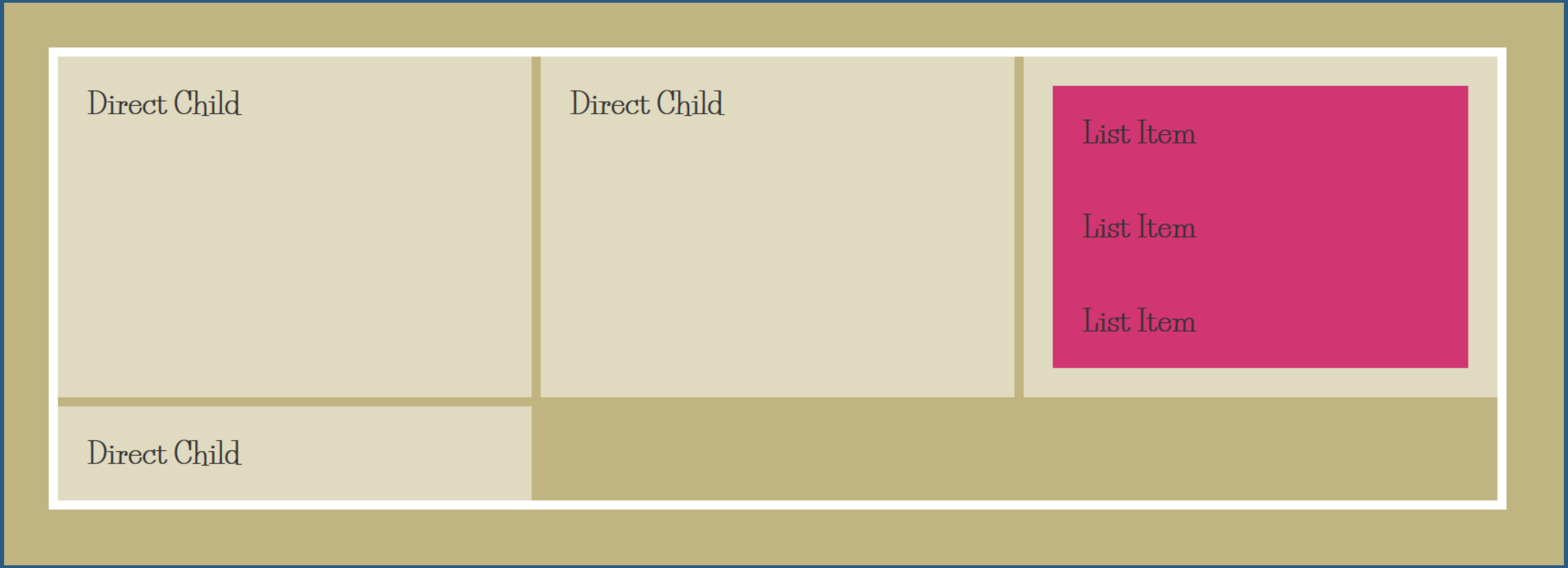
Card Six

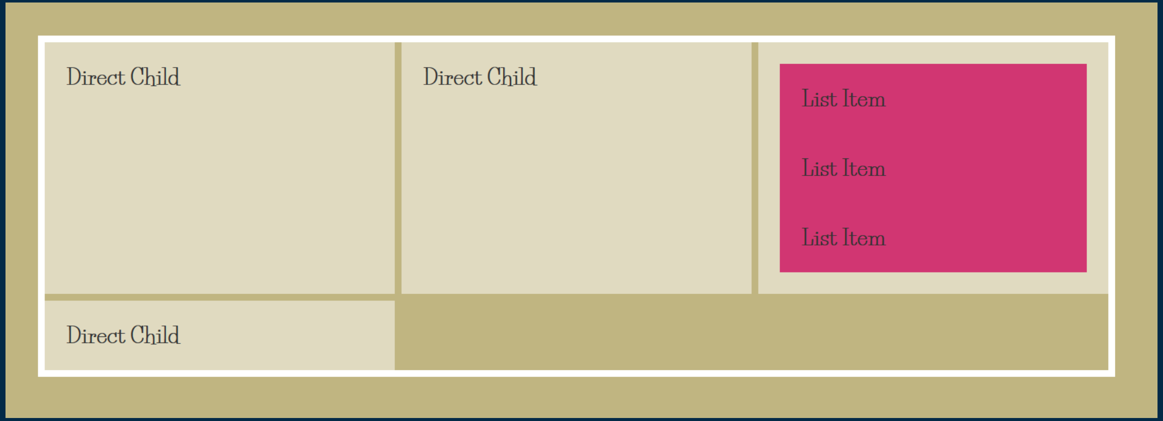
My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

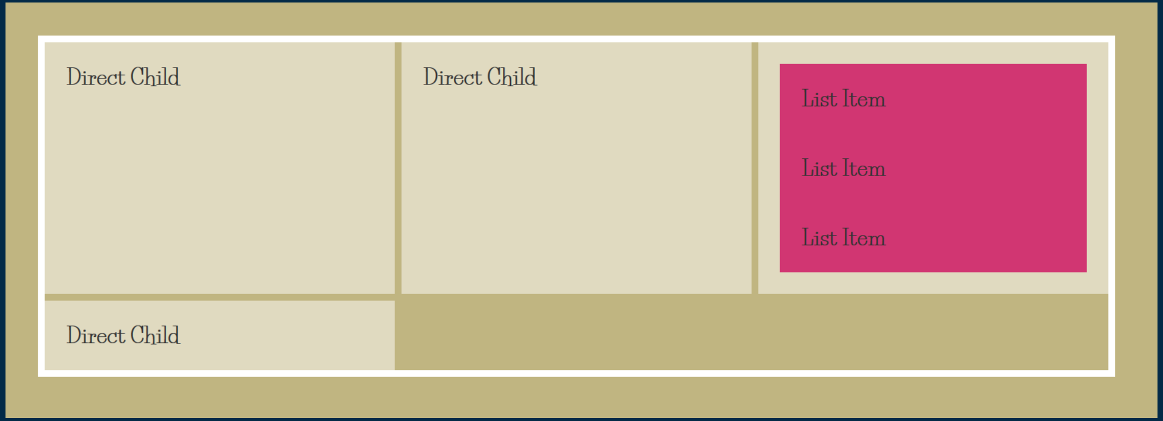
display: contents

Removing the box from the layout.

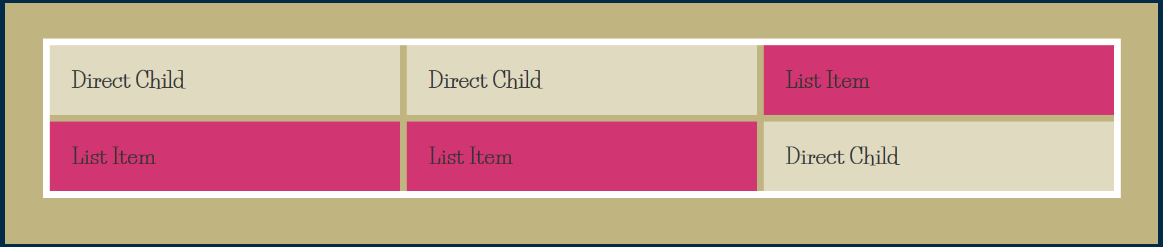

Direct Child
Direct Child
List Item
List Item
List Item
Direct Child



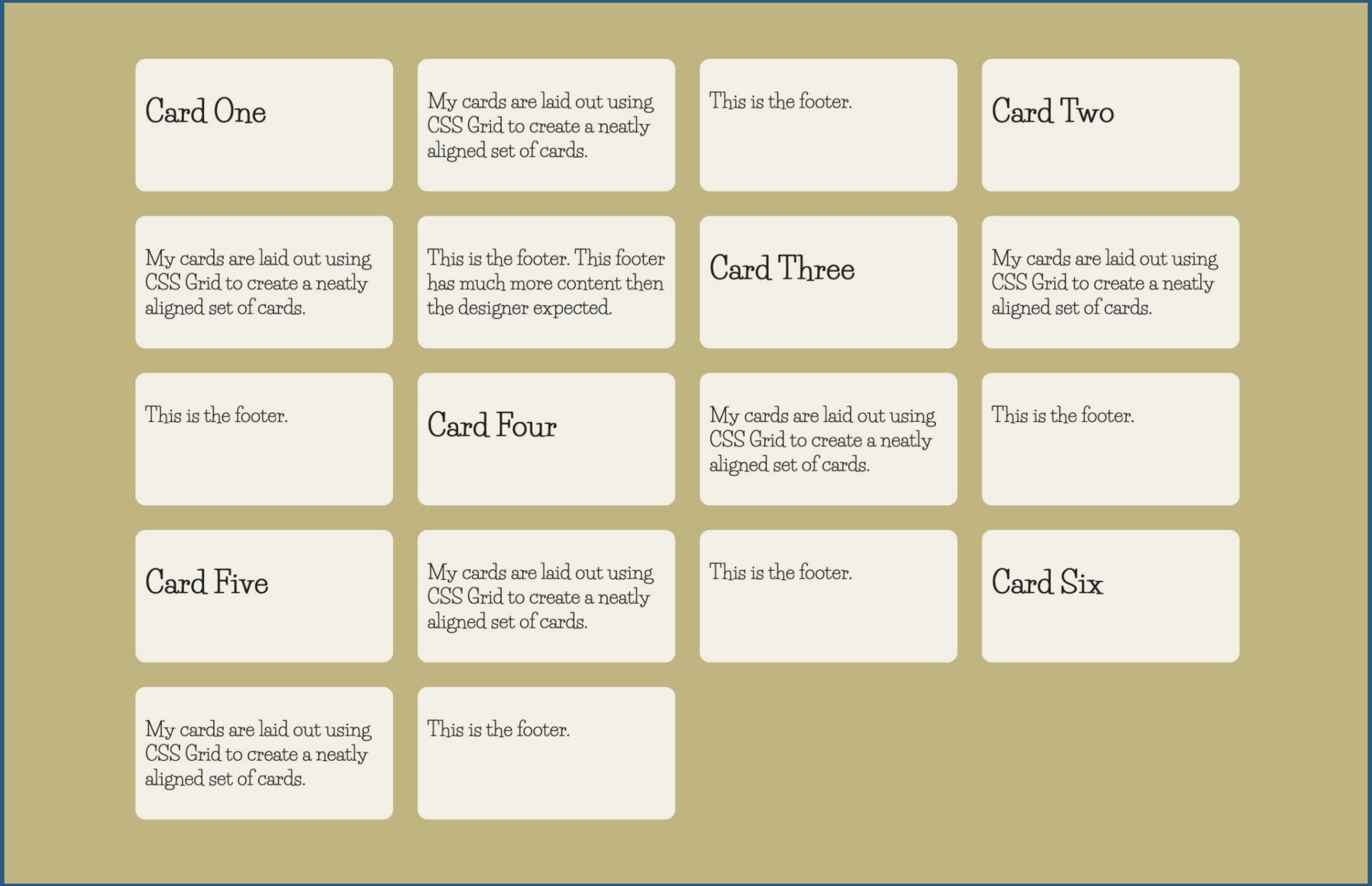
```
<!--HTML-->
<div class="grid">
  <div>Direct Child</div>
  <div>Direct Child</div>
  <ul>
    <li>List Item</li>
    <li>List Item</li>
    <li>List Item</li>
  </ul>
  <div>Direct Child</div>
</div>
```



```css
/* CSS */
.grid {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
}

.grid > * {
  background-color: rgba(255,255,255,.5);
}

ul > * {
  background-color: rgb(209,54,114);
}

ul {

}
```



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

.grid > * {
  background-color: rgba(255,255,255,.5);
}

ul > * {
  background-color: rgb(209,54,114);
}

ul {
  display: contents;
}
```



Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content then the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

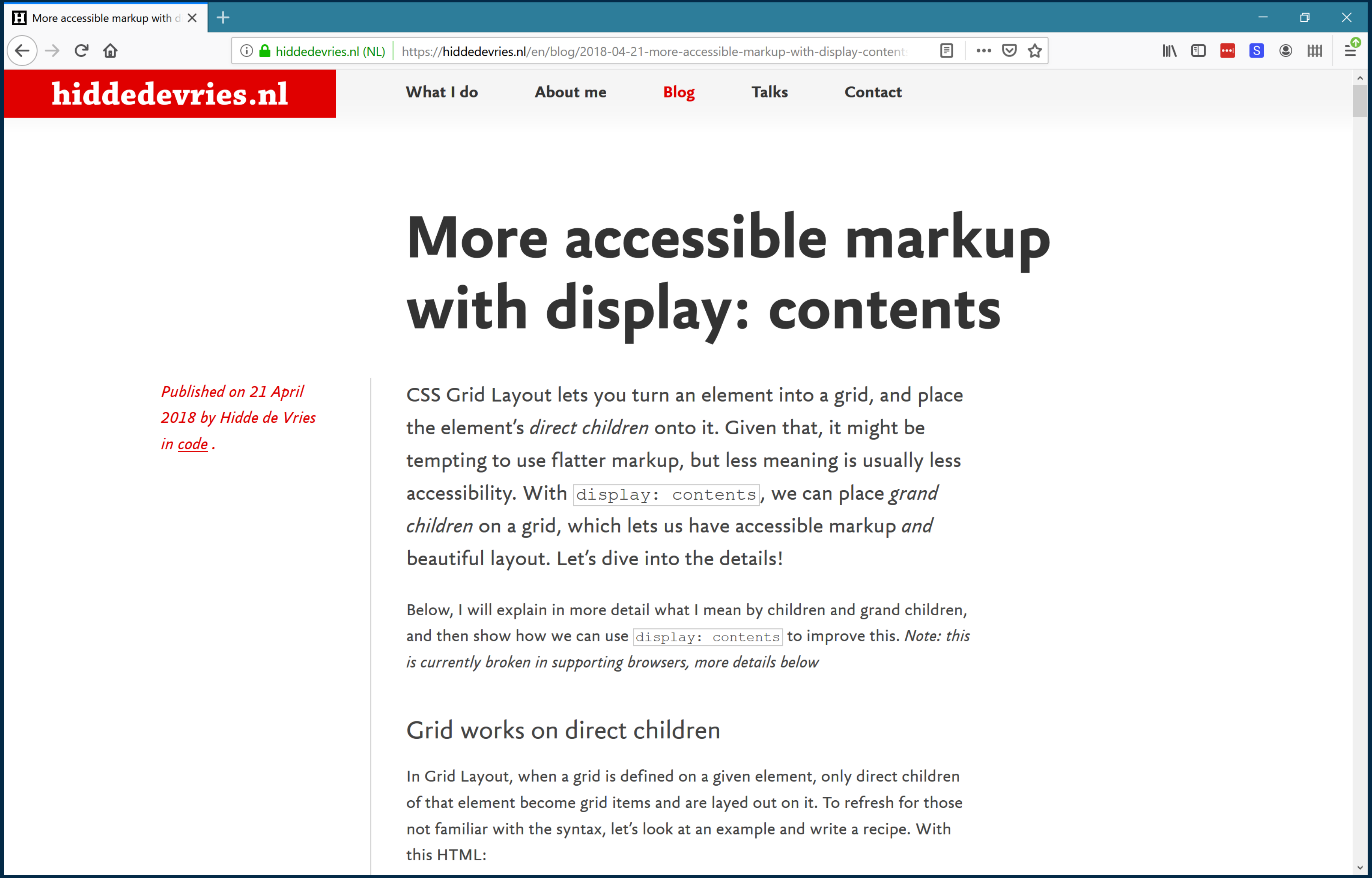
My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.


More accessible markup with d

hiddedevries.nl (NL) https://hiddedevries.nl/en/blog/2018-04-21-more-accessible-markup-with-display-content

hiddedevries.nl

What I do About me Blog Talks Contact

More accessible markup with display: contents

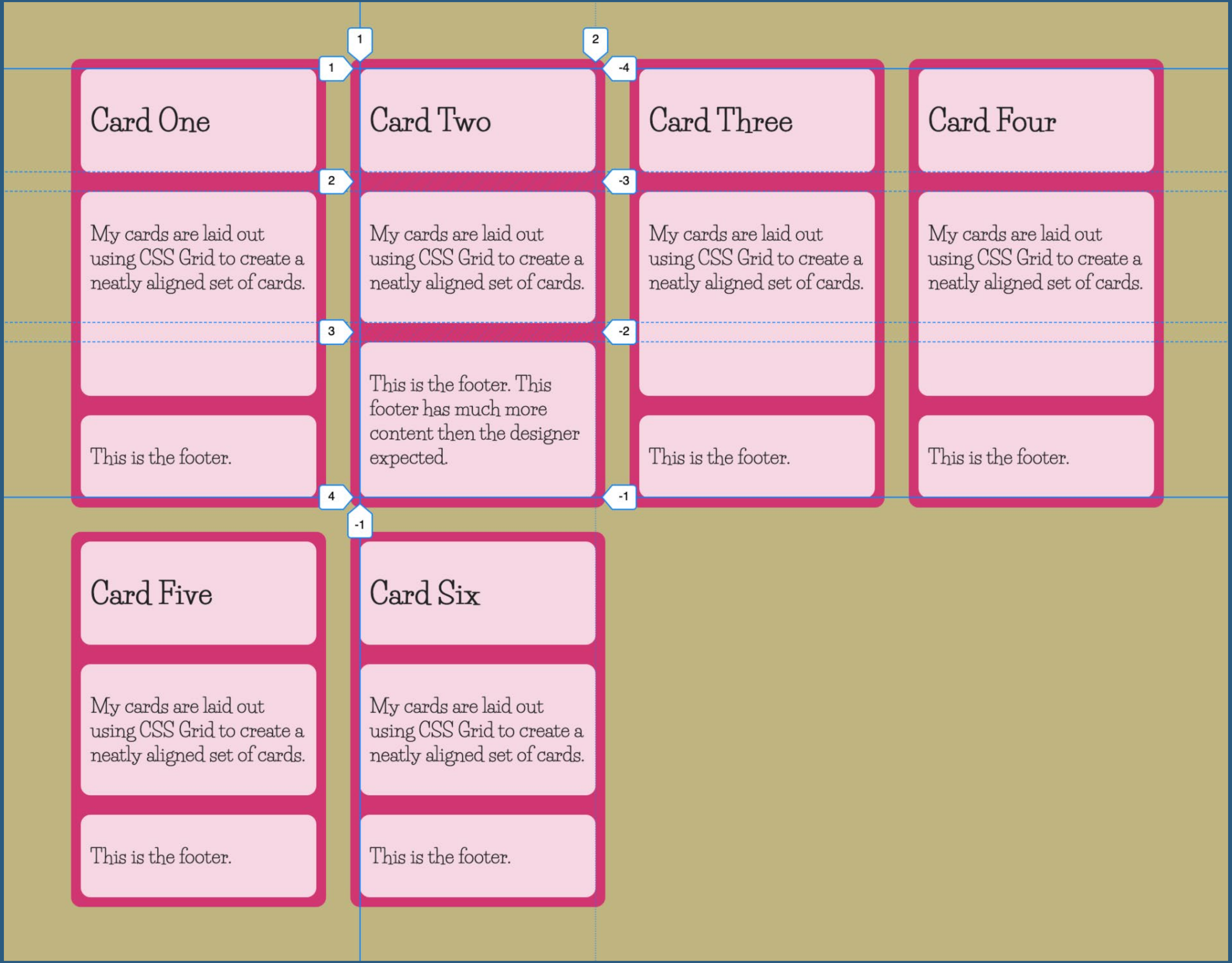
Published on 21 April 2018 by Hidde de Vries in code .

CSS Grid Layout lets you turn an element into a grid, and place the element's *direct children* onto it. Given that, it might be tempting to use flatter markup, but less meaning is usually less accessibility. With `display: contents`, we can place *grand children* on a grid, which lets us have accessible markup *and* beautiful layout. Let's dive into the details!

Below, I will explain in more detail what I mean by children and grand children, and then show how we can use `display: contents` to improve this. *Note: this is currently broken in supporting browsers, more details below*

Grid works on direct children

In Grid Layout, when a grid is defined on a given element, only direct children of that element become grid items and are layed out on it. To refresh for those not familiar with the syntax, let's look at an example and write a recipe. With this HTML:



Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content then the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

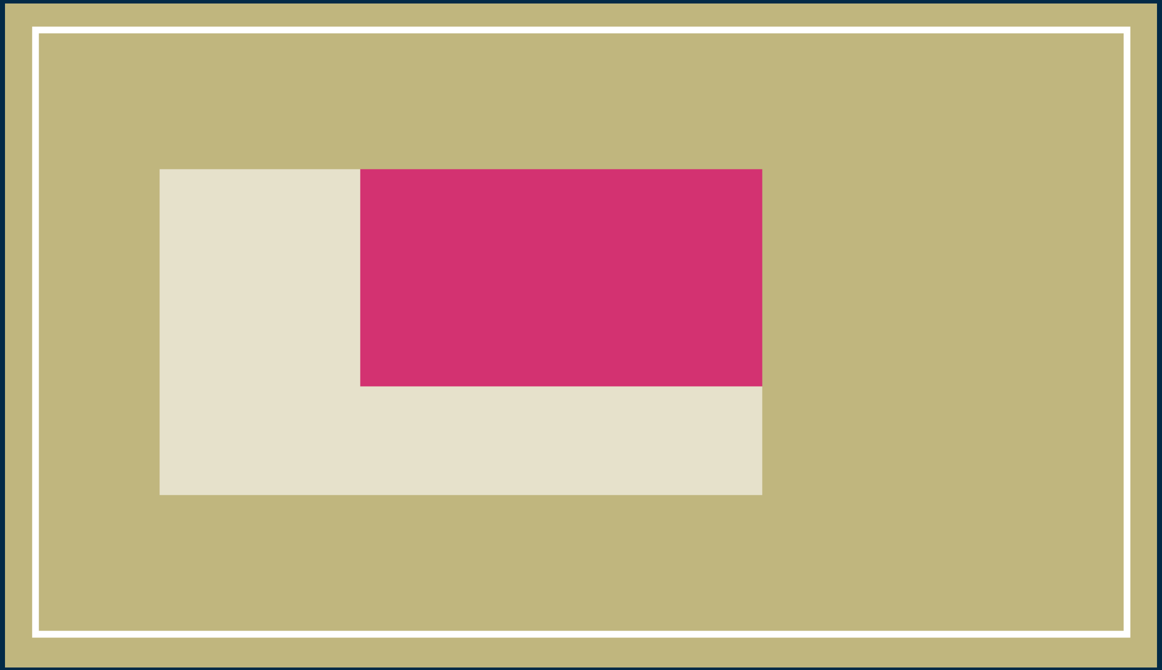
Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

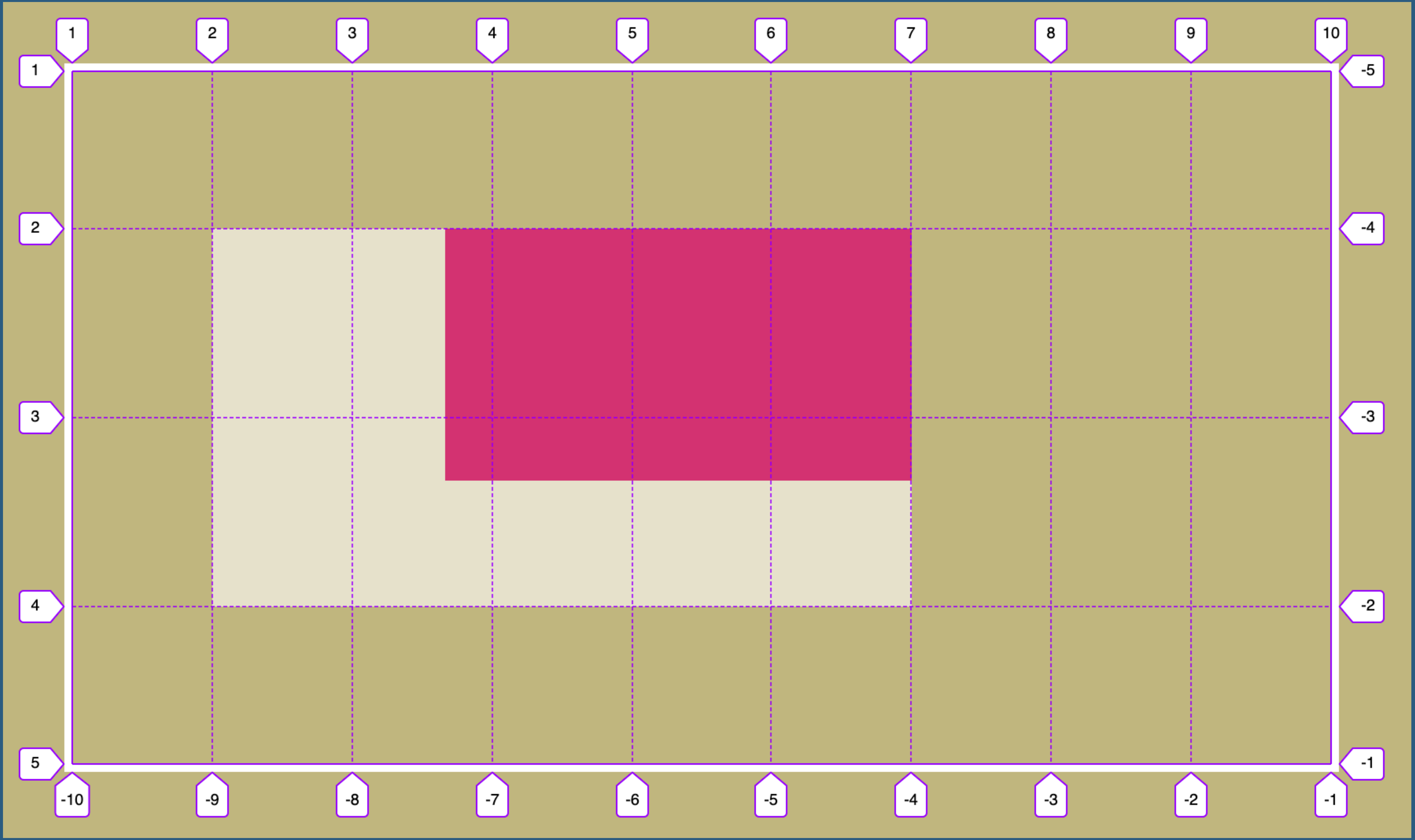
This is the footer.

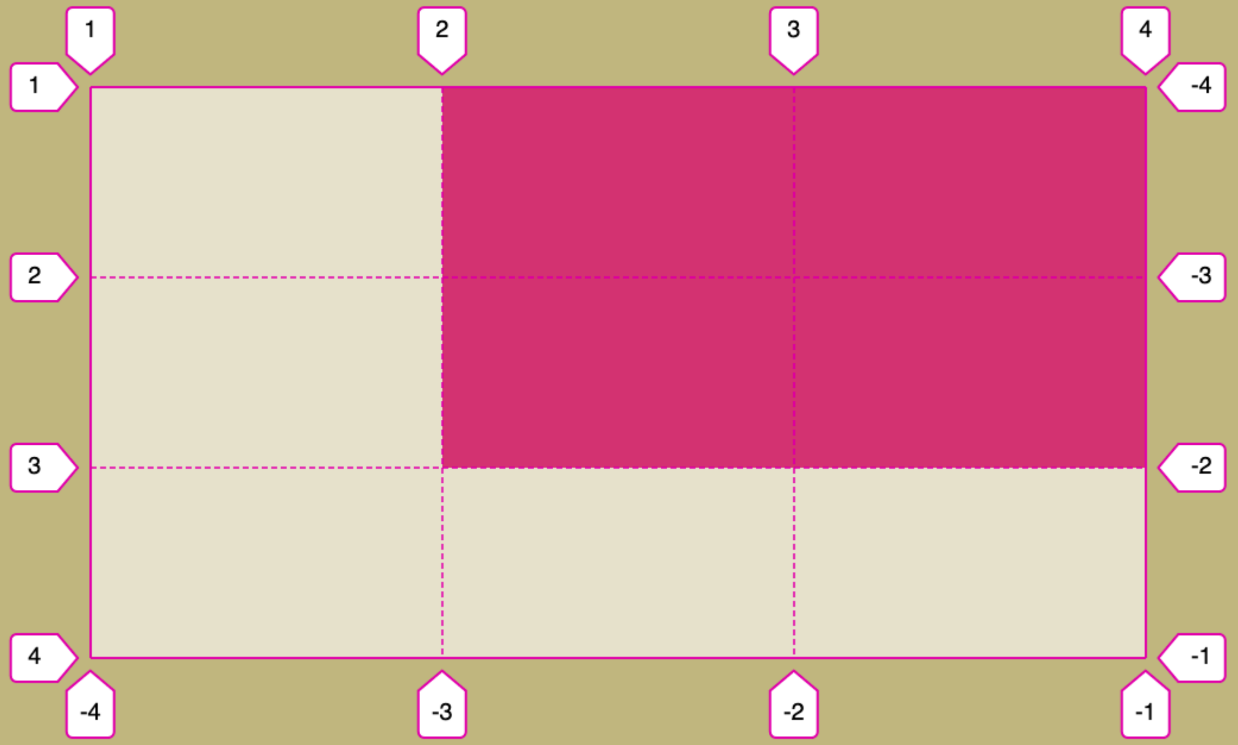
Subgrid

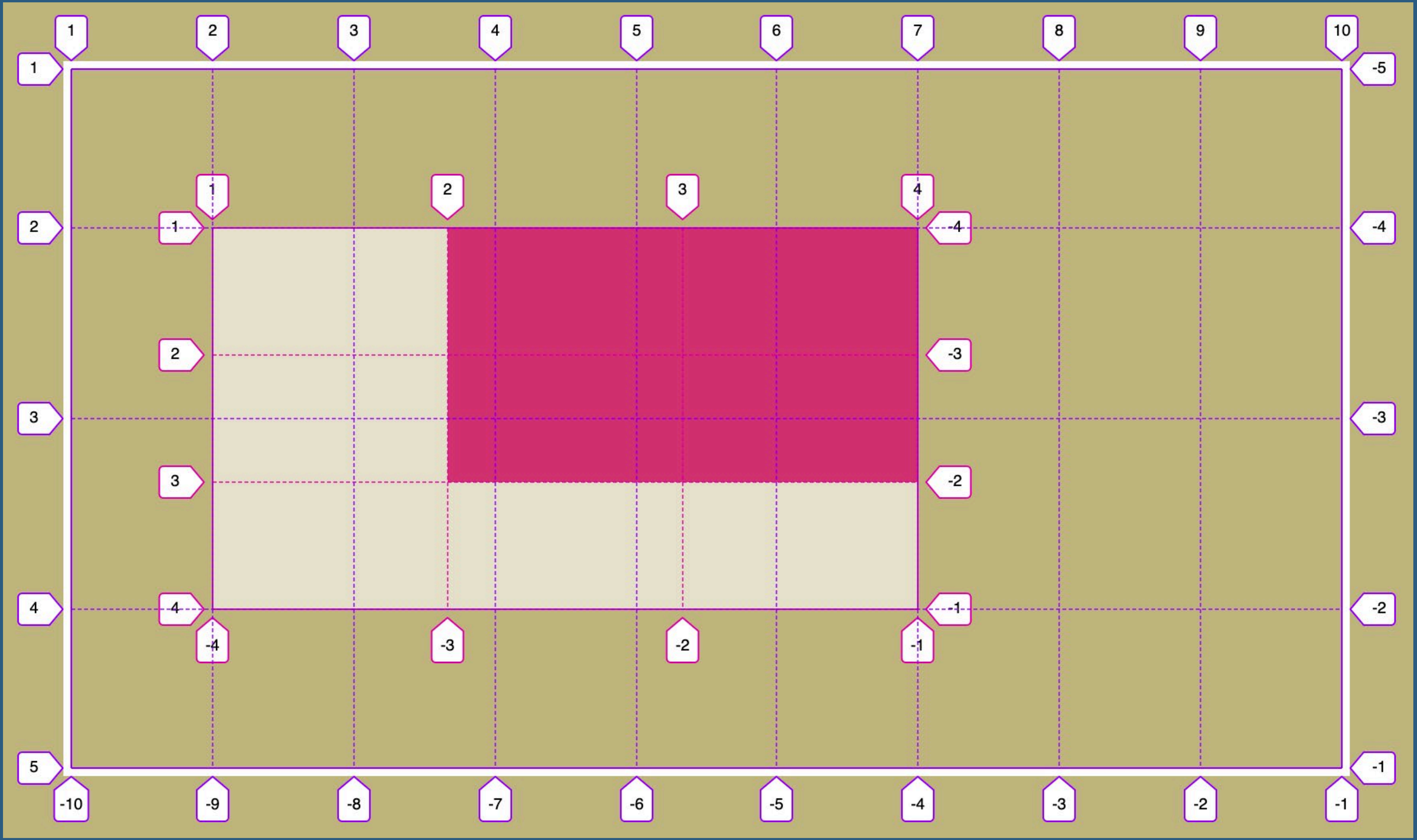
Create a grid on a grid item which uses the grid tracks defined on the parent – for rows, columns or both.

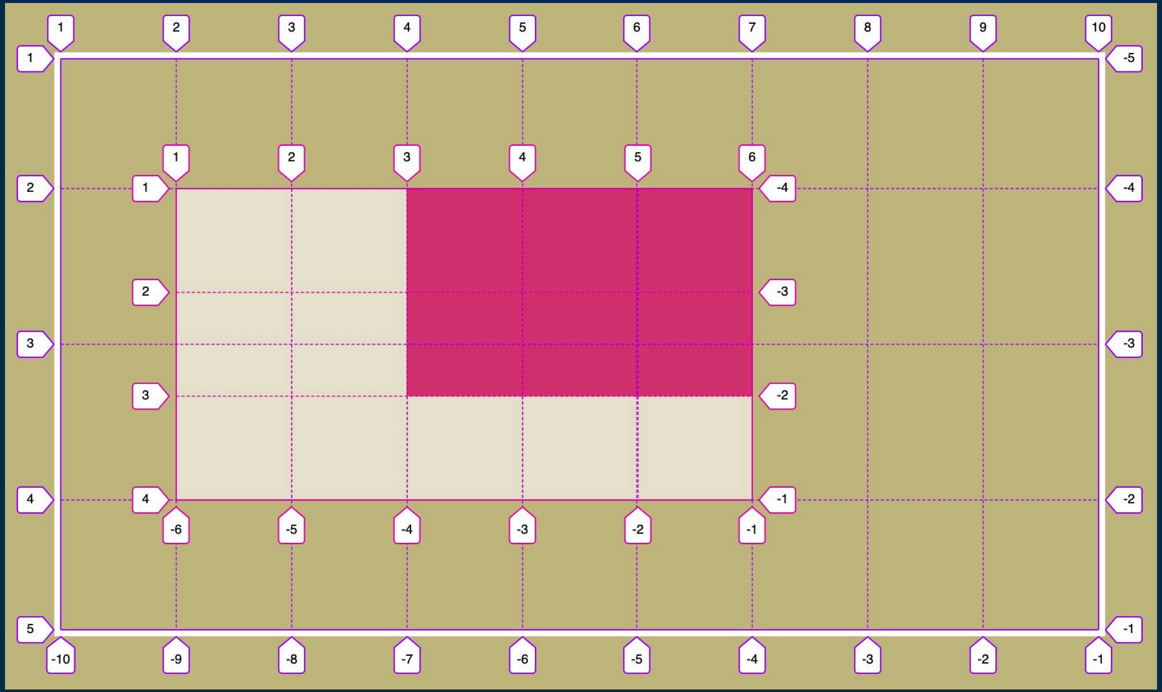


```
.grid {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  grid-template-rows: repeat(4, minmax(100px,
auto));
}

.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: repeat(3, 80px);
}

.subitem {
  grid-column: 2 / 4;
  grid-row: 1 / 4;
}
```

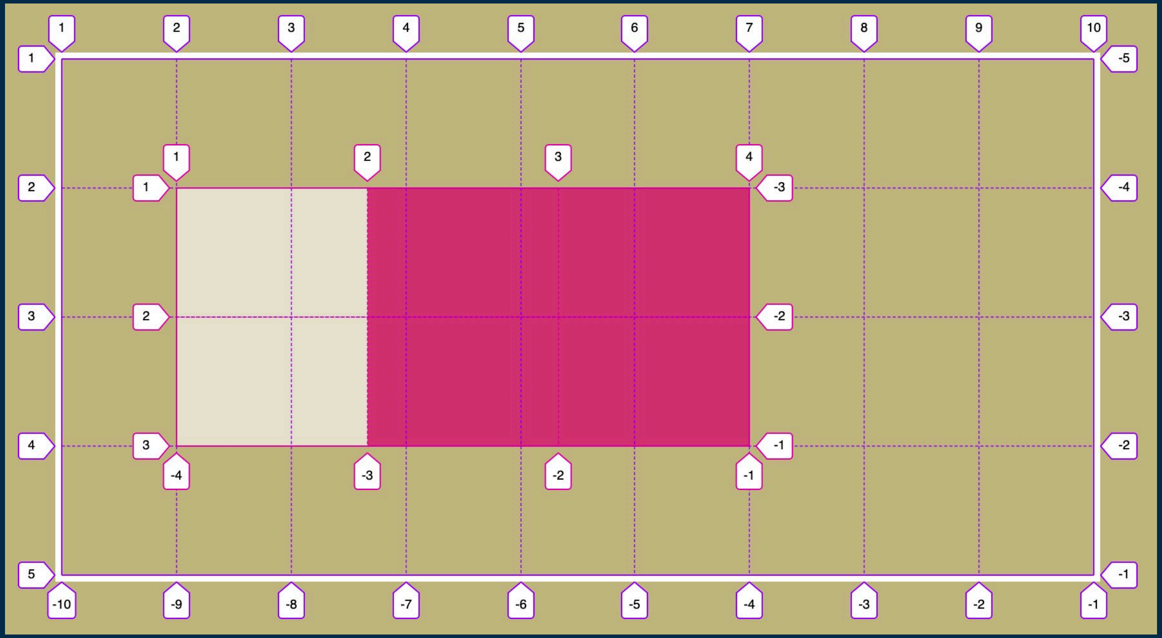

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
-5
-4
-3
-2
-1
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1


1
2
3
4
1
-4
2
-3
3
-2
4
-1
-4
-3
-2
-1

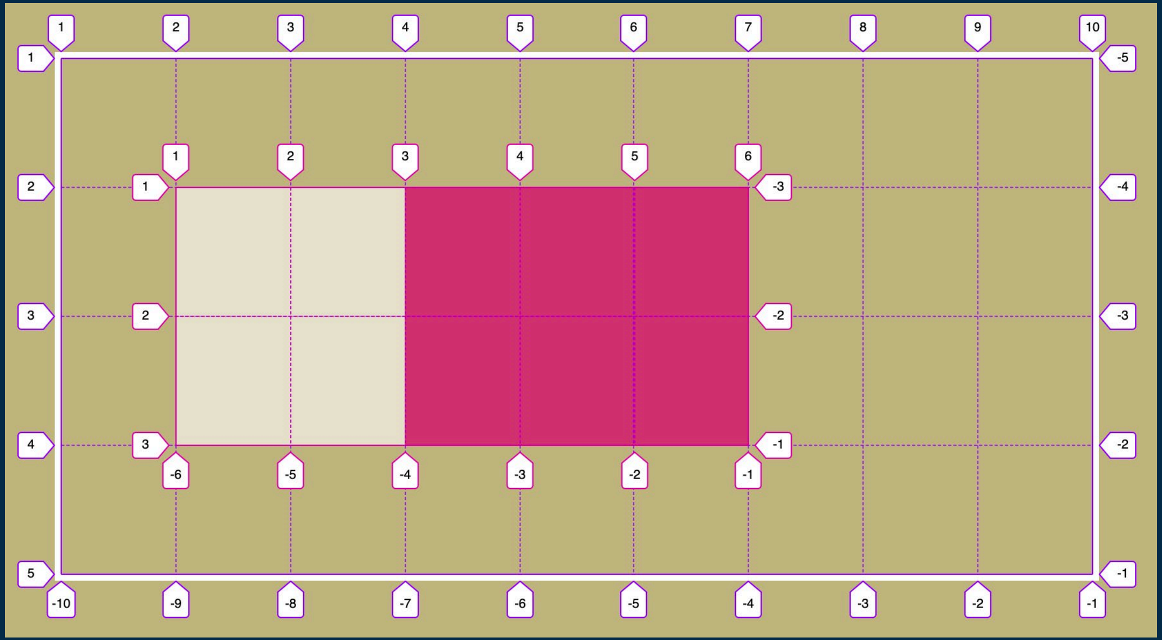

1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
-5
-4
-3
-2
-1
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
1
2
3
4
-4
-3
-2
-1



```
.grid {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  grid-template-rows: repeat(4, minmax(100px,
auto));
}

.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: subgrid;
  grid-template-rows: repeat(3, 80px);
}

.subitem {
  grid-column: 3 / 6;
  grid-row: 1 / 4;
}
```



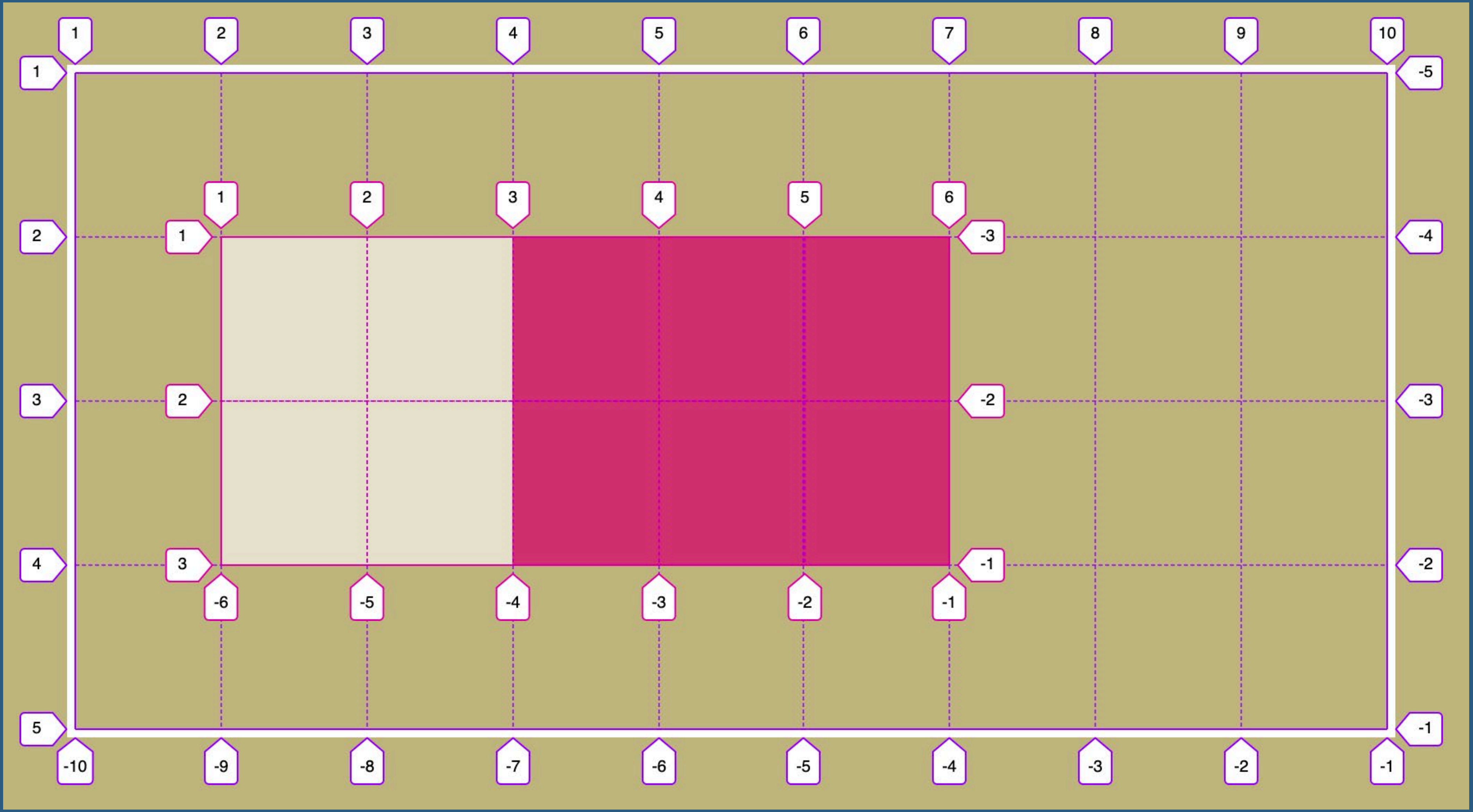
```
.grid {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  grid-template-rows: repeat(4, minmax(100px, 
auto));
}

.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: subgrid;
}

.subitem {
  grid-column: 2 / 4;
  grid-row: 1 / 3;
}
```



```css
.grid {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  grid-template-rows: repeat(4, minmax(100px,
auto));
}

.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: subgrid;
  grid-template-rows: subgrid;
}

.subitem {
  grid-column: 3 / 6;
  grid-row: 1 / 3;
}
```

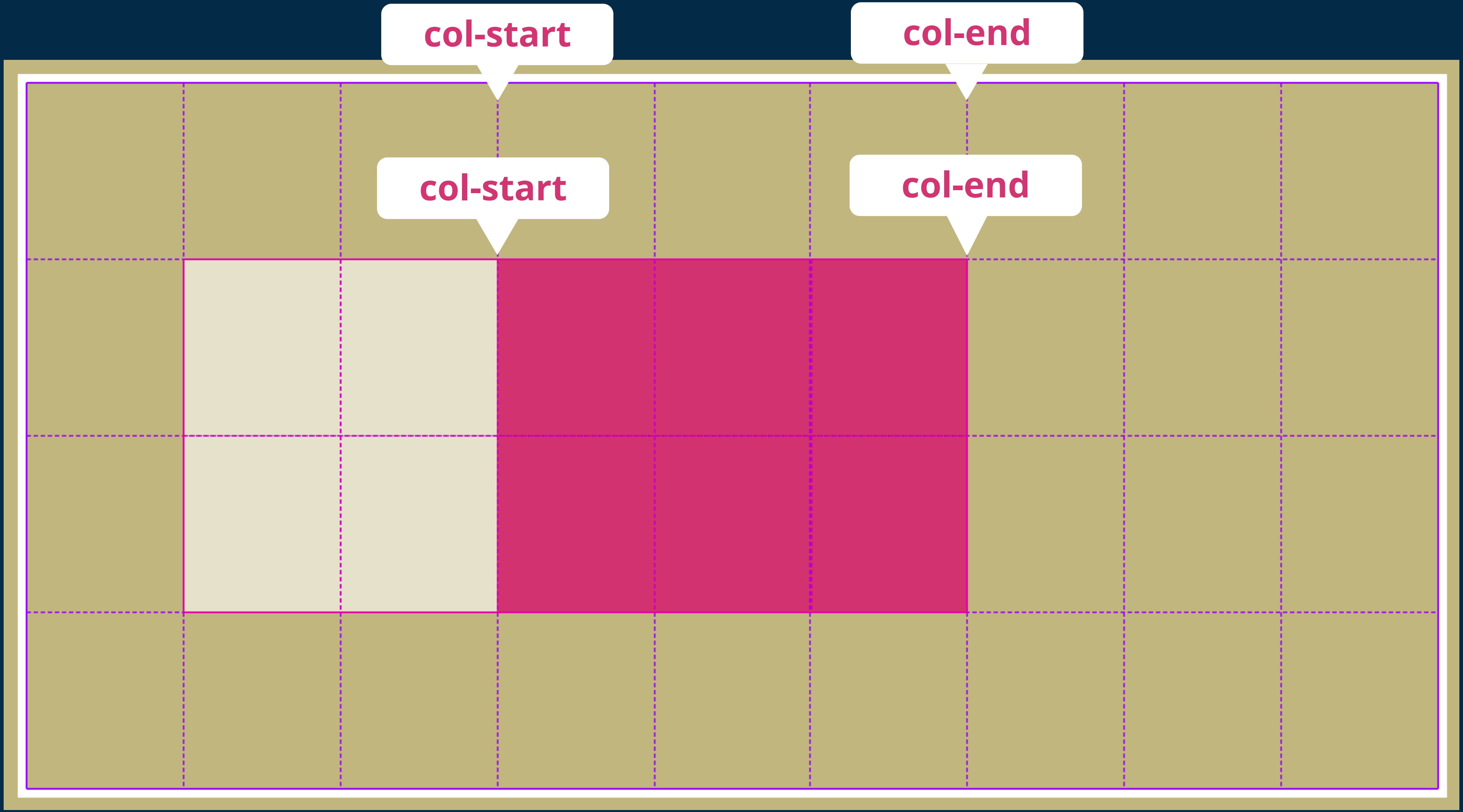
Line **numbers** start from 1 inside the subgrid.

You position child items in the subgrid according to the subgrid line numbering, not those of the parent.


1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
-5
-4
-3
-2
-1
1
2
3
4
5
6
1
2
3
-3
-2
-1
-6
-5
-4
-3
-2
-1
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1

Line **names** on the parent are passed into the subgrid.

If you have named lines on the parent grid they will be passed into the subgrid and added to any names defined there.

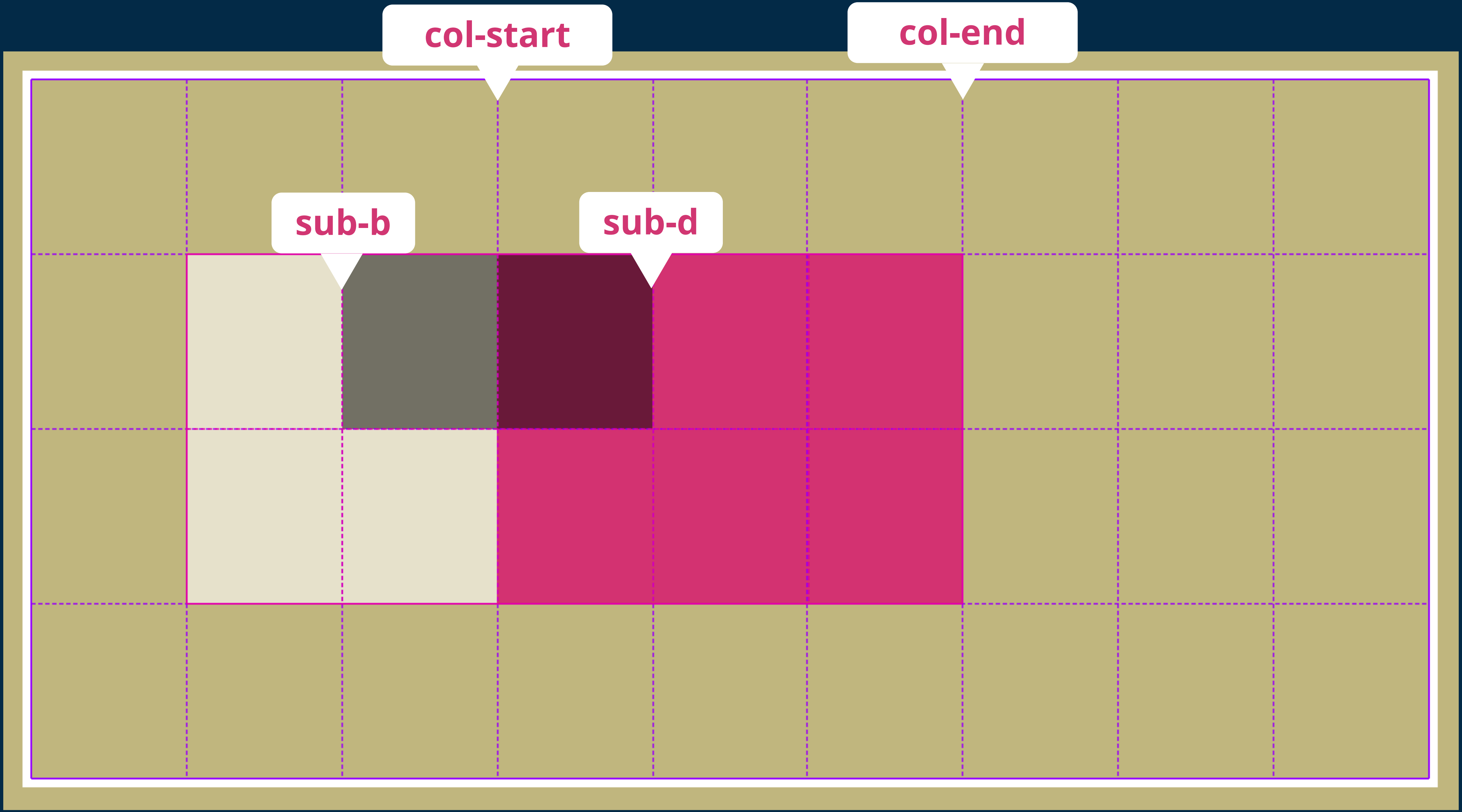

col-start
col-end
col-start
col-end

```css
.grid {
  display: grid;
  grid-template-columns:1fr 1fr 1fr [col-start] 1fr 1fr 1fr [col-end] 1fr 1fr 1fr;
  grid-template-rows: repeat(4, minmax(100px, auto));
}

.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: subgrid;
  grid-template-rows: subgrid;
}

.subitem {
  grid-column: col-start / col-end;
  grid-row: 1 / 3;
}
```

You can **add** named lines to the subgrid.

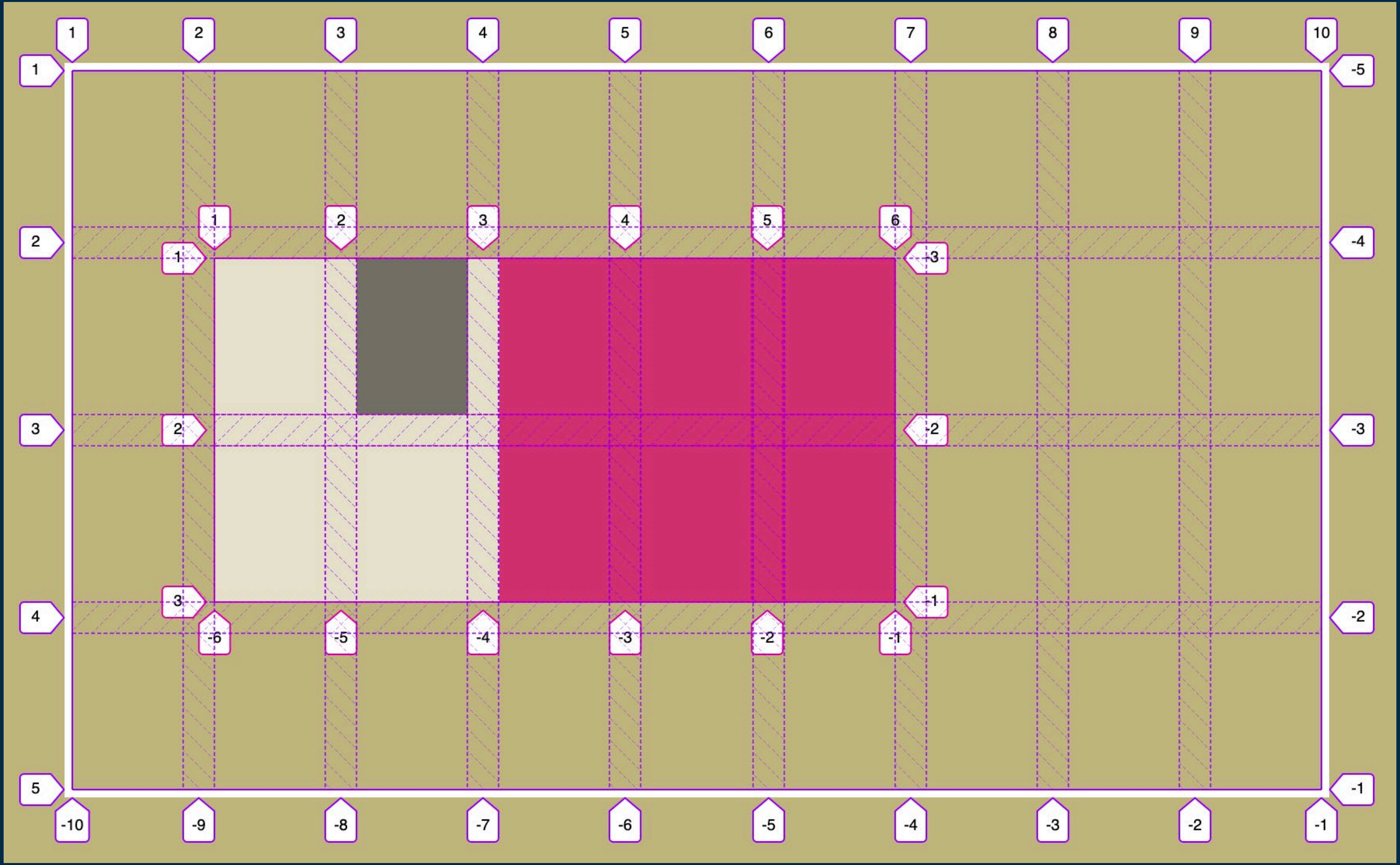
Line names are added after the subgrid keyword.

```
grid-template-columns: subgrid [sub-a] [sub-b];
```

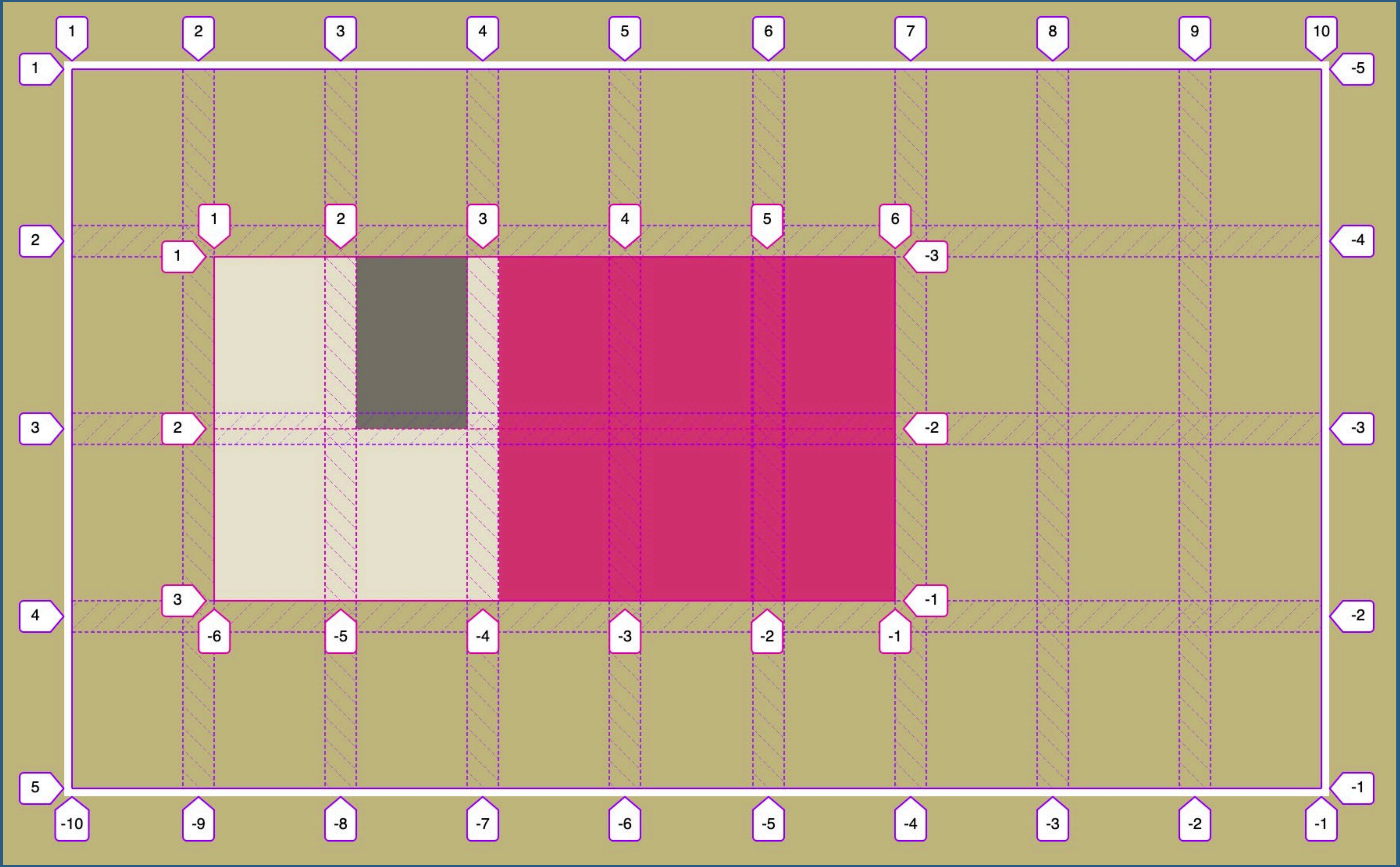

col-start
col-end
sub-b
sub-d

```css
.grid {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr [col-start] 1fr 1fr 1fr [col-end] 1fr 1fr 1fr;
  grid-template-rows: repeat(4, minmax(100px, auto));
}

.item {
  grid-column: 2 / 7;
  grid-row: 2 / 4;
  display: grid;
  grid-template-columns: subgrid [sub-a] [sub-b] [sub-c] [sub-d] [sub-e] [sub-f];
  grid-template-rows: subgrid;
}

.subitem {
  grid-column: col-start / col-end;
  grid-row: 1 / 3;
}

.subitem2 {
  grid-column: sub-b / sub-d;
  grid-row: 1 / 3;
}
```

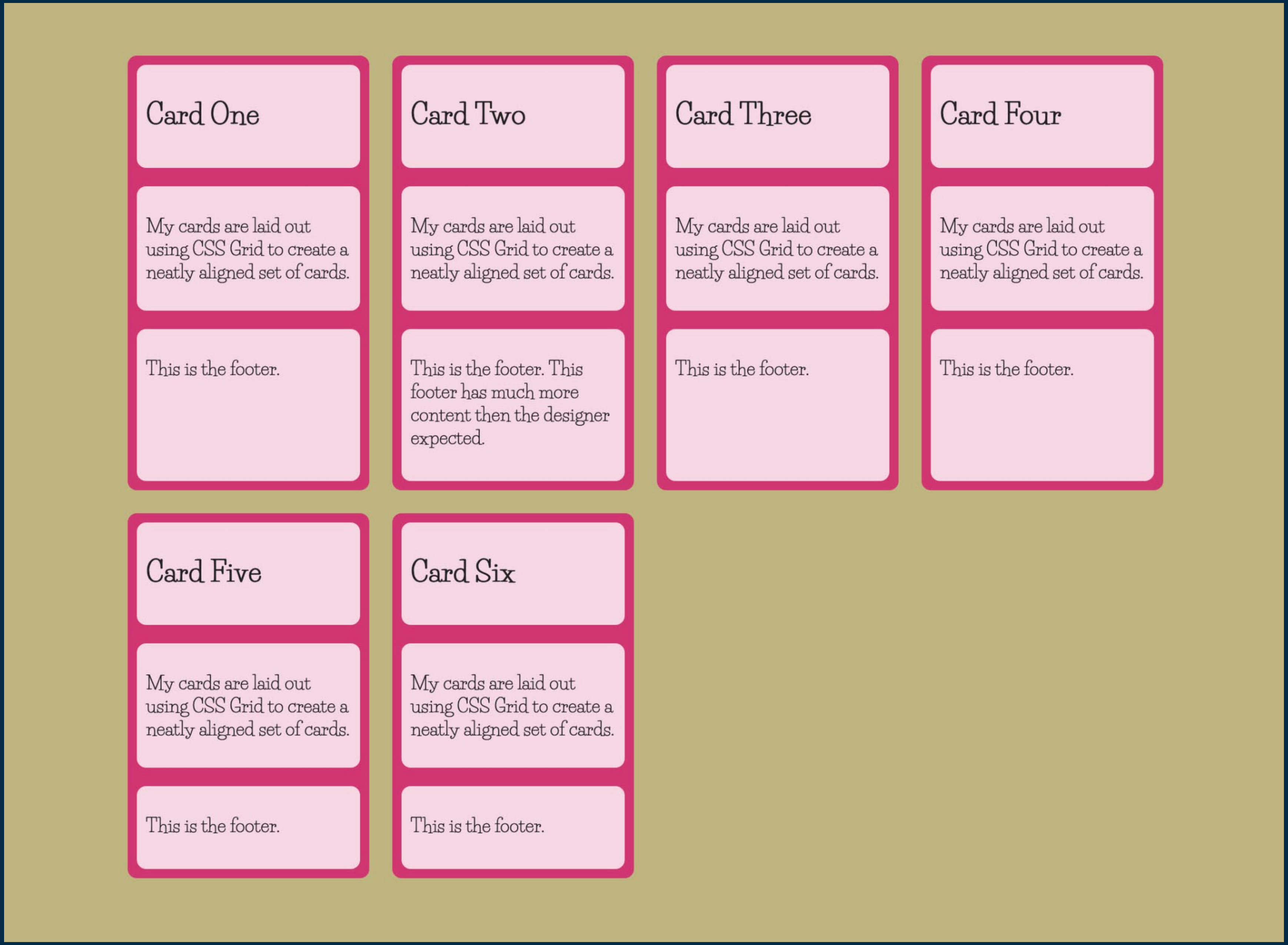
The subgrid inherits the **gaps** from the parent.


1
2
3
4
5
6
7
8
9
10
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1

You can **change** the gaps on the subgrid.


1
2
3
4
5
6
7
8
9
10
-5
-4
-3
-2
-1
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
-6
-5
-4
-3
-2
-1
-3
-2
-1

Sizing of items in the subgrid can change the size of the parent tracks.



Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content then the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

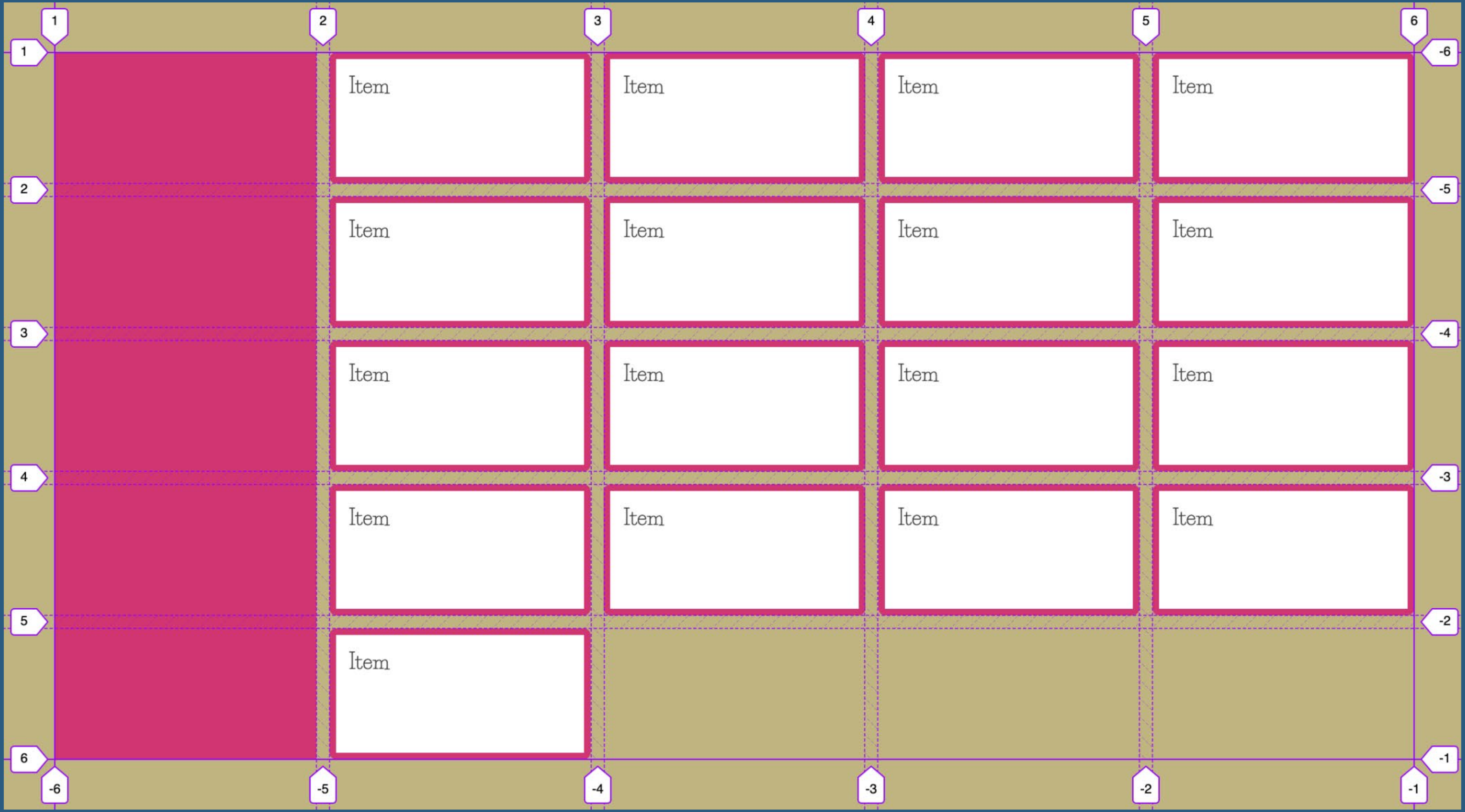
This is the footer.

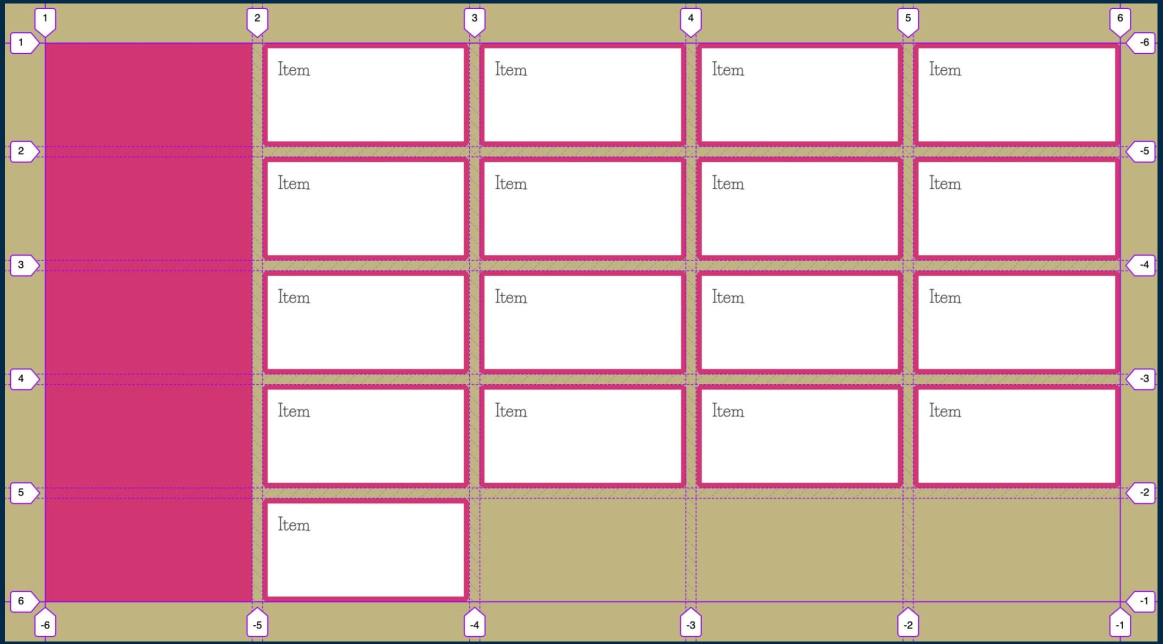
Card Six

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

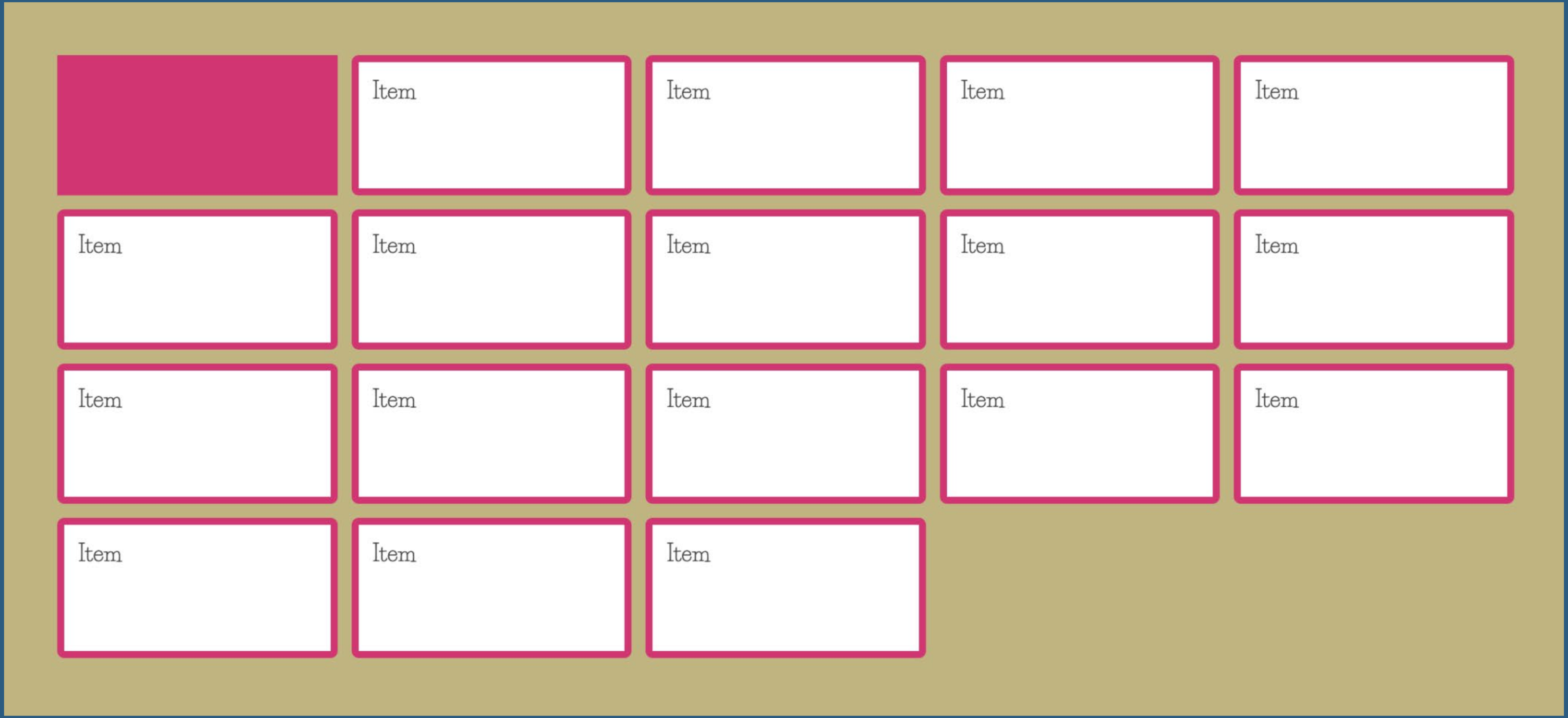
Subgrid enables some previously **difficult patterns**.

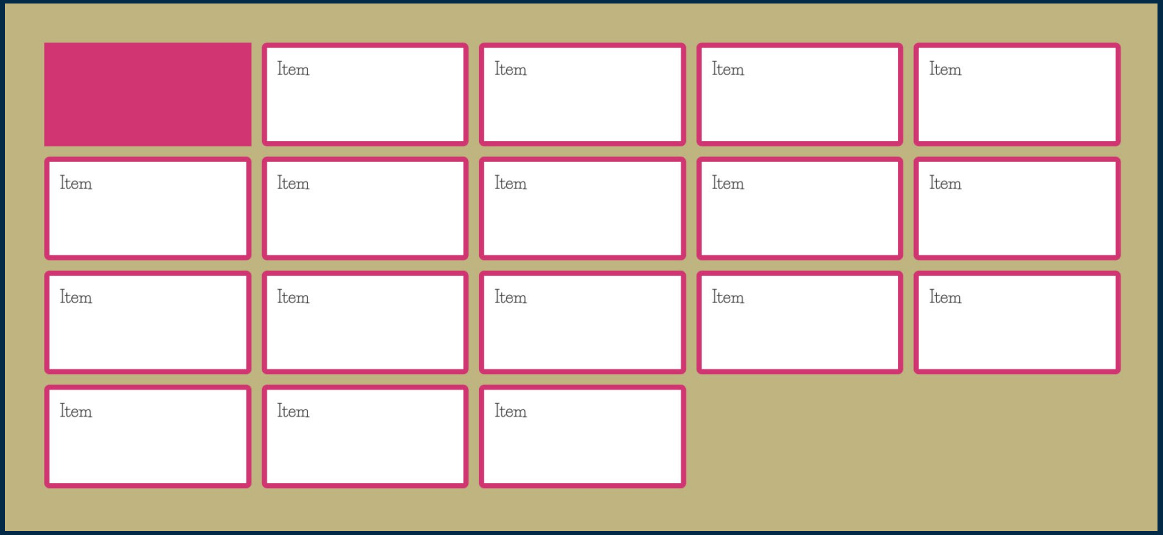

1
2
3
4
5
6
-6
-5
-4
-3
-2
-1
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item



```css
.wrapper {
  display: grid;
  gap: 10px;
  grid-template-columns: repeat(5, 1fr);
  /* 5 explicit rows */
  grid-template-rows:
    repeat(5, minmax(100px, auto));
}

.fullheight {
  background-color: rgb(209,54,114);
  grid-row: 1 / -1;
}
```

Line -1 is the end line of the **explicit grid**.

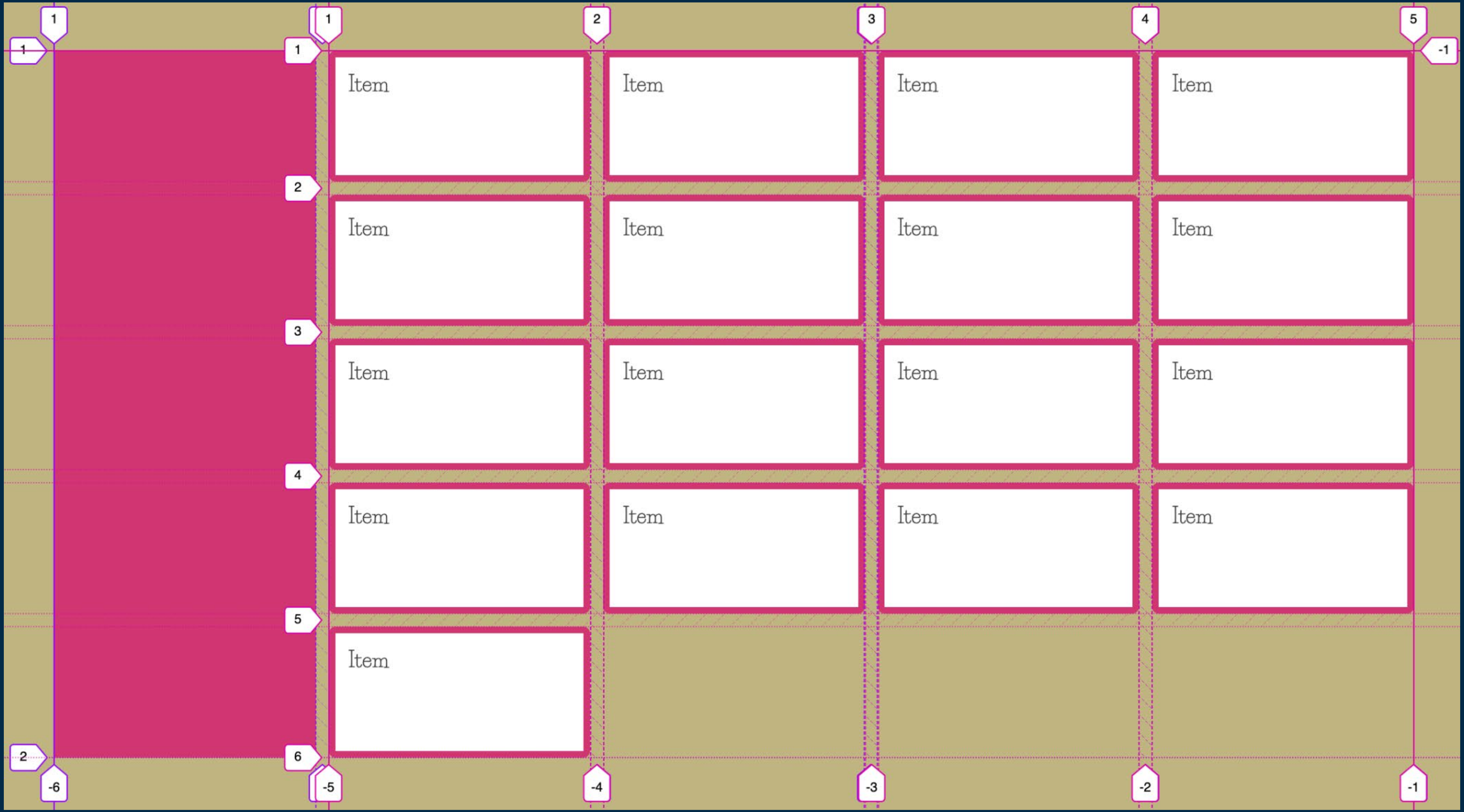

Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item

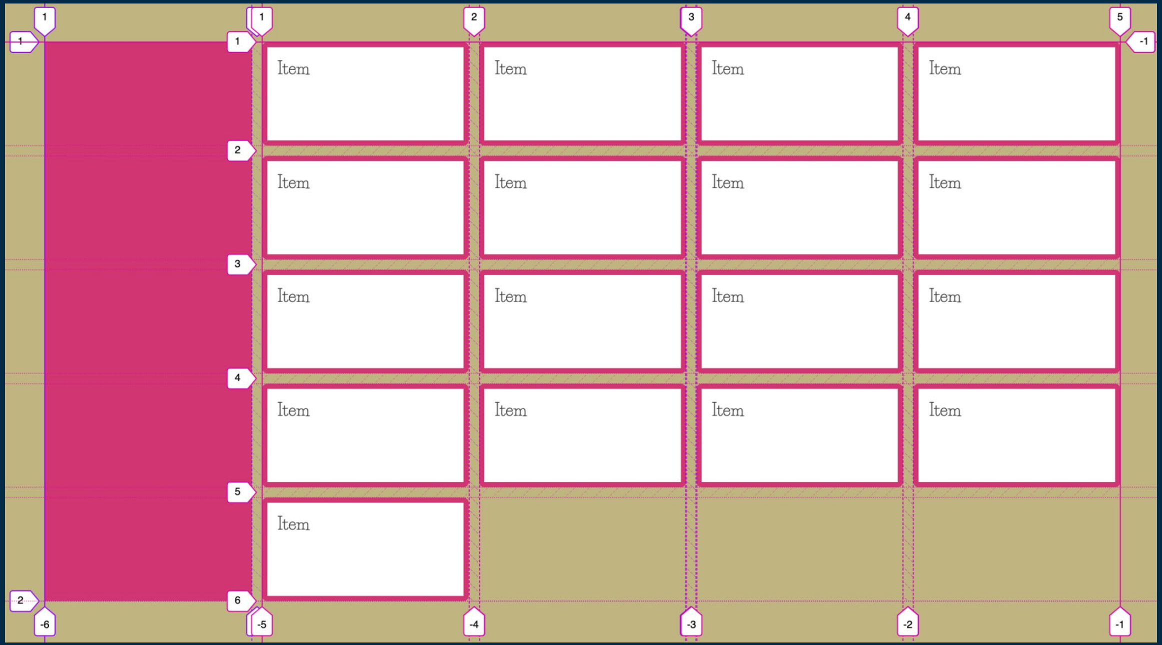


```css
.wrapper {
  display: grid;
  gap: 10px;
  grid-template-columns: repeat(5, 1fr);
  /* no defined explicit rows */
  grid-auto-rows: minmax(100px, auto);
}

.fullheight {
  background-color: rgb(209,54,114);
  grid-row: 1 / -1;
}
```

We can’t target the end line of the **implicit grid**.

Place the items in a container which uses a subgrid for columns.

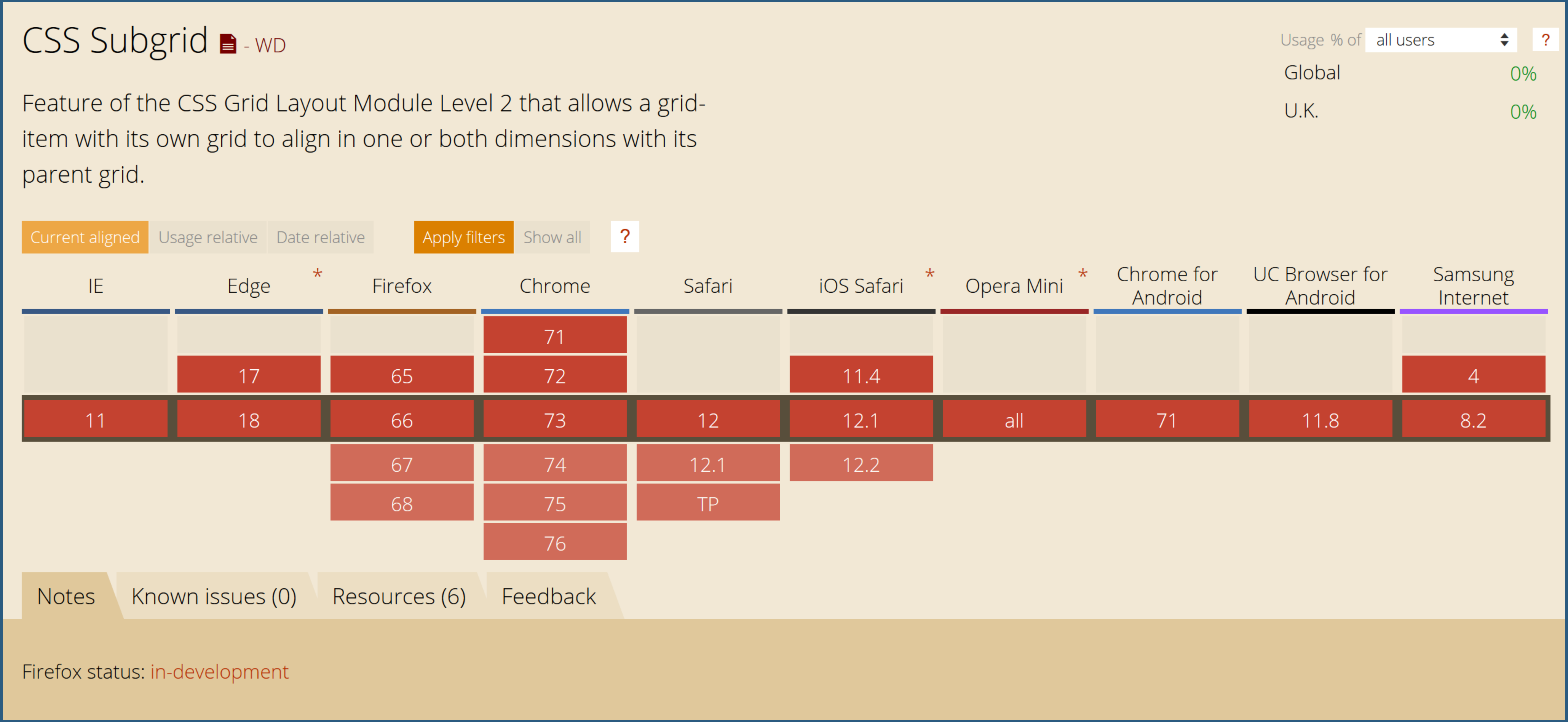

1
1
1
2
3
4
5
-1
1
2
3
4
5
6
2
-6
-5
-4
-3
-2
-1
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item
Item



```
.wrapper {
  display: grid;
  gap: 10px;
  grid-template-columns: repeat(5, 1fr);
  /* no defined explicit rows */
  grid-auto-rows: minmax(100px, auto);
}
.items {
  grid-column: 2 / -1;
  display: grid;
  grid-template-columns: subgrid;
  grid-auto-rows: minmax(100px, auto);
}
.fullheight {
  background-color: rgb(209,54,114);
  grid-row: 1 / -1;
}
```

Now the bad news

There are currently no shipped implementations of subgrid.



CSS Subgrid - WD

Usage % of all users ?

Global	0%
U.K.	0%

Feature of the CSS Grid Layout Module Level 2 that allows a grid-item with its own grid to align in one or both dimensions with its parent grid.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

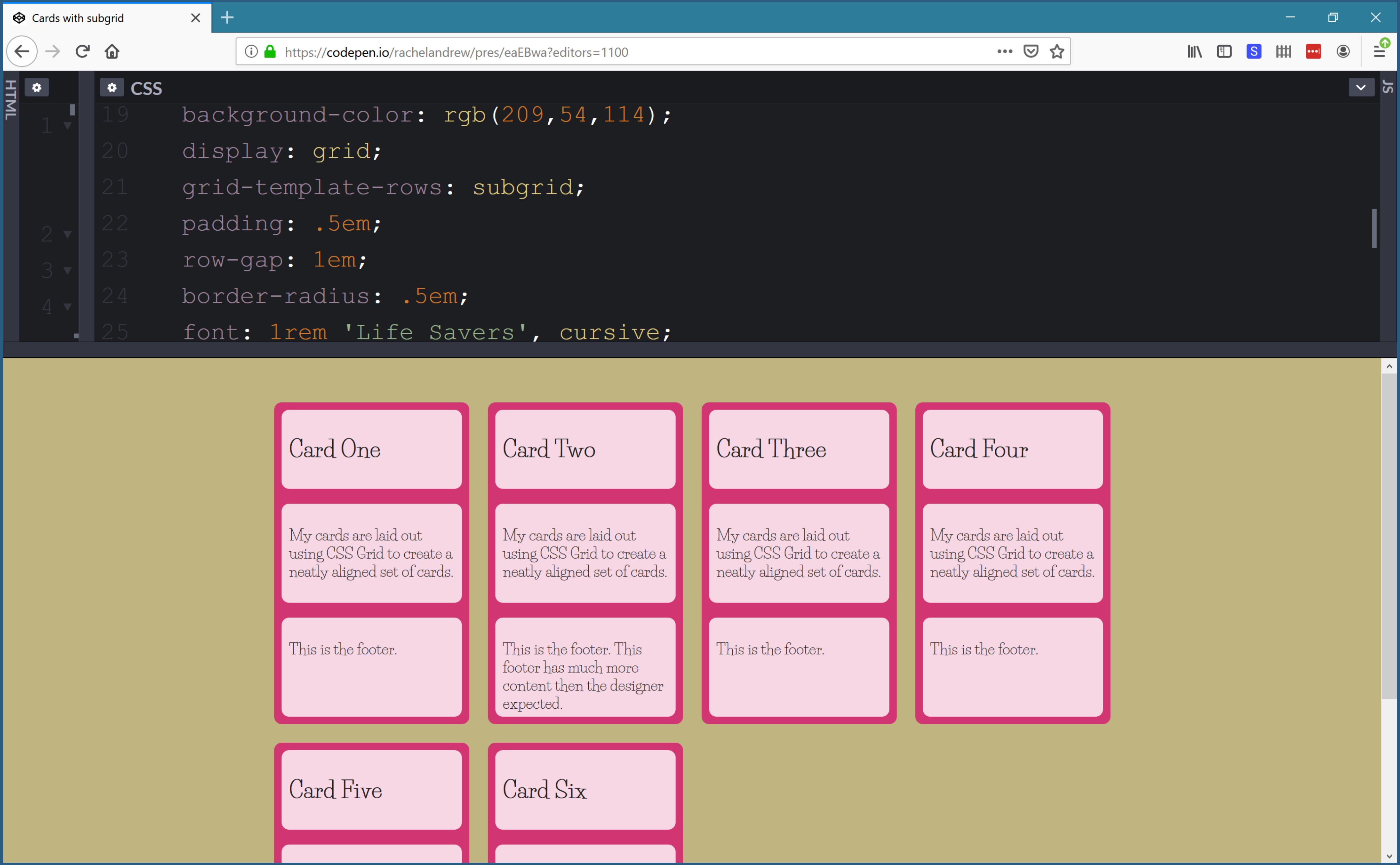
IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			71						
	17	65	72		11.4				4
11	18	66	73	12	12.1	all	71	11.8	8.2
		67	74	12.1	12.2				
		68	75	TP					
			76						

Notes | Known issues (0) | Resources (6) | Feedback

Firefox status: in-development

Slightly better news

Firefox have shipped subgrid in Nightly!


Cards with subgrid

https://codepen.io/rachelandrew/pres/eaEBwa?editors=1100

CSS

```
  background-color: rgb(209,54,114);
  display: grid;
  grid-template-rows: subgrid;
  padding: .5em;
  row-gap: 1em;
  border-radius: .5em;
  font: 1rem 'Life Savers', cursive;
```

Card One

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Two

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer. This footer has much more content then the designer expected.

Card Three

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

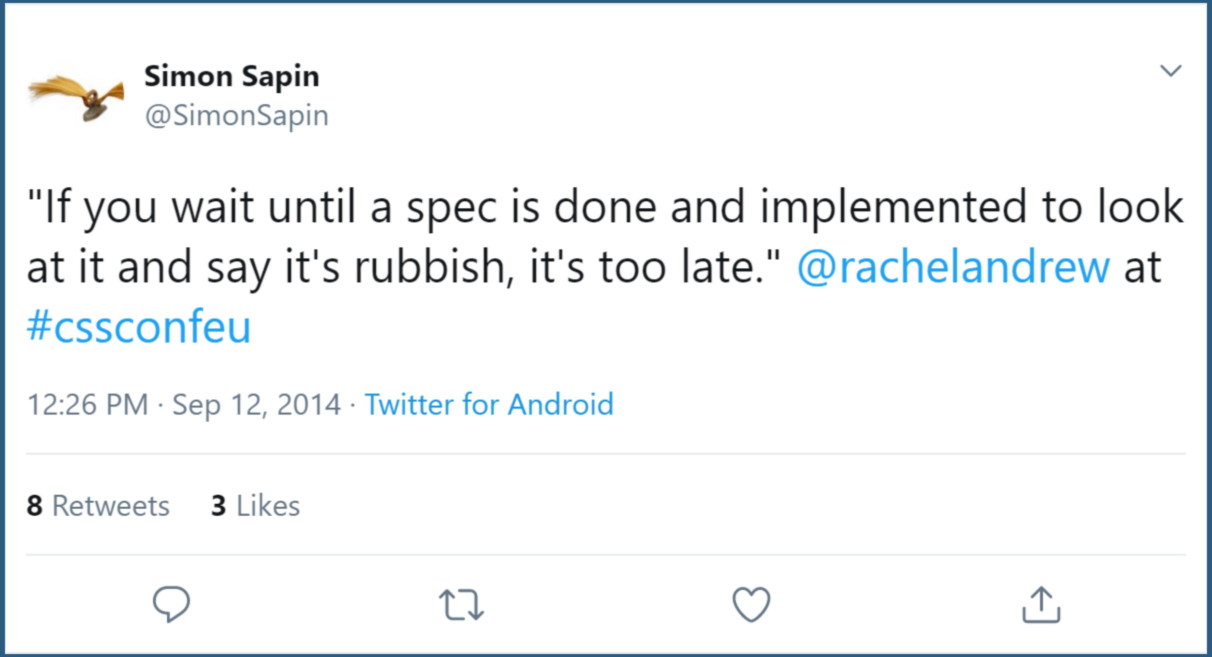
Card Four

My cards are laid out using CSS Grid to create a neatly aligned set of cards.

This is the footer.

Card Five

Card Six



Simon Sapin
@SimonSapin

"If you wait until a spec is done and implemented to look at it and say it's rubbish, it's too late." @rachelandrew at #cssconfeu

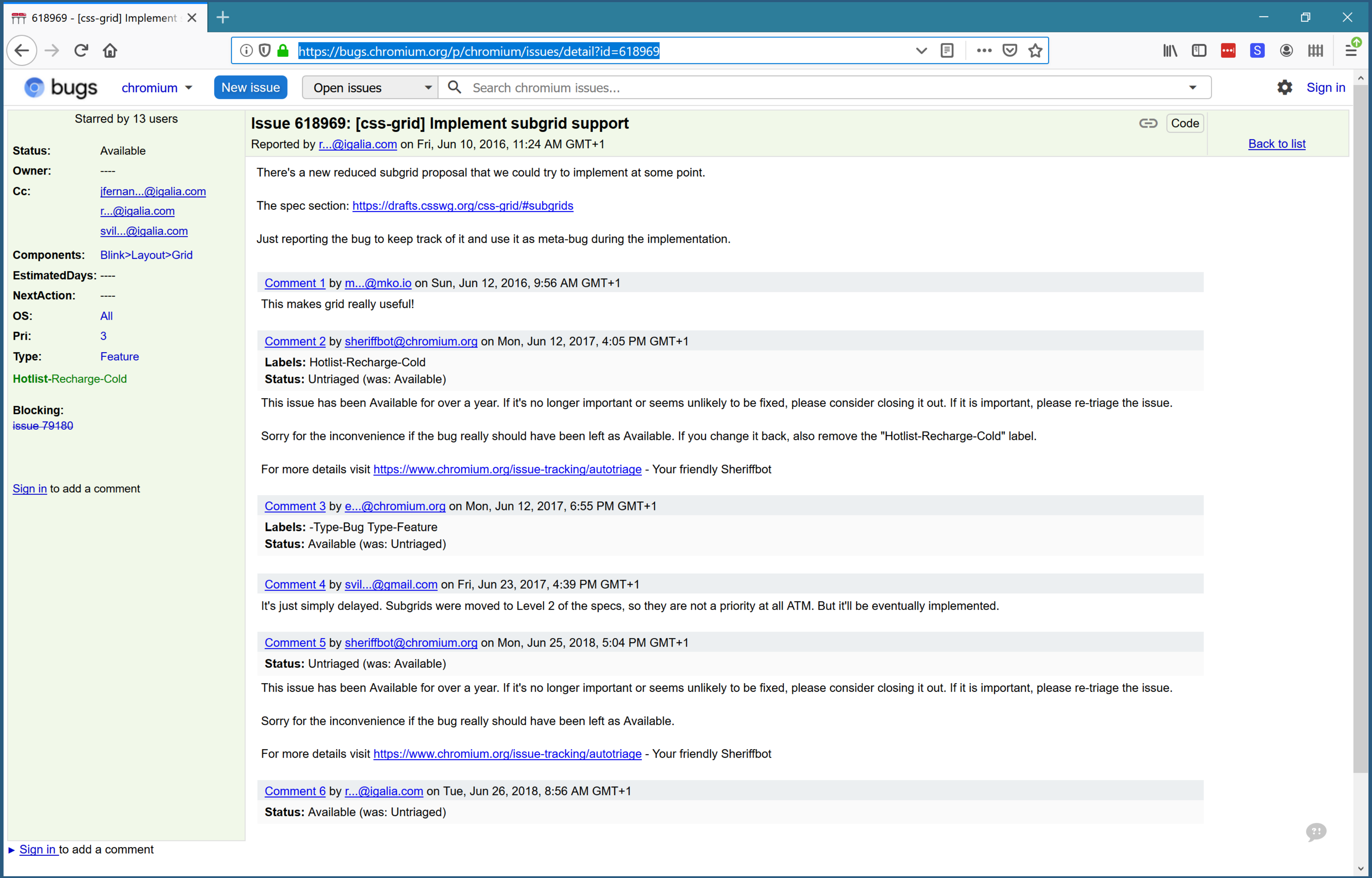
12:26 PM · Sep 12, 2014 · Twitter for Android

8 Retweets **3** Likes

Fewer rendering engines

fewer places where new ground can be broken.

Tell browsers what you want in the platform.


618969 - [css-grid] Implement
https://bugs.chromium.org/p/chromium/issues/detail?id=618969
bugs
chromium
New issue
Open issues
Search chromium issues...
Sign in

Starred by 13 users

Status:	Available
Owner:	----
Cc:	jfernan...@igalia.com r...@igalia.com svil...@igalia.com
Components:	Blink>Layout>Grid
EstimatedDays:	----
NextAction:	----
OS:	All
Pri:	3
Type:	Feature

Hotlist-Recharge-Cold

Blocking:
~~issue 79180~~

Sign in to add a comment

Issue 618969: [css-grid] Implement subgrid support

Reported by r...@igalia.com on Fri, Jun 10, 2016, 11:24 AM GMT+1

Code

Back to list

There's a new reduced subgrid proposal that we could try to implement at some point.

The spec section: https://drafts.csswg.org/css-grid/#subgrids

Just reporting the bug to keep track of it and use it as meta-bug during the implementation.

Comment 1 by m...@mko.io on Sun, Jun 12, 2016, 9:56 AM GMT+1

This makes grid really useful!

Comment 2 by sheriffbot@chromium.org on Mon, Jun 12, 2017, 4:05 PM GMT+1

Labels: Hotlist-Recharge-Cold
Status: Untriaged (was: Available)

This issue has been Available for over a year. If it's no longer important or seems unlikely to be fixed, please consider closing it out. If it is important, please re-triage the issue.

Sorry for the inconvenience if the bug really should have been left as Available. If you change it back, also remove the "Hotlist-Recharge-Cold" label.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 3 by e...@chromium.org on Mon, Jun 12, 2017, 6:55 PM GMT+1

Labels: -Type-Bug Type-Feature
Status: Available (was: Untriaged)

Comment 4 by svil...@gmail.com on Fri, Jun 23, 2017, 4:39 PM GMT+1

It's just simply delayed. Subgrids were moved to Level 2 of the specs, so they are not a priority at all ATM. But it'll be eventually implemented.

Comment 5 by sheriffbot@chromium.org on Mon, Jun 25, 2018, 5:04 PM GMT+1

Status: Untriaged (was: Available)

This issue has been Available for over a year. If it's no longer important or seems unlikely to be fixed, please consider closing it out. If it is important, please re-triage the issue.

Sorry for the inconvenience if the bug really should have been left as Available.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 6 by r...@igalia.com on Tue, Jun 26, 2018, 8:56 AM GMT+1

Status: Available (was: Untriaged)

▸ Sign in to add a comment

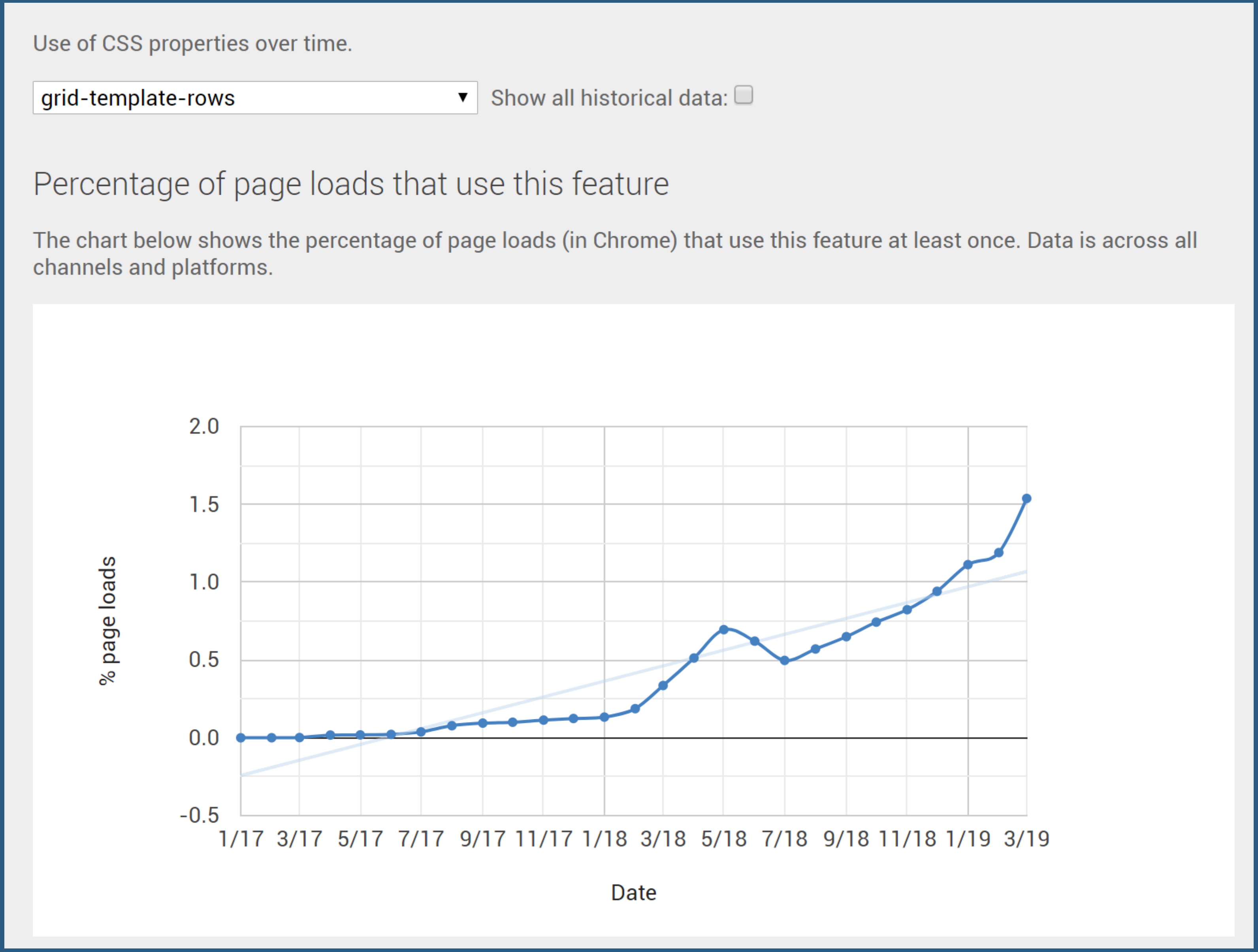
https://bugs.chromium.org/p/chromium/issues/detail?id=618969

Write about features you want to see in browsers

Write up your use cases, the problems having the feature will solve.

Use new features

Browsers are watching.



Use of CSS properties over time.

grid-template-rows ▾ Show all historical data: ☐

Percentage of page loads that use this feature

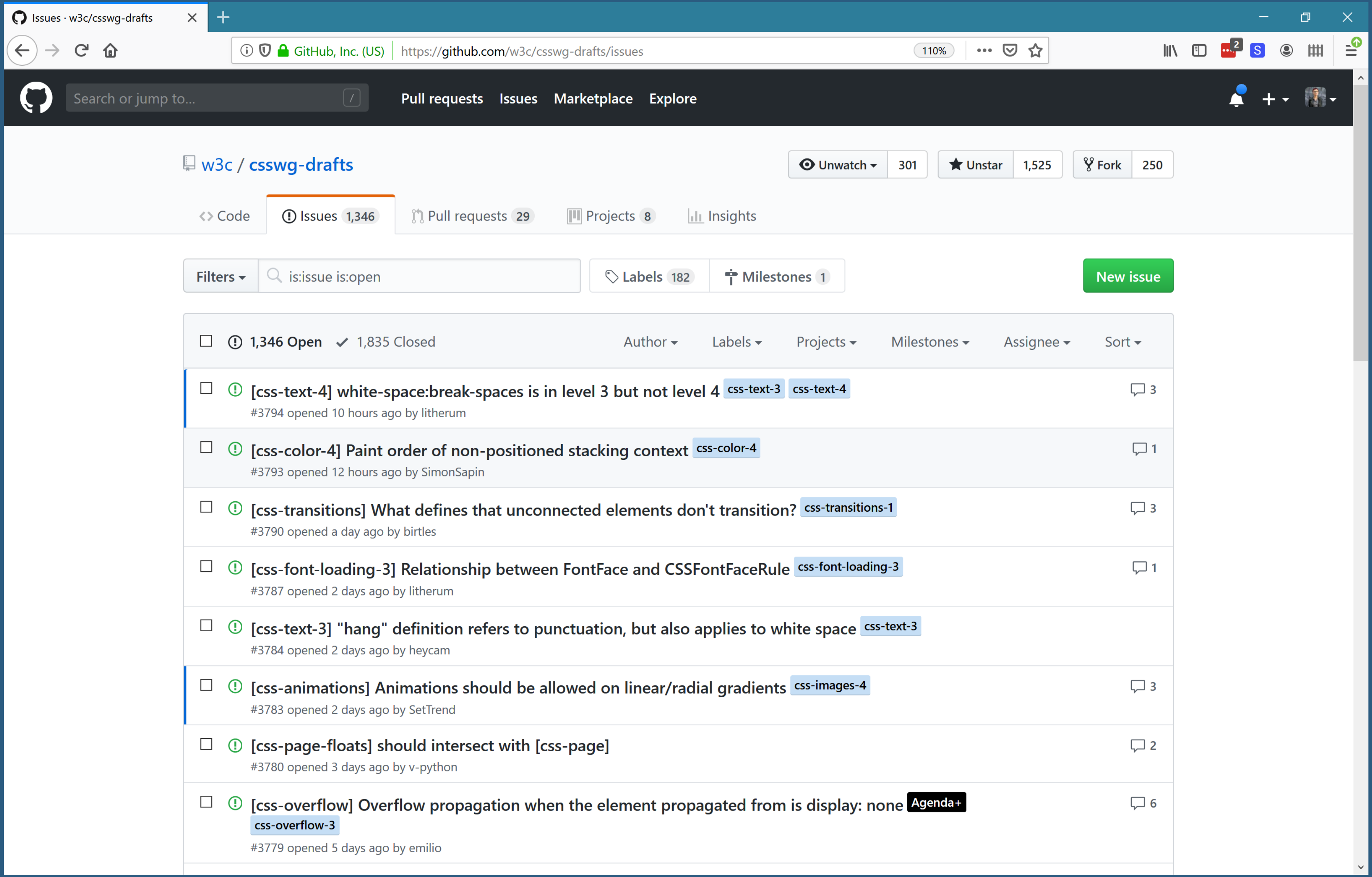
The chart below shows the percentage of page loads (in Chrome) that use this feature at least once. Data is across all channels and platforms.

Use new features when they are behind a flag

You get to beta test the web platform!

Give feedback to the CSS Working Group

https://github.com/w3c/csswg-drafts/issues



Issues · w3c/csswg-drafts

https://github.com/w3c/csswg-drafts/issues

w3c / csswg-drafts

Unwatch 301 · Unstar 1,525 · Fork 250

Code · Issues 1,346 · Pull requests 29 · Projects 8 · Insights

Filters · is:issue is:open · Labels 182 · Milestones 1 · New issue

1,346 Open ✓ 1,835 Closed · Author · Labels · Projects · Milestones · Assignee · Sort

- [css-text-4] white-space:break-spaces is in level 3 but not level 4 `css-text-3` `css-text-4` — 3
  #3794 opened 10 hours ago by litherum
- [css-color-4] Paint order of non-positioned stacking context `css-color-4` — 1
  #3793 opened 12 hours ago by SimonSapin
- [css-transitions] What defines that unconnected elements don't transition? `css-transitions-1` — 3
  #3790 opened a day ago by birtles
- [css-font-loading-3] Relationship between FontFace and CSSFontFaceRule `css-font-loading-3` — 1
  #3787 opened 2 days ago by litherum
- [css-text-3] "hang" definition refers to punctuation, but also applies to white space `css-text-3`
  #3784 opened 2 days ago by heycam
- [css-animations] Animations should be allowed on linear/radial gradients `css-images-4` — 3
  #3783 opened 2 days ago by SetTrend
- [css-page-floats] should intersect with [css-page] — 2
  #3780 opened 3 days ago by v-python
- [css-overflow] Overflow propagation when the element propagated from is display: none `Agenda+` `css-overflow-3` — 6
  #3779 opened 5 days ago by emilio

Participate in the web platform

Or you are leaving your future as a designer or developer in the hands of the very few who do.

Thank you

https://noti.st/rachelandrew/i6gUcF/