

@rachelandrew @ DevFest Nantes

START USING CSS GRID LAYOUT TODAY

- ▶ CSS Working Group Invited Expert
- ▶ Google Developer Expert
- ▶ co-founder Perch CMS
- ▶ Old Nerd.
- ▶ You can find me in most places as [@rachelandrew](#) you can email me@rachelandrew.co.uk or check out my site at <https://rachelandrew.co.uk>



March 2017



March 2017



March 2017



March 2017



March 2017



October 2017

Laying out the future with grid and flexbox

- ▶ What is grid & why is it different to flexbox?
- ▶ How do I get started using grid in production?
- ▶ What about old browsers?
- ▶ How can we help encourage browsers to give us cool new stuff?

CSS Grid Layout

**Why not use
flexbox?**

Flexbox is for one-dimensional layout



1

2

3

4

5

6

7

8

9

10

11

12

13

Grid is for two-dimensional layout



1

2

3

4

5

6

7

8

9

10

11

12

13

Grid minmax() and auto-fill

Creating a flexible number of flexible tracks, with a little bit of grid spec magic.

<http://codepen.io/rachelandrew/pen/evjdLM>

```
.grid {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns: repeat(auto-  
fill, minmax(200px, 1fr));  
}
```


If you are adding widths to all your flex items, you probably need grid.

Flexbox

Using space-between

<https://codepen.io/rachelandrew/pen/rzXXJY>

```
.example {  
  display: flex;  
  justify-content: space-between;  
  margin: 30px;  
}
```


Tomato

Apple

Curly Kale

Cabbage

Parsnip

Flexbox

Allowing items to grow and shrink from a flex-basis of auto.

<https://codepen.io/rachelandrew/pen/MvNNaj>

```
.example {  
  display: flex;  
  flex-wrap: wrap;  
  margin: 30px;  
}
```

```
.example > div {  
  flex: 1 1 auto;  
}
```

Tomato

Apple

Curly Kale

Cabbage

Parsnip

Orange

Kiwi

Lettuce

Pear

Grid works from the container in

Grid

Define column tracks. Items are constrained by those tracks.

<https://codepen.io/rachelandrew/pen/prMMLLe>

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

1fr

Tomato

Parsnip

Pear

1fr

Apple

Orange

1fr

Curly Kale

Kiwi

1fr

Cabbage

Lettuce

Grid

To make some tracks larger than others, we do that when defining the tracks on the **container** not on the item itself.

<https://codepen.io/rachelandrew/pen/LjwwgM>

```
.example {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns: 2fr 1fr 2fr 1fr;  
  margin: 20px;  
}
```

2fr

Tomato

Parsnip

Pear

1fr

Apple

Orange

2fr

Curly Kale

Kiwi

1fr

Cabbage

Lettuce

Grid

Key words based on content size,
change the size of the entire track.

- min-content
- max-content
- fit-content

<https://codepen.io/rachelandrew/pen/NaLExq>

```
.example {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns: min-content max-  
content fit-content(200px);  
  margin: 20px;  
}
```

**I am
sized
with
min-
content**

I am sized with max-content

**This column
uses fit-
content with
a maximum
of 200px**

Apple

Parsnip

Orange

**Curly
Kale**

Lettuce

Pear

Other layout methods start with
the *item*.

A float grid

The float property and widths are added to the items.

```
.box {  
    float: left;  
    width: 33.3333%;  
}
```

inline-block grid

The display property is set to inline-block and width is added to the item.

```
.box {  
    display: inline-block;  
    width: 33.3333%;  
}
```

Flex grid

We add `display: flex` to the container however to make a grid out of flex items we need to use the flex properties in the items.

```
.container {  
  display: flex;  
}  
  
.box {  
  flex: 0 0 33.3333%;  
}
```


Grid Layout

With CSS Grid Layout we create the grid on the parent element. We don't need to add properties to the items.

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

Grid is all about the container

[What We Do](#)[Our Approach](#)[Our Work](#)[Articles](#)[Speaking](#)[Labs](#)[Team](#)

Breaking Out With CSS Grid Layout

Written by [Tyler Sticka](#) on May 11, 2017

A while back I shared [a simple technique](#) for allowing certain elements to fill the full viewport width from within a fixed-width container:

```
.u-containProse {  
  max-width: 40em;  
  margin-left: auto;  
  margin-right: auto;  
}  
  
.u-release {  
  margin-left: calc(-50vw + 50%);  
  margin-right: calc(-50vw + 50%);  
}
```

<https://cloudfour.com/thinks/breaking-out-with-css-grid-layout/>

Just do this!

Magic occurs.

```
.Prose {  
  display: grid;  
  grid-template-columns:  
    [full-start] minmax(1em, 1fr)  
    [main-start] minmax(0, 40em) [main-end]  
    minmax(1em, 1fr) [full-end];  
}  
  
.Prose > * {  
  grid-column: main;  
}  
  
.Prose-splash {  
  grid-column: full;  
}
```

My markup

A div containing three direct child elements, one with a class of 'full'.

```
<div class="grid">  
  <div>Content</div>  
  <div class="full">Full width</div>  
  <div>Content</div>  
</div>
```


A grid with 3 column tracks

Using the line numbers to place our content and full width items.

```
.grid {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    minmax(1em, 1fr)  
    minmax(0, 40em)  
    minmax(1em, 1fr);  
}
```

```
.grid > * {  
  grid-column: 2 ;  
}
```

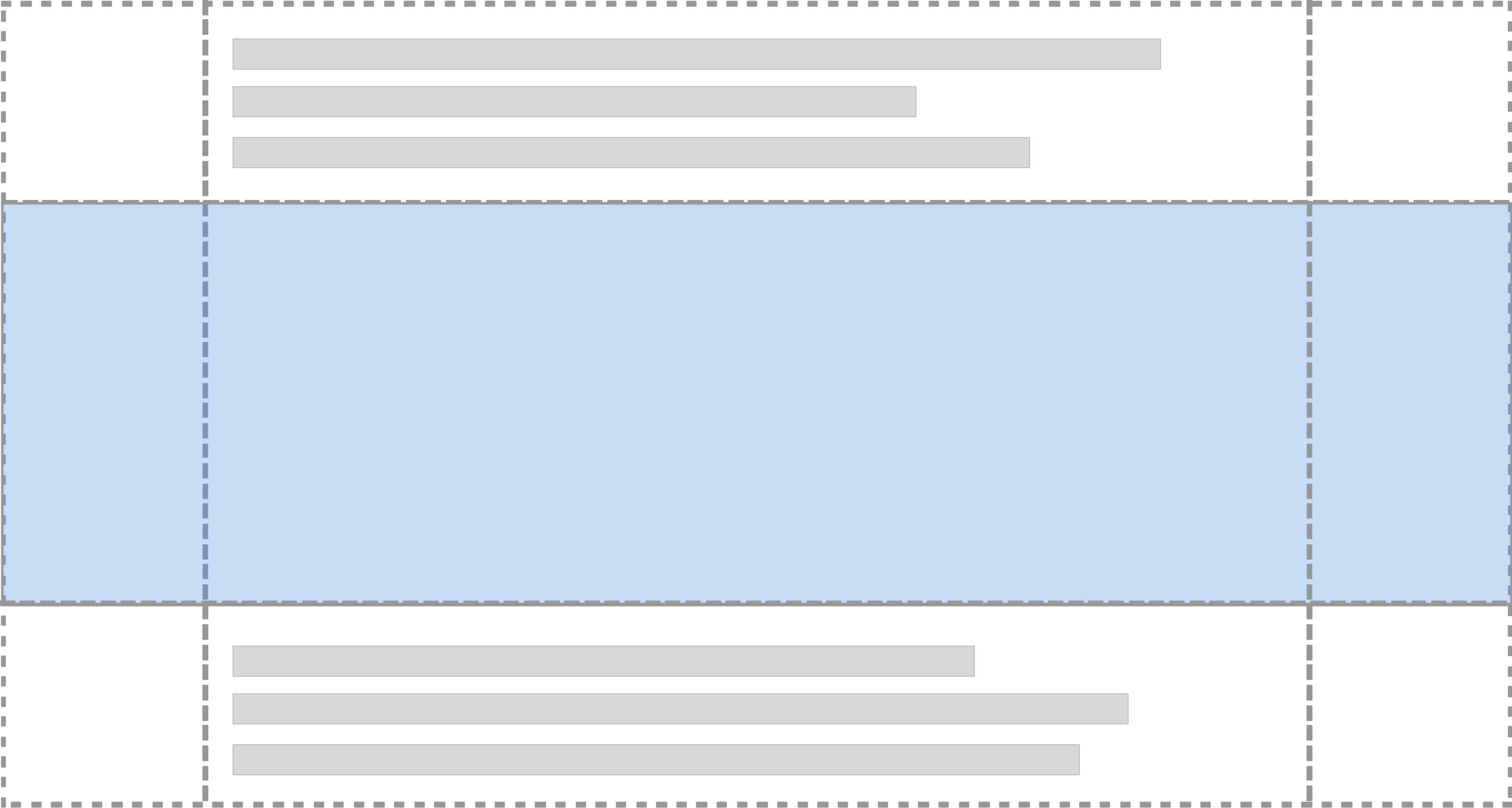
```
.grid > .full {  
  grid-column: 1 / 4 ;  
}
```

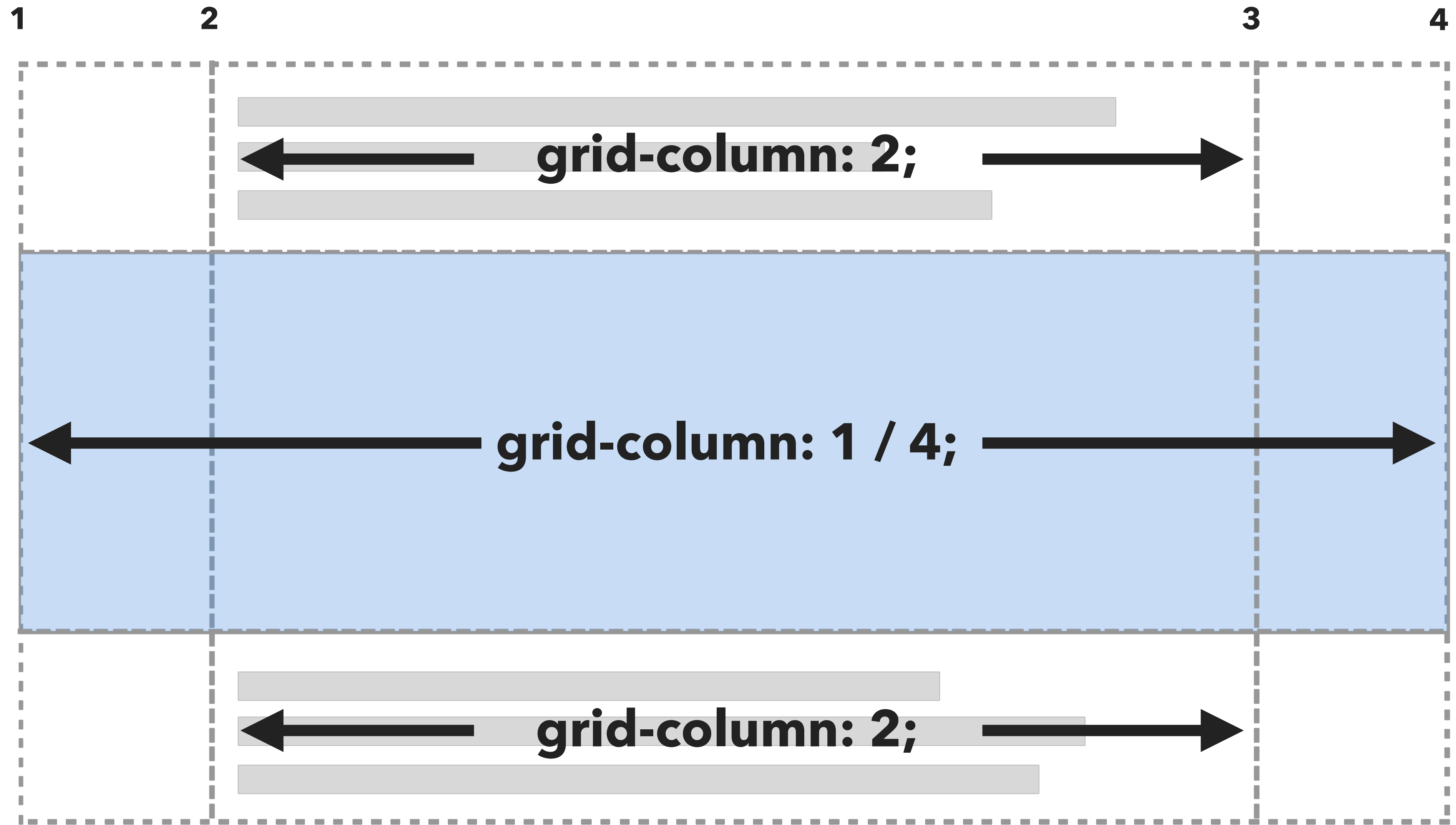
1

2

3

4





Naming lines on the grid

We can now position the items using their line *names*.

```
.grid {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    [full-start] minmax(1em, 1fr)  
    [main-start] minmax(0, 40em) [main-end]  
    minmax(1em, 1fr) [full-end];  
}  
  
.grid > * {  
  grid-column: main-start ;  
}  
  
.grid > .full {  
  grid-column: full-start / full-end;  
}
```

full-start

main-start

main-end

full-end

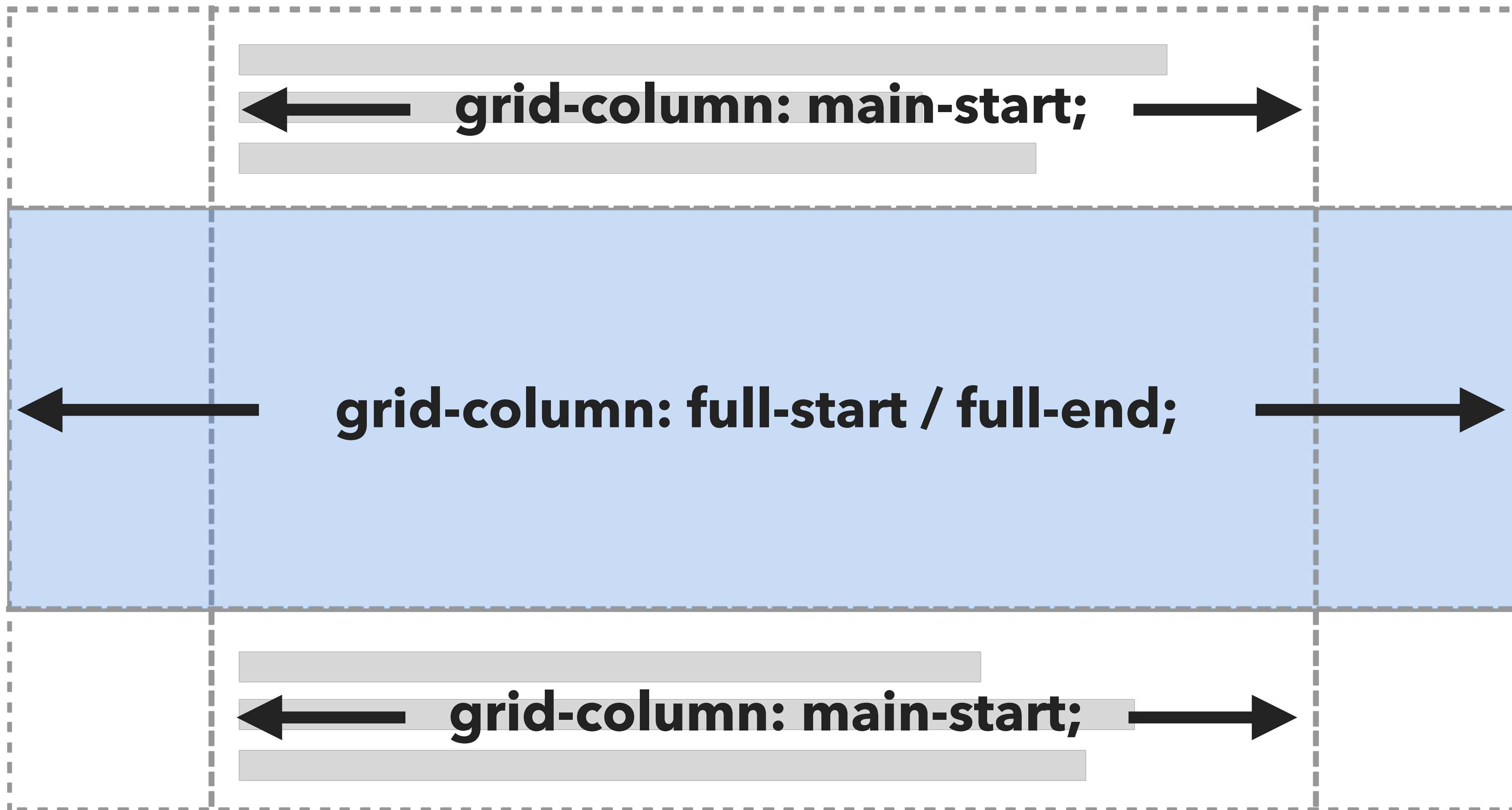


full-start

main-start

main-end

full-end

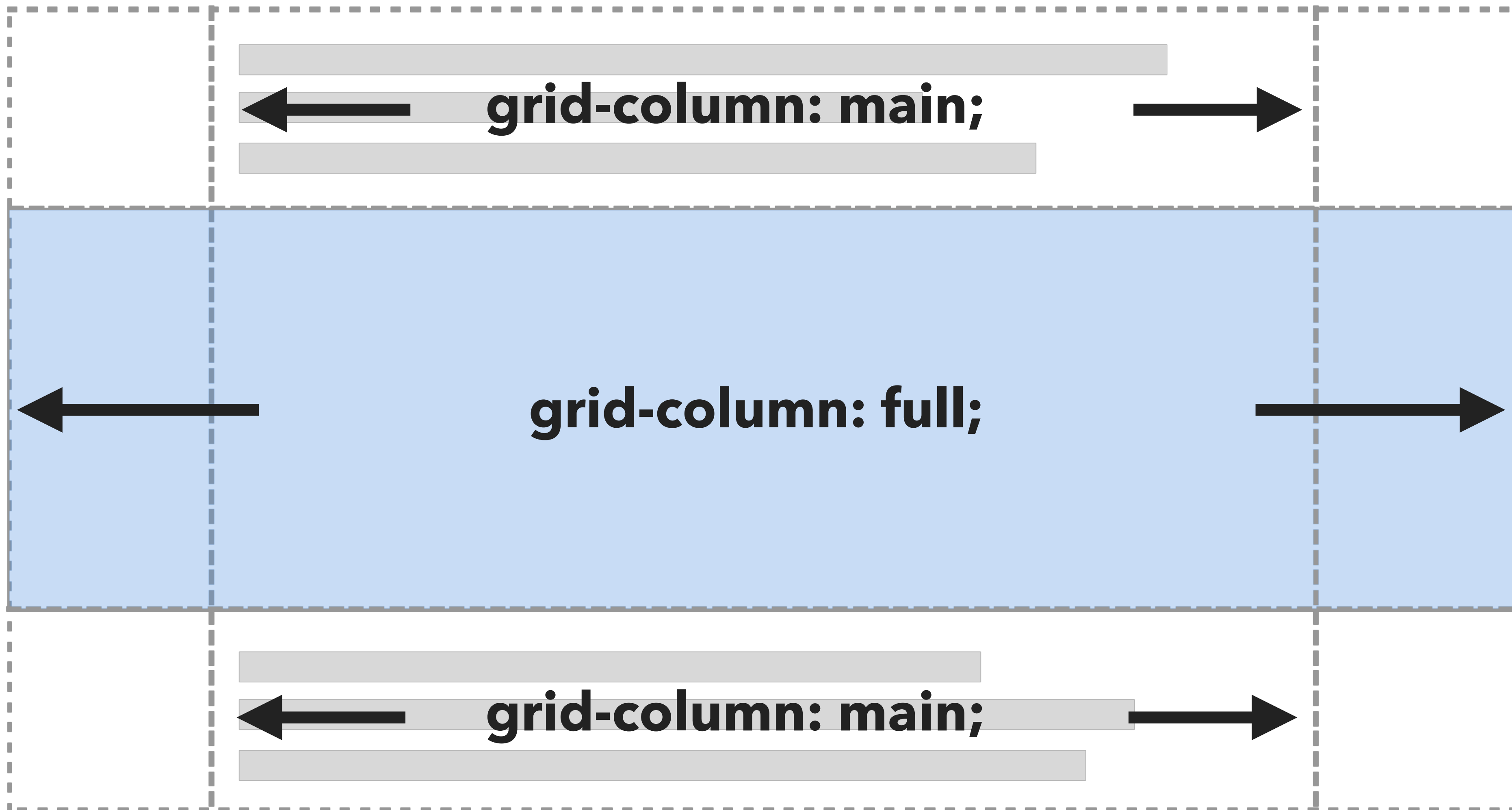


full-start

main-start

main-end

full-end



'main' and 'full'

These line names don't exist
anywhere in our grid definition.

```
.grid {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    [full-start] minmax(1em, 1fr)  
    [main-start] minmax(0, 40em) [main-end]  
    minmax(1em, 1fr) [full-end];  
}  
  
.grid > * {  
  grid-column: main;  
}  
  
.grid > .full {  
  grid-column: full;  
}
```


“Since a named grid area is referenced by the implicit named lines it produces, explicitly adding named lines of the same form (foo-start/foo-end) effectively creates a named grid area.”

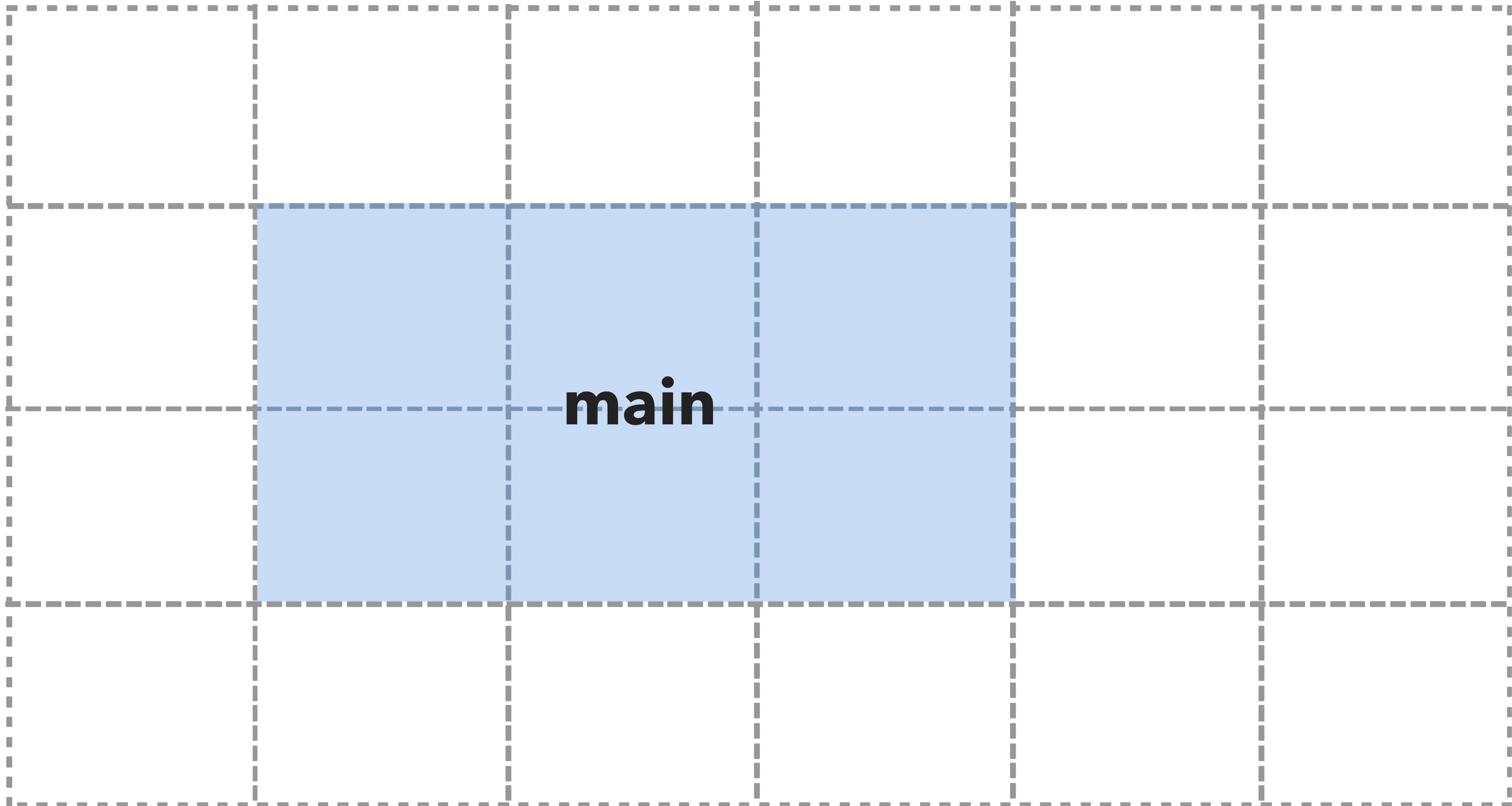
main-start

main-end

main-start

main-end

main



Implicit named areas

Created by having named lines using an ident with *-start and *-end.

<https://codepen.io/rachelandrew/pen/eWoKmd/>

```
.grid {  
  display: grid;  
  grid-gap: 20px;  
  grid-template-columns:  
    100px [main-start]  
    100px 100px 100px [main-end]  
    100px 100px;  
  grid-template-rows:  
    100px [main-start]  
    100px 100px [main-end] 100px;  
}  
  
.item {  
  grid-area: main;  
}
```


“Note: Named grid areas automatically generate implicit named lines of this form, so specifying `grid-row-start: foo` will choose the start edge of that named grid area (unless another line named `foo-start` was explicitly specified before it).”

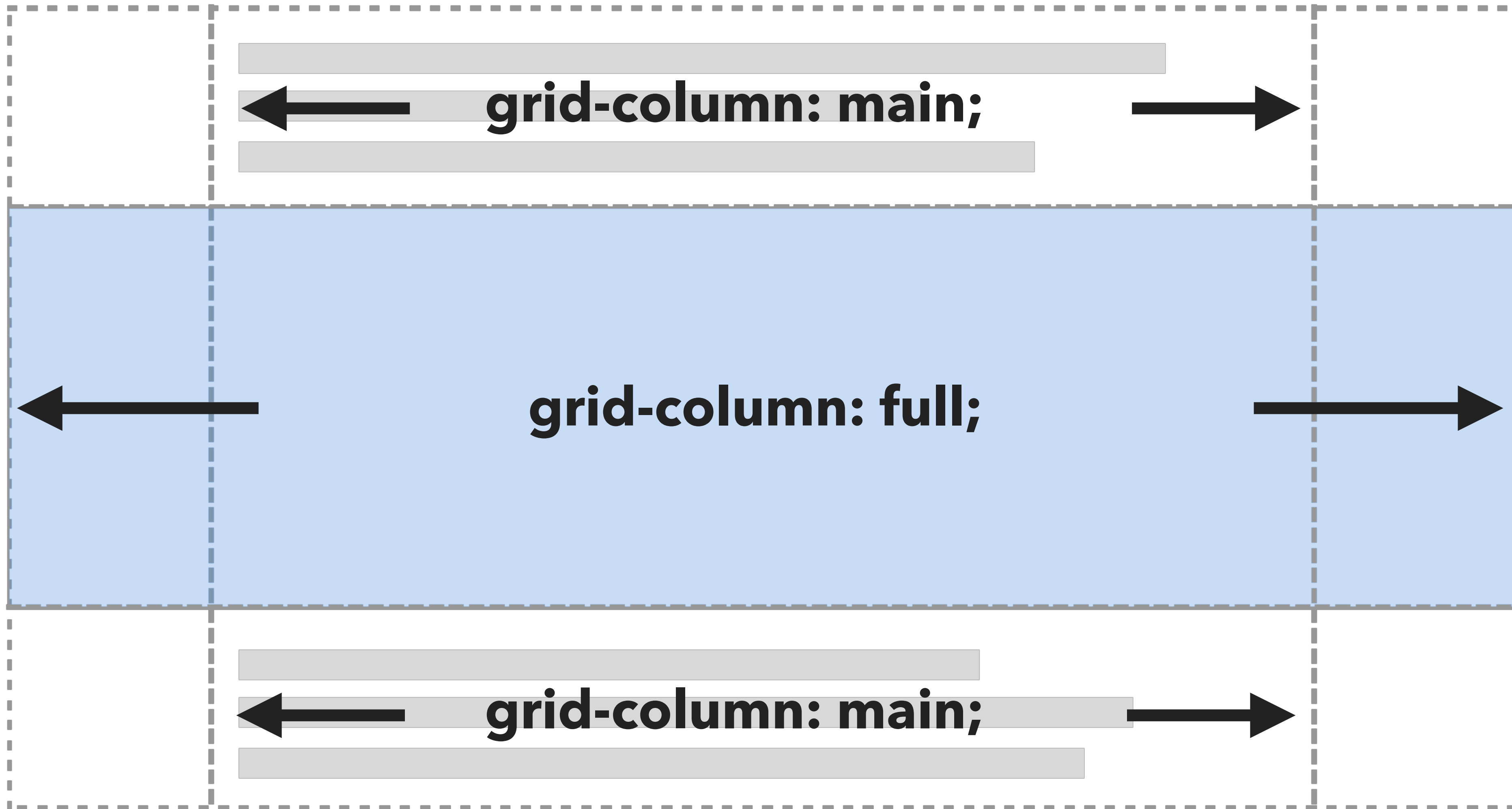
“[when using grid-row and grid-column shorthands] ... When the second value is omitted, if the first value is a <custom-ident>, the grid-row-end/grid-column-end longhand is also set to that <custom-ident>; otherwise, it is set to auto.”

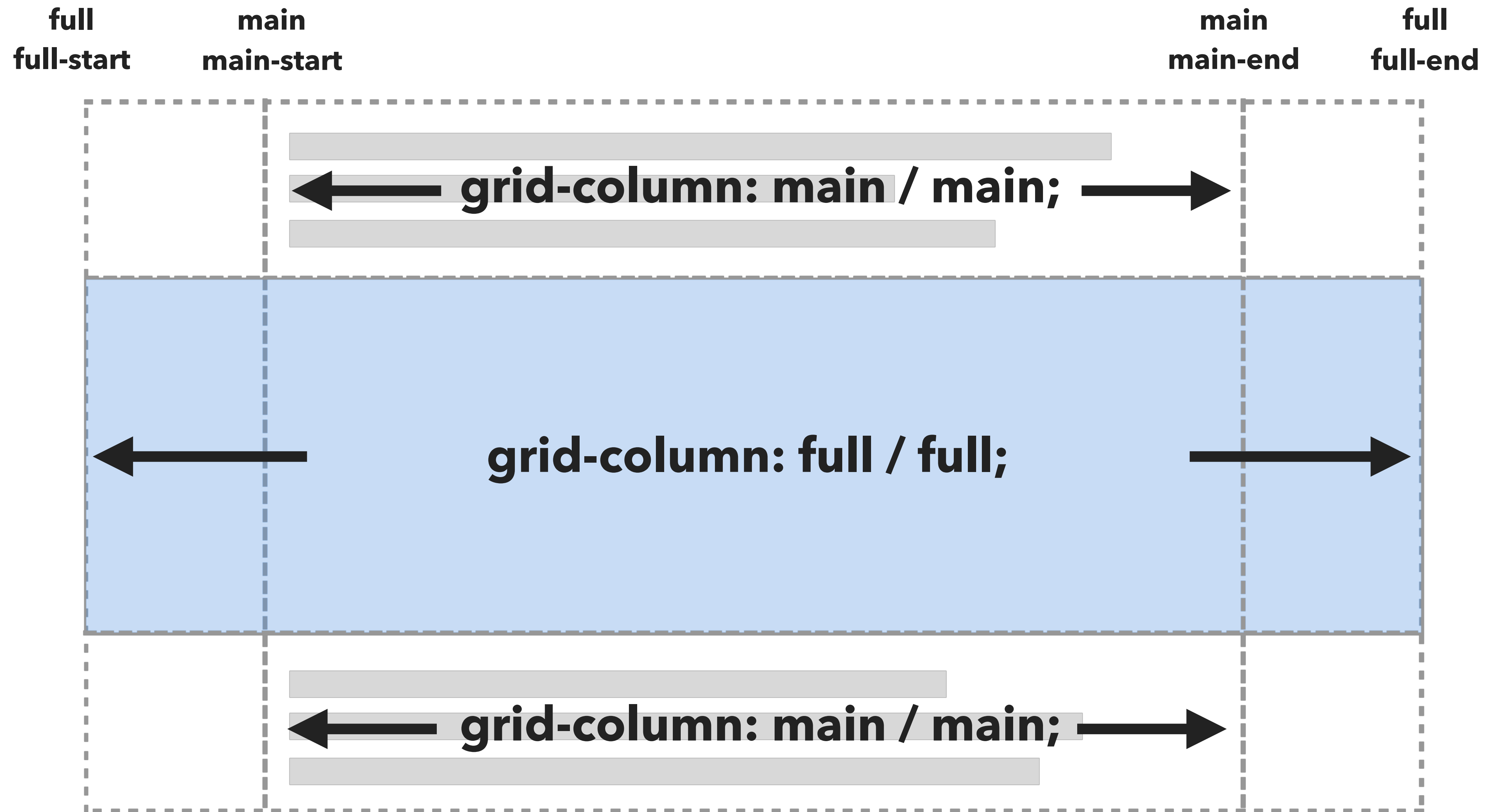
full-start

main-start

main-end

full-end





Content

Full width

Content

HTML

```
1
2 <div class="grid">
3   <div>Content</div>
4   <div class="full">Full width</div>
5   <div>Content</div>
```

CSS

```
1 .grid {
2   display: grid;
3   grid-gap: 20px;
4   grid-template-columns:
5     [full-start] minmax(1em, 1fr)
6     [main-start] minmax(0, 40em)
7     [main-end]
8     minmax(1em, 1fr) [full-end];
9 }
```

JS

Collections

Console

Assets

Comments

Delete

Keyboard

Last saved less than a minute ago

Share

Export

Embed

Inspector

Console

Debugger

Style Editor

Performance

Memory

Network

<!DOCTYPE html>

<html class="">

<head>

</head>

<body>

<div class="grid">

<div>Content</div>

<div class="full">Full width</div>

<div>Content</div>

</div>

</body>

</html>

</html>

</iframe>

<div id="assets-wrap" class="assets-wrap">

</div>

<section id="drawer" class="drawer">

</section>

<div id="editor-drag-cover" class="drag-cover">

</div>

Search HTML

Rules

Computed

Animations

Fonts

Lay

Filter Styles

element {

inline

.grid > .full {

inline:14

grid-column: full;

grid-column-start: full;

grid-column-end: full;

}

.grid > div {

inline:18

background-color: rgb(144, 122, 214);

font-weight: bold;

font-family: Baloo, sans-serif;

font-size: 3.157rem;

padding: 10px;

border-radius: 5px;

}

.grid > * {

inline:10

grid-column: main;

<https://codepen.io/rachelandrew/pen/xdedMy/>

Grid gives us powerful tools to control layout via the containing element.

CSS Grid Layout

**Using grid in
production**

This is a title

This is the introductory text for this section of the site.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.



New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.



Feature box

The feature has an image with a heading and body text overlaid.

```
<div class="box-feature">

  <h2 class="box-feature-title">Featured Item</h2>
  <div class="box-content">...</div>

</div>
```

Feature box

display: grid turns on grid layout

grid-gap defines gutters between grid items

grid-template-columns creates column tracks. In this case creating a grid with 6 columns.

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

The fr unit defines a fraction of the available space in the grid container

900×450

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

Feature box

The image starts at **grid column line 1** and ends at **-1**, which is the end line.

It starts at **grid row 1**, ending at **grid row 4**.

Using **box alignment** properties to stretch the image over that area.

```
.box-feature .box-image {  
  align-self: stretch;  
  justify-self: stretch;  
  grid-column: 1 / -1;  
  grid-row: 1 / 4;  
}
```

Grid lines respect writing mode.
Column line 1 is on the left and -1 on
the right in a LTR language.

Explicit vs. Implicit Grid

- ▶ The **Explicit Grid** is created when you define tracks with `grid-template-columns` and `grid-template-rows`
- ▶ If you place an item outside of that grid, or auto-placed content requires further row or column tracks these are added by grid as the **Implicit Grid**.

900×450

Featured

The first,
therefore

Feature box

Positioning the content inside the area that the image is stretched over.

```
.box-feature .box-feature-title {  
  grid-column: 3 / -1;  
  grid-row: 1;  
  background-color: rgba(0,0,0,0.7);  
  color: #fff;  
  align-self: start;  
  padding: 20px;  
}
```

```
.box-feature .box-content {  
  grid-column: 2 / -1;  
  grid-row: 2;  
  background-color: rgba(0,0,0,0.7);  
  color: #fff;  
  padding: 20px;  
}
```

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

900×450

Layering items on the grid

- ▶ You can position items into the same grid cells
- ▶ Items further down in the source appear on top of earlier items
- ▶ Control the stack using z-index

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

This is a title

This is the introductory text for this section of the site.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

500×350

The listing

The container for our boxes has 12 equal columns.

```
.listing {  
  display: grid;  
  grid-template-columns: repeat(12,1fr);  
  grid-gap: 20px;  
}
```


This is a title

This is the introductory text for this section of the site.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

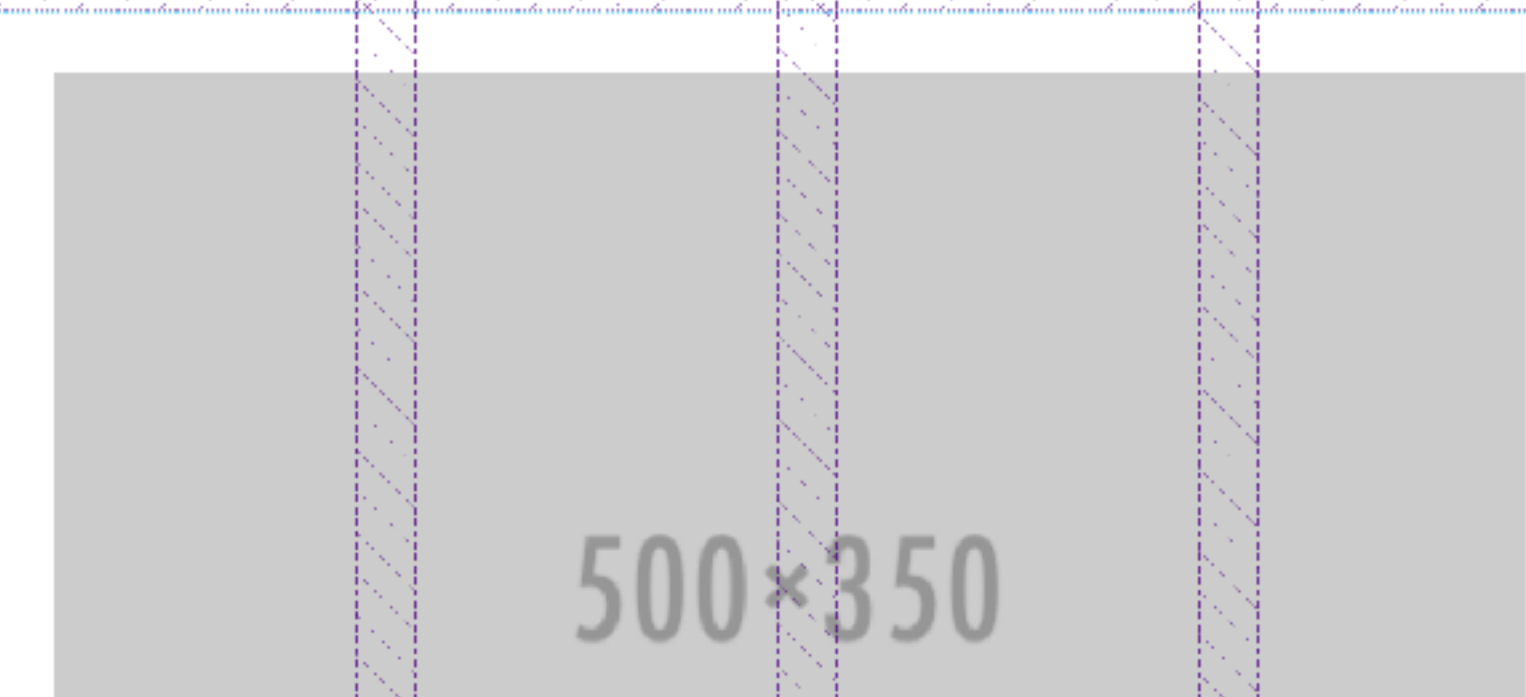
The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.



New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.



The listing

Positioning the title top left and the
feature top right

```
.box-title {  
  grid-column: 1 / 4;  
  grid-row: 1 / 2;  
}  
  
.box-feature {  
  grid-column: 4 / -1;  
  grid-row: 1 / 2;  
}
```

This is a title

This is the introductory text for this section of the site.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show



New Item

The newer items or next in priority after the feature are displayed in a larger

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Larger boxes

Newer items span 4 column tracks. If they also have a class of box-media they span 2 row tracks.

```
.box-newer {  
    grid-column: auto / span 4;  
}
```

```
.box-newer.box-media {  
    grid-row-end: span 2;  
}
```


Smaller boxes

The boxes for older items span 3 tracks.

```
.box-older {  
  grid-column: auto / span 3;  
}
```


This is a title

This is the introductory text for this section of the site.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

<http://codepen.io/rachelandrew/pen/Opaopw>

CSS Grid

**Going
responsive**

Going responsive

Inside media queries we can redefine where items sit on the grid.

```
.box-title {  
  grid-column: 1 / -1;  
  grid-row: 1;  
}
```

```
@media all and (min-width: 53.125em) {  
  .box-title {  
    grid-column: 1 / 6;  
    grid-row: 1 / 3;  
  }  
}
```

```
@media all and (min-width: 75em) {  
  .box-title {  
    grid-column: 1 / 4;  
    grid-row: 1 / 2;  
  }  
}
```


Going responsive

Or redefine how many columns they span.

```
.box-newer {  
  grid-column: 1 / -1;  
}
```

```
@media all and (min-width: 28.125em) {  
  .box-newer {  
    grid-column: auto / span 6;  
  }  
}
```

```
@media all and (min-width: 53.125em) {  
  .box-newer {  
    grid-column: auto / span 4;  
  }  
}
```


New Item

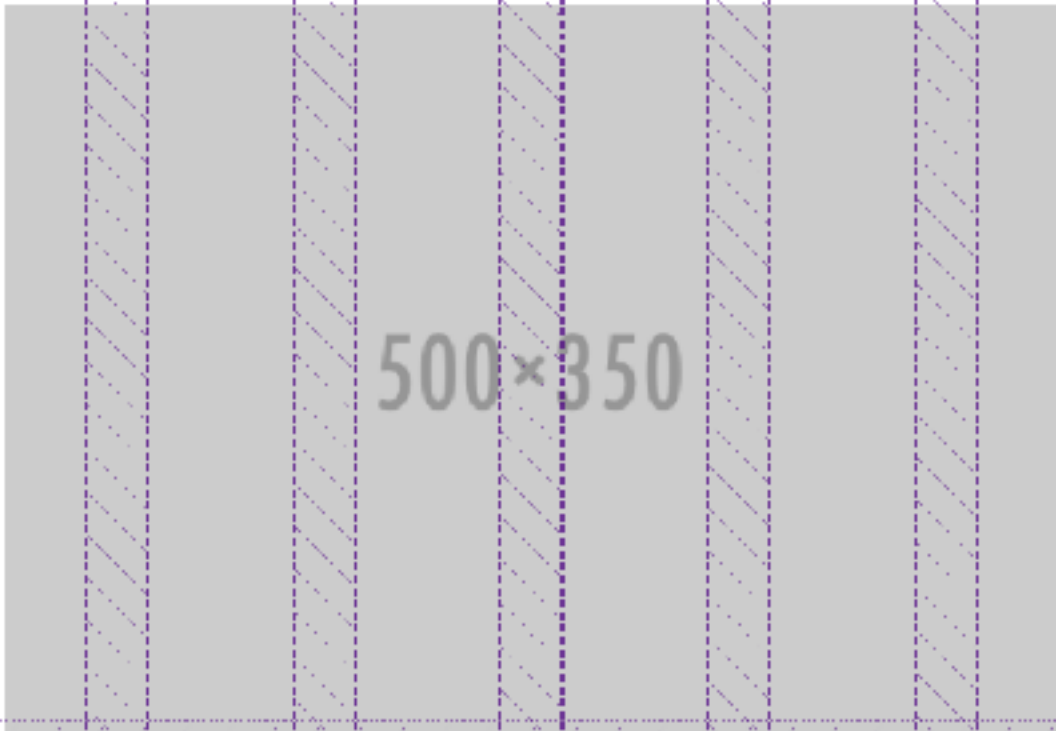
The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.



New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

Older Item

The newer items or next in

This is a title

This is the introductory text for this section of the site.

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item is the big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.



New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

Older Item

The newer items or next in priority after the feature are displayed in a smaller box. We can show additional content for these items.

<http://codepen.io/rachelandrew/pen/gmQdgz>



CSS Grid Layout

**What about
old browsers?**

What about old browsers?

If using display: grid on a container, child items:

- ▶ Using float, lose their float behaviour
- ▶ The vertical-align property has no effect
- ▶ Flex items become grid items
- ▶ Items set to display: block or inline-block become grid items
- ▶ Items set to display: table-cell stop creating anonymous boxes

You do not need to build “two layouts”

This is a title

This is the introductory text for this section of the site.

900×450

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

Adding a flex fallback

Browsers that support `display: flex` and not `grid` will turn the children into flex, not grid, items.

The flex properties applied to those items will be ignored by grid layout.

```
.listing {  
  display: flex;  
  flex-wrap: wrap;  
  margin: 0 20px;  
  display: grid;  
  grid-template-columns: repeat(12,1fr);  
  grid-gap: 20px;  
}  
  
.listing > * {  
  flex: 1 1 30%;  
  margin: 0 20px 20px 20px;  
}
```

Feature Queries are your new best friend

CSS Feature Queries - CR

Global91.76%

CSS Feature Queries allow authors to condition rules based on whether particular property declarations are supported in CSS using the @supports at rule.

Current alignedUsage relativeDate relativeShow all

IE	Edge*	Firefox	Chrome	Safari	Opera	iOS Safari*	Opera Mini*	Android Browser*	Chrome for Android
			49						
			59					4.4	
	14	54	60			10.2		4.4.4	
11	15	55	61	10.1	47	10.3	all	56	59
	16	56	62	11	48	11			
		57	63	TP	49				
		58	64						

- Notes
- Known issues (2)
- Resources (7)
- Feedback

See also the [CSS.supports\(\) DOM API](#)

Using feature queries

Add a margin for flex layout, remove it if we are using grid layout.

```
.listing > * {  
  flex: 1 1 30%;  
  margin: 0 20px 20px 20px;  
}
```

```
@supports(display: grid) {  
  .listing > * {  
    margin: 0;  
  }  
}
```

This is a title

This is the introductory text for this section of the site.

900×450

Featured Item

The first, therefore usually the latest but could be flagged as a feature, item is the feature. This item gets a big spot up next to the title of this section.

It might have an image too. It's special like that.

New Item

The newer items or next in priority after the feature are displayed in a larger box. We can show additional content for these items.

New Item

New Item

Flex layout

Give the feature box a larger flex-basis percentage.

```
.listing .box-feature {  
    flex: 1 1 60%;  
}
```


Float and Clear

The float and clear properties have no effect on a grid item.

<https://codepen.io/rachelandrew/pen/YZeqZv>

```
.grid > div {  
  float: left;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```

display: inline-block

The properties associated with something being inline-block cease to apply.

<https://codepen.io/rachelandrew/pen/vxdGjQ>

```
.grid > div {  
  display: inline-block;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```


display: table

Anonymous boxes will not be generated and the item will become a grid item.

<https://codepen.io/rachelandrew/pen/bqLpQN>

```
.grid > div {  
  display: table-cell;  
  vertical-align: top;  
}
```

```
.grid {  
  border-spacing: 10px;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```


The vertical-align property

Can be used as a fallback for box alignment and ceases to apply on grid items.

<https://codepen.io/rachelandrew/pen/vxdGaQ>

```
.grid > div {  
  display: inline-block;  
  vertical-align: top;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
  width: 500px;  
}
```

Multiple-column layout

Multiple-column layout properties cease to apply in grid layout.

<https://codepen.io/rachelandrew/pen/JWpXxv>

```
.grid {  
  column-count: 3;  
  width: 500px;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
}
```

Flex layout

Grid will override flex layout and shares box alignment properties.

<https://codepen.io/rachelandrew/pen/YZeqMB>

```
.grid {  
  display: flex;  
  align-items: center;  
  width: 500px;  
  height: 200px;  
  border: 1px dotted #694486;  
}
```

```
.grid > div {  
  flex: 1;  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, auto);  
}
```


Overrides inside @supports are
mostly widths & margins

Override widths in feature queries

Watch out for widths in your fallback layouts.

<https://codepen.io/rachelandrew/pen/JWpXNr>

```
* { box-sizing: border-box; }
```

```
.grid > div {  
  float: left;  
  width: 33.333%;  
}
```

```
@supports (display: grid) {  
  .grid > div {  
    width: auto;  
  }  
}
```

```
.grid {  
  display: grid;  
  grid-gap: 10px;  
  grid-template-columns: repeat(3, 1fr);  
  width: 500px;  
}
```

Grid “fallbacks” and overrides

[Defined in the CSS Grid Specification](#) are the ways in which grid interacts with other layout methods. These definitions mean that as soon as an item becomes a grid item, other behaviour that you may have used for older browsers is overwritten. This means that you do not have to completely fork your code and build two versions. Where you do need to overwrite CSS used for older browsers, you can do so inside a CSS Feature Query. This enables safe encapsulation of any CSS you only want a grid supporting browser to apply.

Here is a quick reference to the defined overrides with simple examples. See the resources at the end of this cheatsheet for more advice and examples.

Floated items

Float and clear have no effect on a grid item

If you float an item, in your CSS for non-grid browsers, when that item becomes a grid item float ceases to have any effect. The clear property applied to an item that becomes a grid item also no longer takes effect.

Example

```
<div class="grid">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</div>
```

```
.grid > div {
  float: left;
}
```


Edge Grid implementation

- ▶ Updated implementation in Windows Insider build - will ship in October
- ▶ Currently tied to the IE10 implementation
- ▶ Prefixed with -ms
- ▶ No auto-placement or grid-template-areas layout
- ▶ For simple line-based positioning **it works**
- ▶ More at <https://rachelandrew.co.uk/archives/2017/04/04/edge-starts-work-on-their-grid-implementation-update/>

Autoprefixer can add -ms-grid
prefixes. This is rarely helpful.



March 2017



March 2017



March 2017



March 2017



March 2017



October 2017

Let browser vendors know which
features you want.

How can we improve the Microsoft Edge developer experience?

[← Microsoft Edge Developer](#)New and returning users may [sign in](#)

6,502

votes

Vote

Update CSS Grid

Update the IE implementation of Grid Layout to match the current spec. The current implementation from the IE 10 days is outdated.



[mnsth](#) shared this idea · October 02, 2014 · [Flag idea as inappropriate...](#)

**WORKING ON IT**· **Microsoft Edge Developer UserVoice** (Admin, Windows Developer) responded · April 03, 2017

Since proposing Grid Layout to the CSS Working Group and shipping an experimental implementation in IE10, we have been heavily invested in advancing it through the standard process.

Today we're pleased to announce that work has begun on updating our implementation to the latest Candidate Recommendation spec. We're excited about the level of developer and implementor interest in Grid Layout, and look forward to see how it will be used on the Web.

To track our progress with updating CSS Grid go to <https://developer.microsoft.com/en-us/microsoft-edge/platform/status/gridupdate/?q=grid>

[Show previous admin responses \(1\)](#)

165 comments

[Tweet](#) [Like](#) 48

Add a comment...

Your email address

or sign in with [f](#) [s](#)Microsoft Edge Developer: **CSS**[Post a new idea...](#)[All ideas](#)[My feedback](#)[Accessibility](#) 10**CSS** 97[Document Object Model \(DOM\)](#) 22[Extensions](#) 52[F12 Developer Tools](#) 174[Graphics](#) 24[HTML](#) 43[JavaScript](#) 77[Media](#) 29[Microsoft Edge Dev site](#) 17[Miscellaneous](#) 67[Networking](#) 33[Performance](#) 21[Security](#) 24

Search

**Give feedback**[Command Prompt / Console / Bash on Ubuntu on Windows](#) 413[Universal Windows Platform](#) 1,801[Windows App Studio](#) 795<https://wpdev.uservoice.com/forums/257854-microsoft-edge-developer/>

[Home](#) \ [Platform](#) \ [Data](#) \ CSS Usage

CSS usage on the web platform

Where is the data from? ▾

352 properties

Filter properties

property-name

font-size	98.3%
color	98.2%
width	97.2%
margin-top	97.1%
margin-left	97%
background-color	96.9%
margin-bottom	96.9%
margin-right	96.7%
padding-left	95.8%

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam posuere quam non enim commodo, ac pellentesque tortor lacinia. Aenean iaculis interdum tempor. Nullam porttitor lectus eget mollis egestas. Cras posuere velit id vehicula tempor. Mauris id ex id nisi sagittis ornare vitae a erat. Aliquam nisi risus, gravida sit amet augue id, feugiat tempor dui. Vestibulum tincidunt iaculis tellus, eget finibus odio.

Quisque ac dapibus massa. Donec sagittis turpis lobortis ac. Praesent velit, id placerat enim imperdiet sit consequat ligula tellus, quis lacinia Praesent sit amet faucibus purus. augue.

Aliquam tempor interdum volutpat. Nunc nibh elit, sagittis nec libero a, sapien. Aliquam eu risus eu dui

Curabitur ut ex luctus, egestas dui sit amet, malesuada lectus. Cras a nibh in velit mattis feugiat at at eros. Etiam et leo arcu. Maecenas egestas enim purus, at ornare velit laoreet a. Etiam egestas pharetra



placerat nisi velit, quis ornare malesuada amet. Vestibulum mauris porta in. Integer vel sagittis

Suspendisse potenti. pretium fermentum varius elementum.

diam, non accumsan sapien posuere ornare. Ut sed velit porttitor dolor tincidunt tincidunt. Pellentesque sit amet pulvinar diam. Cras et purus vulputate, consequat dui vel, eu sus risus. Quisque purus diam, convallis tempor. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Exclusions

Defines the wrap-flow property, which enables wrapping content round all sides of an element.

```
.exclusion {  
    -ms-wrap-flow: both;  
    wrap-flow: both;  
}
```




Features: 862



CSS Exclusions

CSS

Define areas inline content should avoid, and how inline content should wrap around them.

Chromium status

Proposed

Launch bug

[#234749](#)

Owner(s)

betravis@adobe.com

Consensus

Web developers: No signals



Working draft or equivalent

Comments

Rossen Atanassov from Microsoft is co-editor on the spec. IE shipped with a prefix. Apple removed the code saying it "needs some evolution before the actual implementation happens" trac.webkit.org/changeset/166618

CSS Selector Pseudo class :dir()

CSS

The :dir CSS pseudo-class matches elements based on the directionality of the text contained in it. Pseudo-cl...

CSS Selectors Level 4: :focus-within pseudo-class

CSS

The :focus-within pseudo-class applies to elements for which the :focus pseudo class applies. An element als...

CSS backdrop-filter property

CSS

Add backdrop-filter css property, which allows filters to be applied to the content behind an element.

<https://www.chromestatus.com/features/6296903092273152>

You can get involved in the future of
CSS.



This repository

Search

Pull requests

Issues

Gist



w3c / csswg-drafts

Unwatch

156

★ Unstar

437

🍴 Fork

107

<> Code

🔔 Issues 557

🔗 Pull requests 11

📁 Projects 3

📶 Pulse

📊 Graphs

Filters

🔍 is:issue is:open css-grid

Labels

Milestones

New issue

✕ Clear current search query, filters, and sorts

☐ 🔔 40 Open ✓ 78 Closed Author Labels Projects Milestones Assignee Sort☐ 🔔 [css-align][css-flexbox][css-grid] partial implementations of space-evenly in grid but not flexbox Agenda+ css-align-3 css-flexbox-1 css-grid-1
#1167 opened a day ago by frivoal 3☐ 🔔 [css-grid] syntax errors in examples css-grid-1
#1166 opened a day ago by phax 1☐ 🔔 [css-grid] Grid Area Section 9. not logically consistant with Section 3.3 definition of Grid Area
#1159 opened 4 days ago by sunright☐ 🔔 [css-grid] Stretching tracks fails to feed back into sizing algorithm css-grid-1
#1150 opened 6 days ago by fantasai 1☐ 🔔 [css-grid] Automatic Minimum Size for grid items should not min against content size css-grid-1
#1149 opened 6 days ago by fantasai 1☐ 🔔 Implementations don't match the Flex Container Intrinsic Main Sizes algo, resulting in gratuitous differences from equivalent Grid layouts css-flexbox-1
#1147 opened 6 days ago by tabatkins 2☐ 🔔 [css-grid] grammar of grid-row-start and friends is ugly and harms value space for line names Agenda+ css-grid-1
#1137 opened 8 days ago by therealglazou 9<https://github.com/w3c/csswg-drafts/issues>



This repository

Search

Pull requests

Issues

Gist



w3c / csswg-drafts

Unwatch

156

★ Unstar

437

Fork

107

<> Code

Issues 557

Pull requests 11

Projects 3

Pulse

Graphs

[css-grid] Decorative grid-cell pseudo-elements #499

Edit

New Issue

Open

tabatkins opened this issue on 19 Sep 2016 · 6 comments



tabatkins commented on 19 Sep 2016

Member



@jensimmons proposes adding a grid-cell pseudo so you can add (responsive) decorative elements to grids without having to add empty elements to your page.

Syntax like:

```
#grid::grid-area(1 / 2 / 3 / 4) {  
  background-color: red; /* etc */  
  /* grid-positioning properties are blacklisted */  
}
```

Also some way of controlling whether or not that takes up auto-flow space. Might want to reserve space or not. Can we base this on 'content' or something? Or do we need a new property?



3



tabatkins added the `css-grid-2` label on 19 Sep 2016



jensimmons commented on 19 Sep 2016



The slides from the presentation I just gave are at: <https://speakerdeck.com/jensimmons/to-csswg-sept-2016>

And actually, the syntax above is not what I proposed. This is what I proposed.

Assignees



No one—assign yourself

Labels



css-grid-2

Projects



None yet

Milestone



No milestone

Notifications

Unsubscribe

You're receiving notifications because you're subscribed to this repository.

7 participants



<https://github.com/w3c/csswg-drafts/issues/499>

Get involved with CSS

- ▶ Comment on or raise new issues against CSS specifications
- ▶ Raise bugs against browsers
- ▶ Vote on features where browsers have a platform to do so
- ▶ Write about new features - it demonstrates we want them
- ▶ **Be nice** while doing it. Browser engineers and spec editors work within constraints just as you do in your projects.

CSS Grid

is here!

Find out more

I made you some resources

Visit Grid by Example for worked examples, and a free video tutorial:

<http://gridbyexample.com>

I created a huge set of guides for MDN:

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout

Over 5 years of grid thoughts on my site at:

<https://rachelandrew.co.uk/archives/tag/cssgrid>

GridBugs! I'm collecting and trying to get fixed interop issues:

<https://github.com/rachelandrew/gridbugs>

Rachel Andrew

THE NEW CSS LAYOUT

FOREWORD BY Jen Simmons

Out now!

The New CSS Layout

@rachelandrew

<https://rachelandrew.co.uk/speaking/event/gdg-nantes>

THANK YOU!