

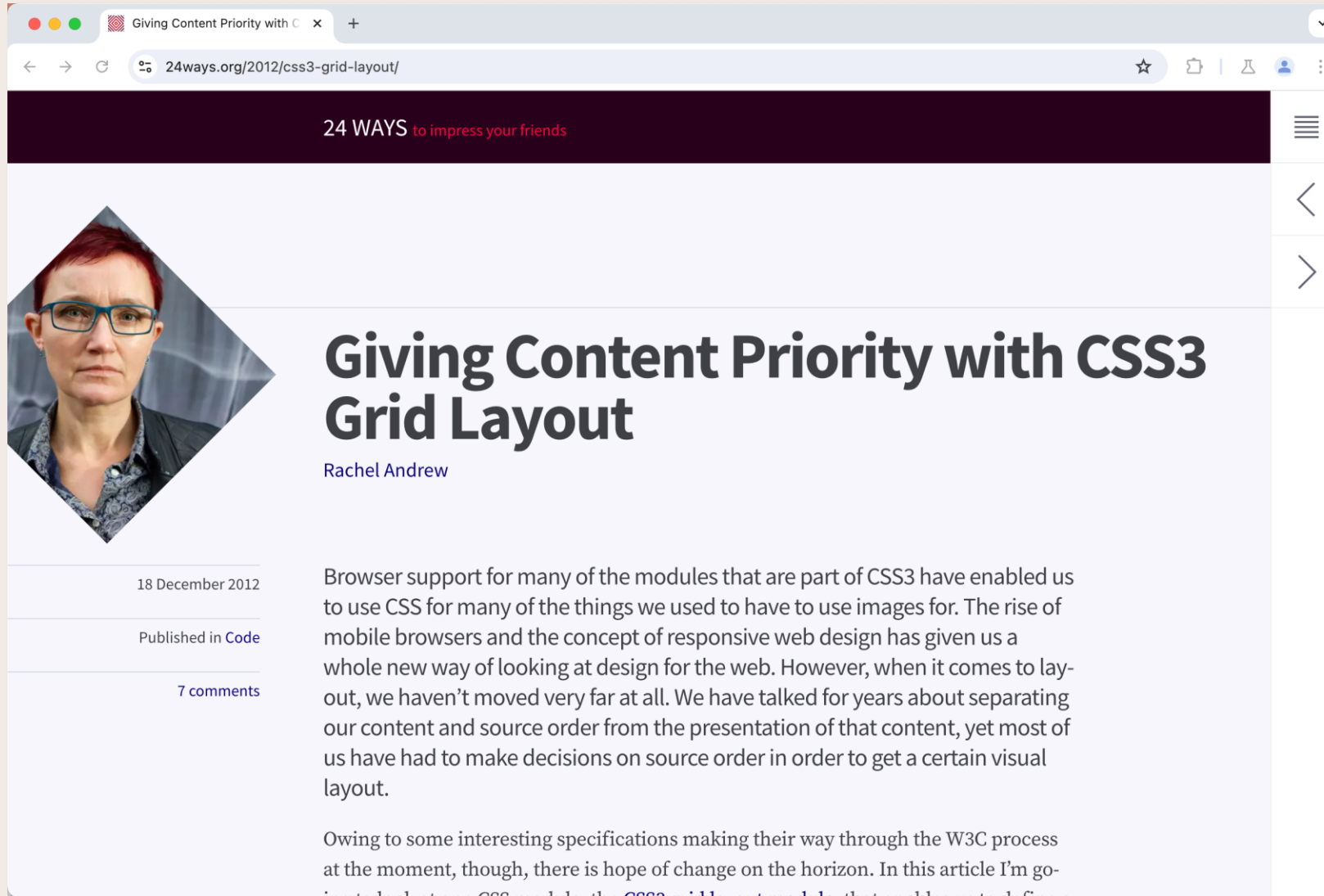
Layout and
reading order

Rachel Andrew

Content lead for Chrome
Developer Relations

CSS WG member

rachelandrew.co.uk



24ways.org/2012/css3-grid-layout



Baseline Widely available



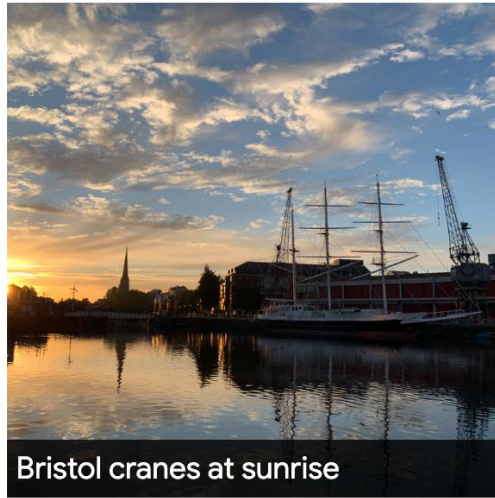
CSS Grid, Baseline Widely available since January 2023

Without subgrid we create the potential for accessibility problems. Authors may remove semantic markup in order to use grid layout.

Slide from 2015 CSS Day grid talk

“You never know how tall anything is on
the web”

Me ... for most of my web development career.



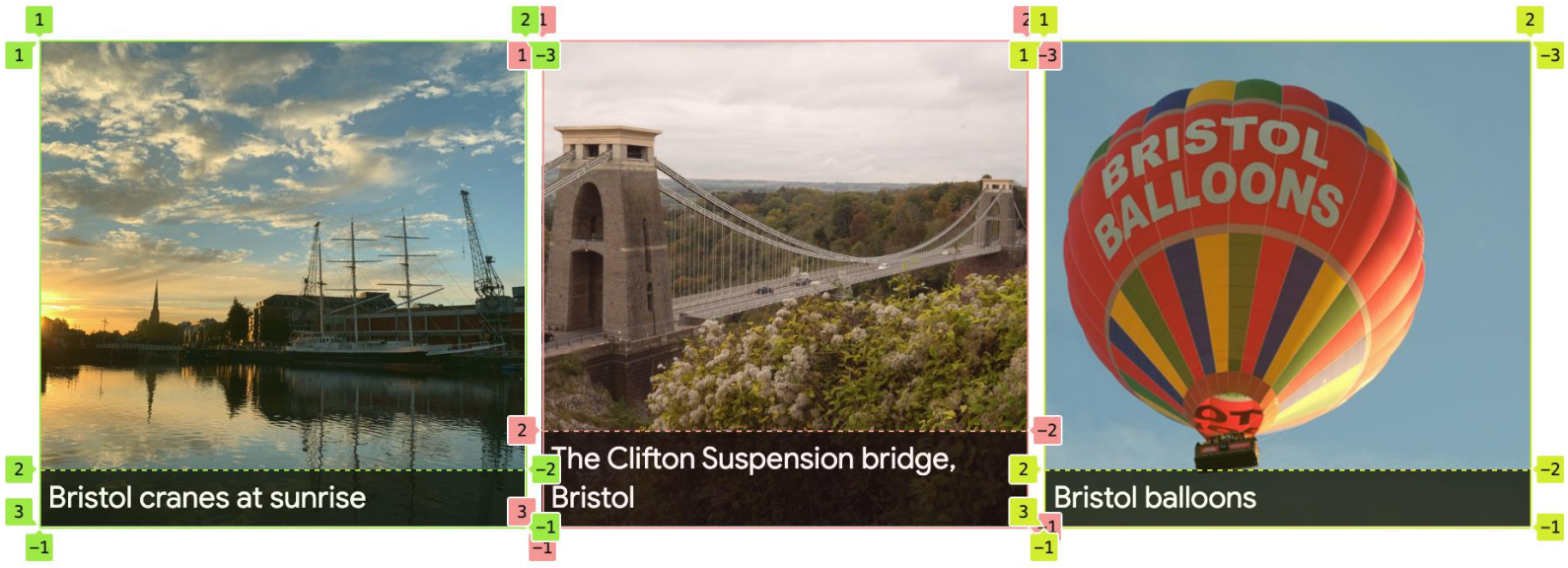
Bristol cranes at sunrise



The Clifton Suspension bridge,
Bristol



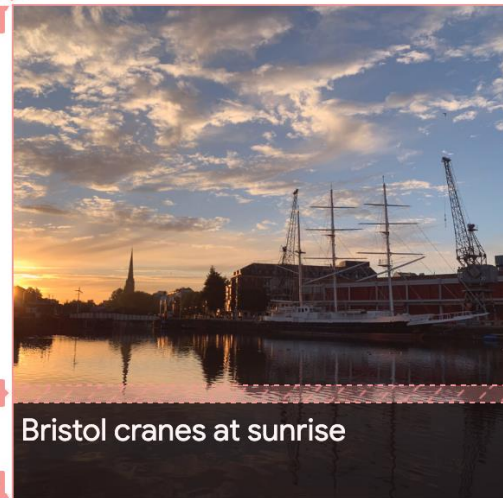
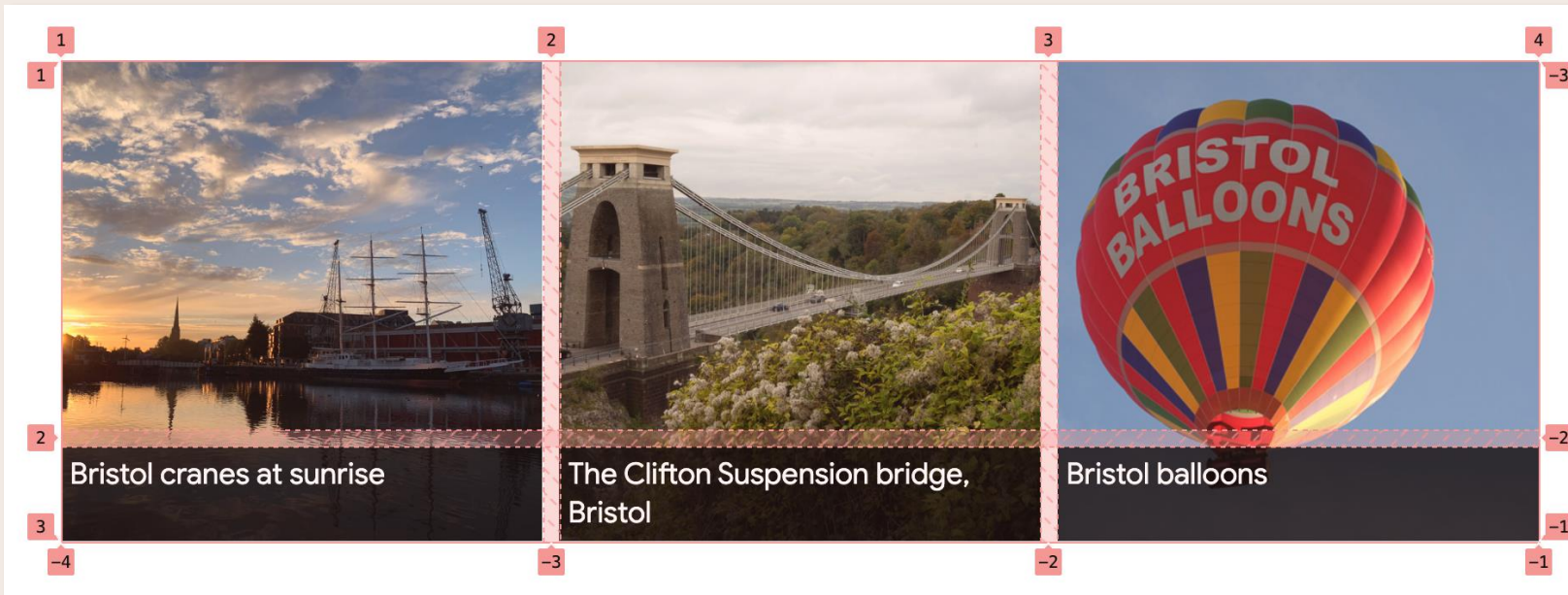
Bristol balloons



Bristol cranes at sunrise

The Clifton Suspension bridge,
Bristol

Bristol balloons





Baseline 2023

NEWLY AVAILABLE

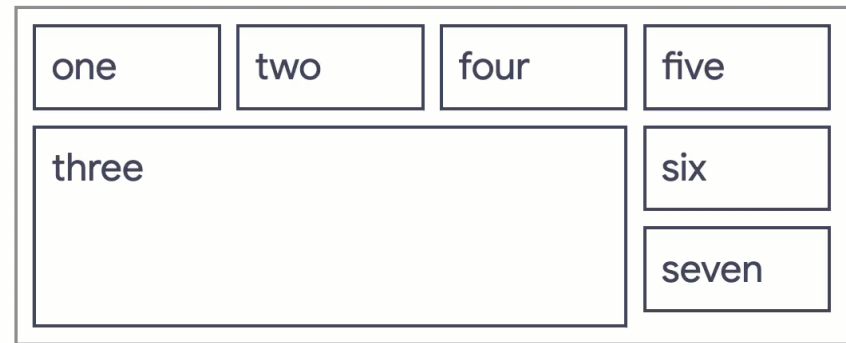


web.dev/articles/css-subgrid

reading-flow

A missing puzzle piece that can prevent problems and unlock capabilities.

```
.grid a:nth-child(3) {  
  grid-column: span 3;  
  grid-row: span 3;  
}  
  
.grid {  
  display: grid;  
  grid-auto-flow: dense;  
  grid-template-columns: repeat(4,1fr);  
}
```





Here's a cool thing

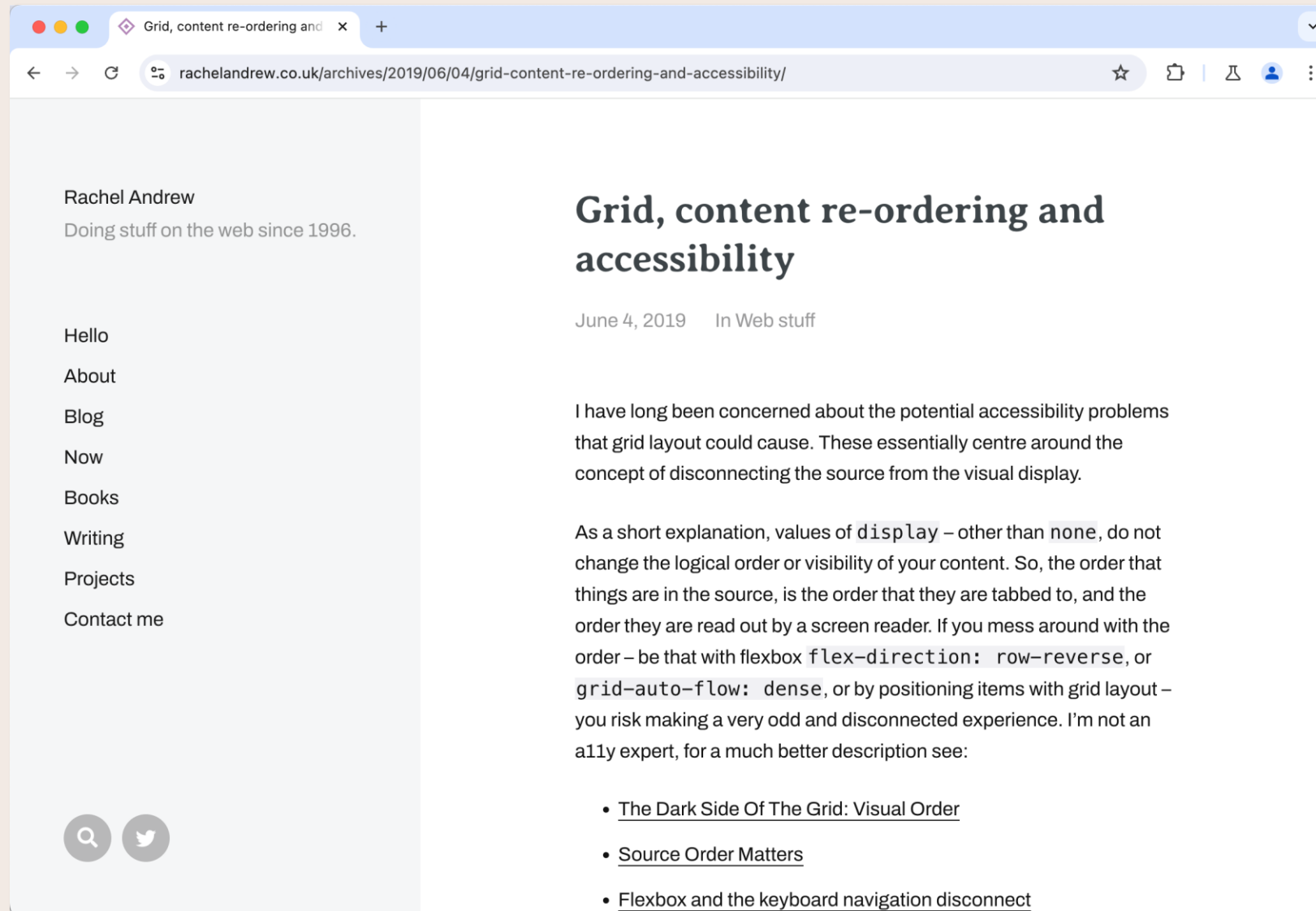
Please don't use it.

CSS Grid Level 1

Correct source order is important for speech, for sequential navigation (such as keyboard navigation), and non-CSS UAs such as search engines, tactile browsers, etc. Grid placement *only* affects the visual presentation! This allows authors to optimize the document source for non-CSS/non-visual interaction modes, and use grid placement techniques to further manipulate the visual presentation so as to leave that source order intact.

Separating content from presentation

Something we've tried to do since the early days of CSS layout.



<https://rachelandrew.co.uk/archives/2019/06/04/grid-content-re-ordering-and-accessibility/>

The screenshot shows a web browser window with the address bar displaying `github.com/w3c/csswg-drafts/issues/5675`. The page header includes the repository name `w3c/csswg-drafts` (Public), along with buttons for Notifications, Fork (636), and Star (4.3k). A navigation bar below the header contains links for Code, Issues (3.2k), Pull requests (79), Actions, Projects (17), Security, and Insights.

The main heading of the issue is `[css-grid] Masonry layout in CSS Grid 3 has potential to cause accessibility problems with reading order. #5675`, with a `New issue` button to its right. Below the heading, a green `Open` button and text indicate that `nattarnoff` opened this issue on Oct 28, 2020, with 12 comments.

The comment section shows a comment by `nattarnoff` from Oct 28, 2020. The comment text is as follows:

Assuming all authors code their masonry grid in a valid reading order, screen reader user who are non-visual will have no problem with it. But once we introduce visual screen reader users, low vision with no assistance, magnification software users, and users with cognitive problems, the layout begins to cause problems in focus order and reading order.

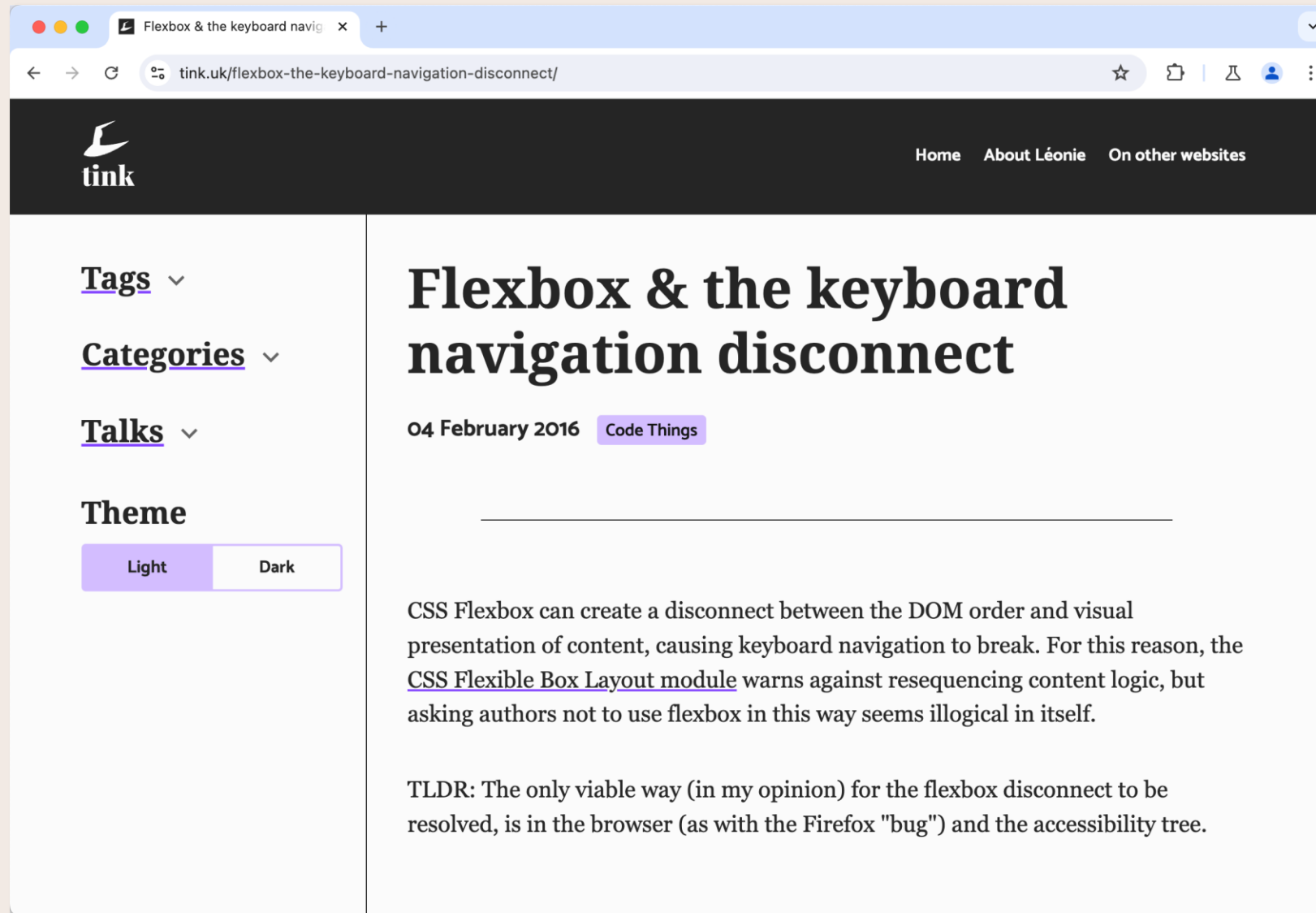
If we have

```
<div>
<item>The quick</item>
<item>brown fox</item>
<item>jumped</item>
<item>over</item>
<item>the lazy</item>
<item>dog</item>
</div>
```

The screen reader user will hear "the quick brown fox jumped over the lazy dog." When we add the CSS and visual layout, the reading order could be realistically be:

On the right side of the comment, there are sections for Assignees (No one assigned), Labels (`css-grid-3` and `topic: masonry`), Projects (None yet), Milestone (No milestone), Development (No branches or pull requests), and 9 participants (represented by avatars).

<https://github.com/w3c/csswg-drafts/issues/5675>



<https://tink.uk/flexbox-the-keyboard-navigation-disconnect/>

reading-flow

CSS Display Level 4

Can we just opt into the order that flex or grid items are laid out?

Reading or visual order is subjective



`flex-direction: row-reverse`

In most cases DOM order is what you want

No universal switch to let developers forget about source order.

Scoped to particular layout methods

We're not trying to do this on block layout.

reading-flow: normal

Follow the order of items in the DOM. The initial value.

```
.flex {  
  display: flex;  
  flex-direction: row-reverse;  
}
```



normal
flex-flow
flex-visual
grid-rows
grid-columns
grid-order

Values for reading-flow

reading-flow: flex-visual

Follow the visual layout, taking writing mode and direction into account.

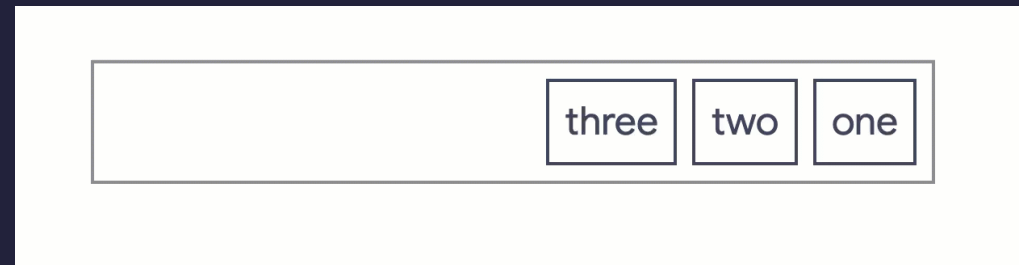
```
.flex {  
  display: flex;  
  flex-direction: row-reverse;  
  reading-flow: flex-visual;  
}
```



reading-flow: flex-flow

Follow the flow order.

```
.flex {  
  display: flex;  
  flex-direction: row-reverse;  
  reading-flow: flex-flow;  
}
```



Adding the order property

The reading flow now takes into account the modification made by order.

```
.flex a:nth-child(3) {  
  order: -1;  
}
```

```
.flex {  
  display: flex;  
  flex-direction: row-reverse;  
  reading-flow: normal;  
}
```



```
.flex a:nth-child(3) {  
  order: -1;  
}
```

```
.flex {  
  display: flex;  
  flex-direction: row-reverse;  
  reading-flow: flex-visual;  
}
```



```
.flex a:nth-child(3) {  
  order: -1;  
}
```

```
.flex {  
  display: flex;  
  flex-direction: row-reverse;  
  reading-flow: flex-flow;  
}
```



Grid and reading flow

Automatic layout—grid-auto-flow: dense

one

two

three

four

five

six

seven

one

two

four

five

three

six

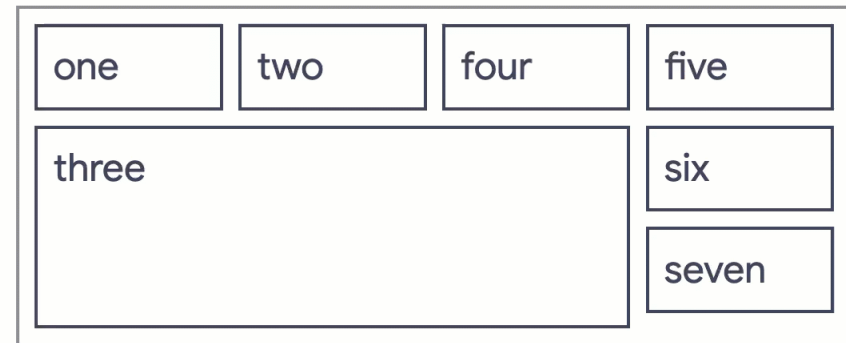
seven

reading-flow: grid-rows

Follow the reading order by row.


```
.grid a:nth-child(3) {  
  grid-column: span 3;  
  grid-row: span 3;  
}
```

```
.grid {  
  display: grid;  
  grid-auto-flow: dense;  
  grid-template-columns: repeat(4,1fr);  
  reading-flow: grid-rows;  
}
```



reading-flow: grid-columns

Follow the reading order by column.

one

two

five

three

six

four

seven

```
.grid a:nth-child(2) {  
  grid-column: span 3;  
  grid-row: span 3;  
}
```

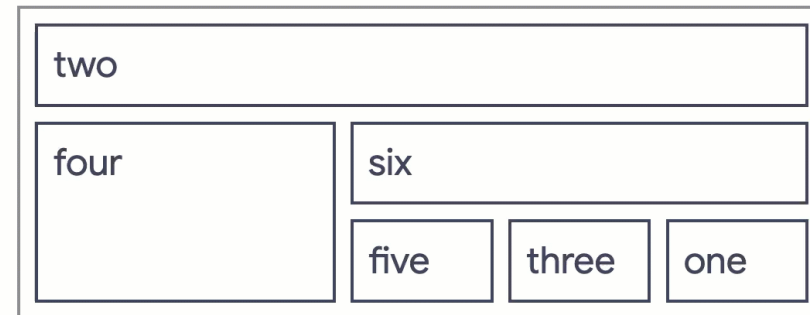
```
.grid {  
  display: grid;  
  grid-auto-flow: column dense;  
  grid-template-columns: repeat(5,1fr);  
  reading-flow: grid-columns;  
}
```



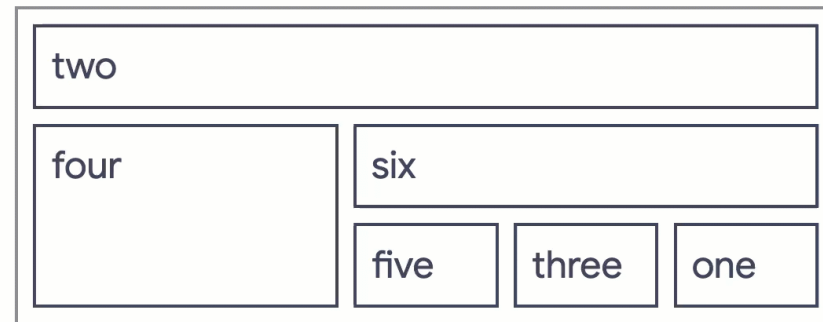
Grid and reading flow

Line-based placement, grid-template-areas

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(5,1fr);  
  grid-template-areas:  
    "b b b b b"  
    "d d f f f"  
    "d d e c a";  
}
```



```
.grid {  
  display: grid;  
  grid-template-columns: repeat(5,1fr);  
  grid-template-areas:  
    "b b b b b"  
    "d d f f f"  
    "d d e c a";  
  reading-flow: grid-rows;  
}
```



reading-flow: grid-order

As normal, but take the order property into account.

two

four

six

five

three

one

The reading-order property

For the children of a reading flow container.



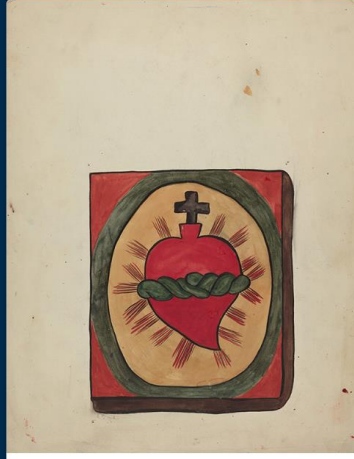
Feeding the Bird
American 19th Century

oil on canvas 1800
Not on View 1953.5.63



Keelmen Heaving in Coals by Moonlight
Joseph Mallord William Turner

oil on canvas 1835
West Building, Main Floor 1942.9.86
Gallery 57

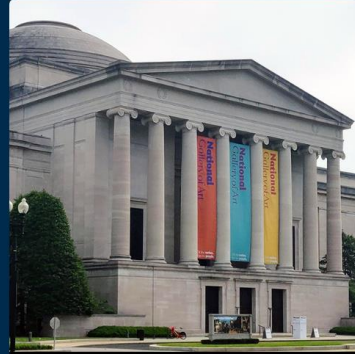


Retablo
E. Boyd

watercolor, colored pencil 1935
Not on View 1943.8.6831



Painted Chest
Majel G. Claflin



National Gallery of Art

The National Gallery of Art is an art museum in Washington, D.C., United States, located on the National Mall, between 3rd and 9th Streets, at Constitution Avenue NW.

Open to the public and free of charge, the museum was privately established in 1937 for the American people by a joint resolution of the United States Congress.

[Visit](#)

```
.grid a:nth-child(3) {  
  grid-column: span 3;  
  grid-row: span 3;  
}
```

```
.grid {  
  display: grid;  
  grid-auto-flow: dense;  
  grid-template-columns: repeat(4,1fr);  
  reading-flow: grid-rows;  
}
```

```
.grid a:nth-child(5) {  
  grid-column: 4;  
  grid-row: 1;  
  grid-order: -1;  
}
```



Source order is important

Never use reading-flow to avoid the work of updating your HTML.

Create a sensible document

Then consider small tweaks to optimize the reading experience at different breakpoints.

Try it out in Chrome Canary

<https://chrome.dev/reading-flow-examples>

Thank
you

Rachel Andrew

<https://noti.st/rachelandrew/mqtF84/>