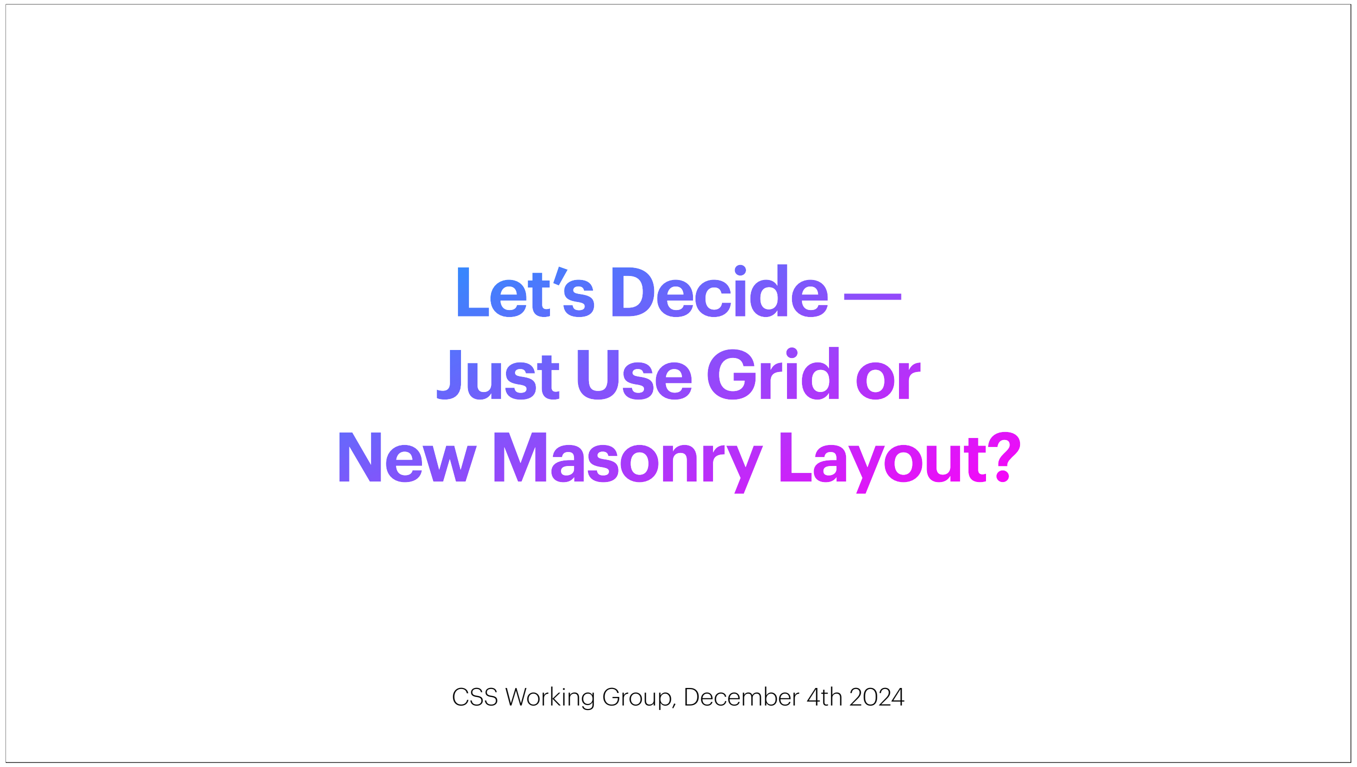
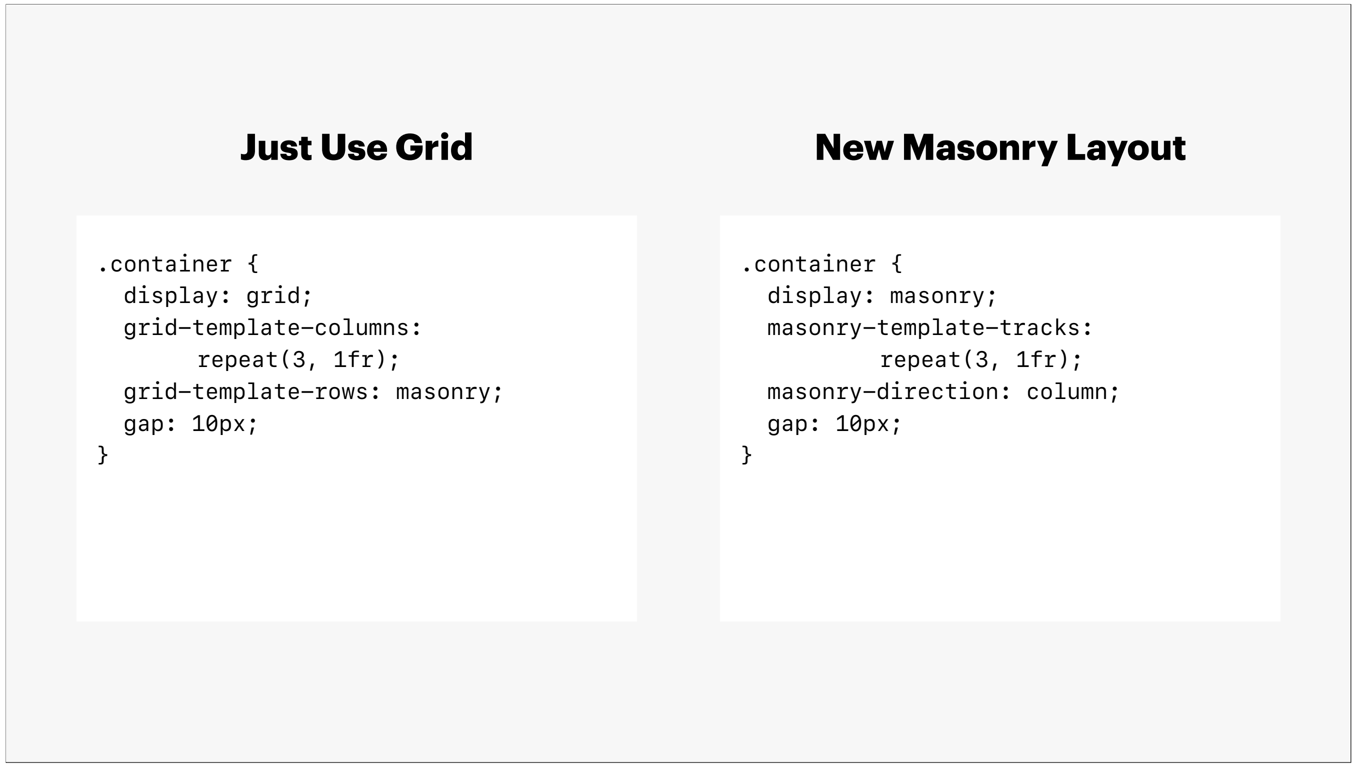


Let’s Decide — Just Use Grid or New Masonry Layout?

CSS Working Group, December 4th 2024

Let’s decide — shall we Just Use Grid or create a whole new Masonry Layout mode?



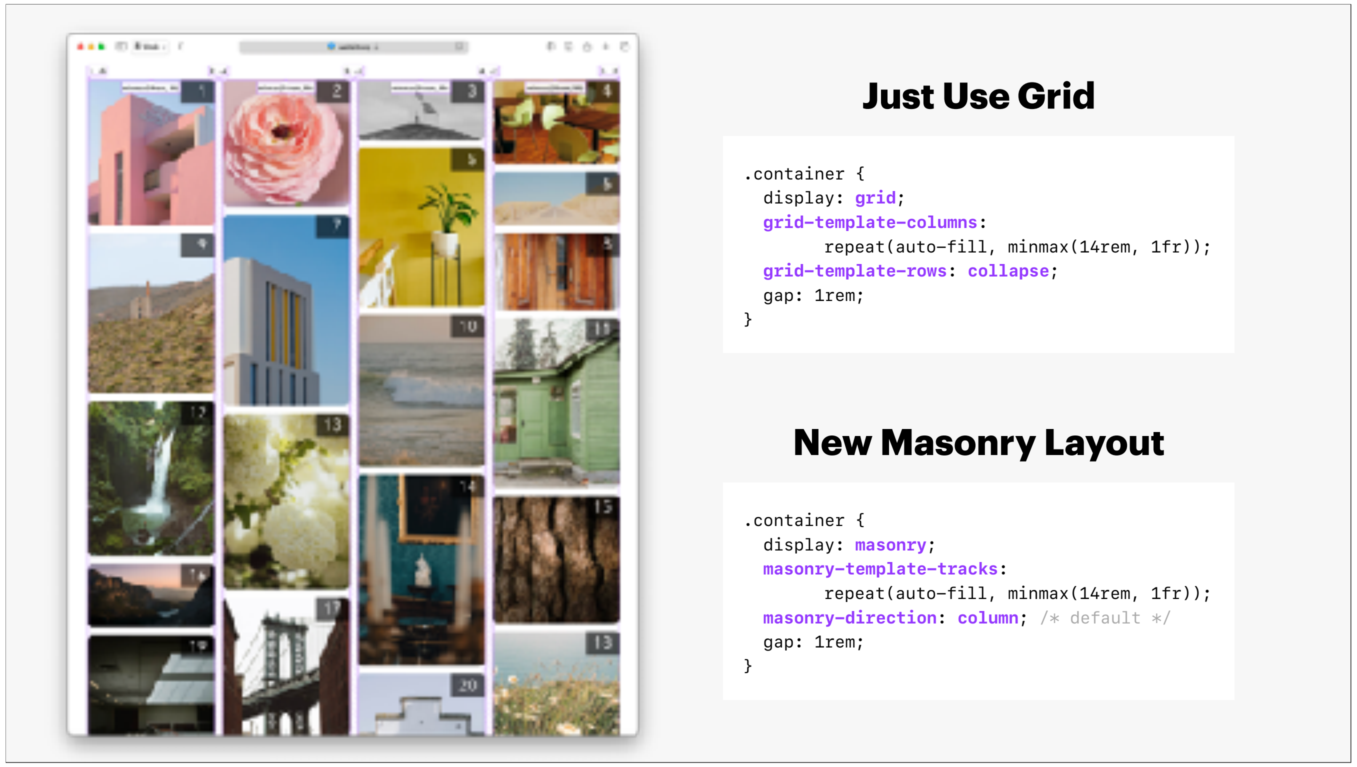
Just Use Grid

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

New Masonry Layout

```
.container {
  display: masonry;
  masonry-template-tracks:
          repeat(3, 1fr);
  masonry-direction: column;
  gap: 10px;
}
```

At this point, the decision is about syntax. About the mental model. The functionality is identical either way.



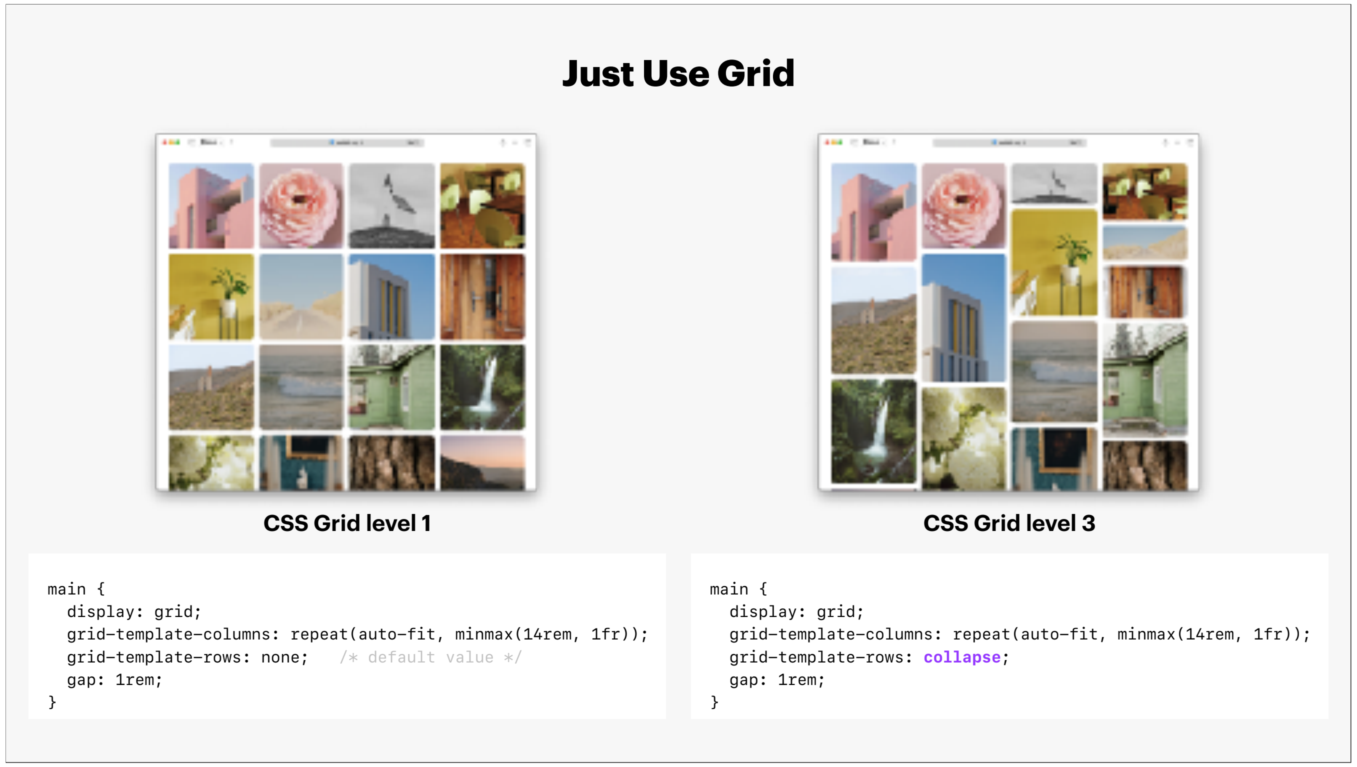
Just Use Grid

```
.container {
  display: grid;
  grid-template-columns:
        repeat(auto-fill, minmax(14rem, 1fr));
  grid-template-rows: collapse;
  gap: 1rem;
}
```

New Masonry Layout

```
.container {
  display: masonry;
  masonry-template-tracks:
        repeat(auto-fill, minmax(14rem, 1fr));
  masonry-direction: column; /* default */
  gap: 1rem;
}
```

To get the layout on the left, in both we ▮▶ declare the display type on the container, ▮▶ define the column sizes, ▮▶ specify which axis gets the masonry-style packing, ▮▶ and create a gap.



Just Use Grid

CSS Grid level 1

```
main {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(14rem, 1fr));
  grid-template-rows: none;   /* default value */
  gap: 1rem;
}
```

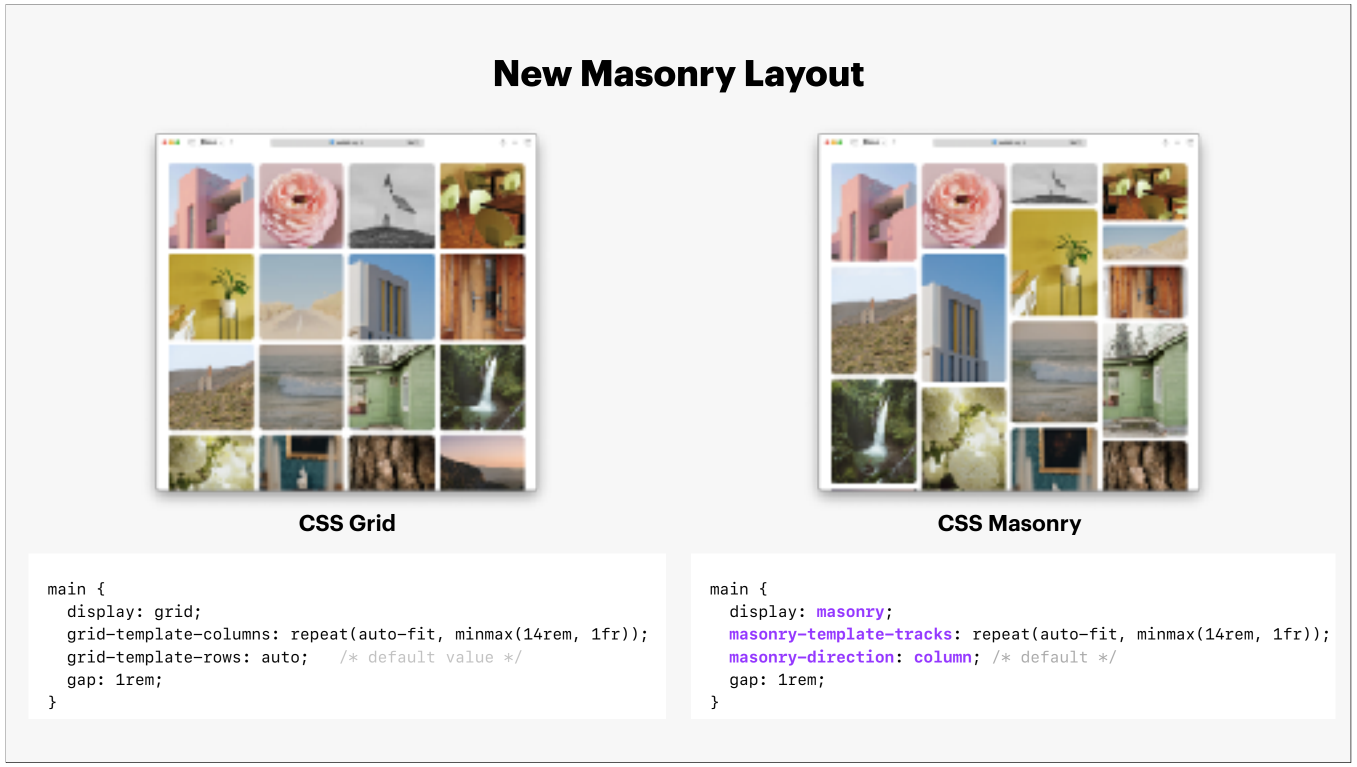
CSS Grid level 3

```
main {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(14rem, 1fr));
  grid-template-rows: collapse;
  gap: 1rem;
}
```

The choice to Just Use Grid is a choice to extend Grid level 1 to now be able to create the kind of packing made popular in masonry / waterfall frameworks.

On the left, we’ve got CSS Grid.
On the right, we add one new value and now Grid has more power.

(BTW “collapse” is just a suggestions for the value — we can bike shed it later).



New Masonry Layout

CSS Grid

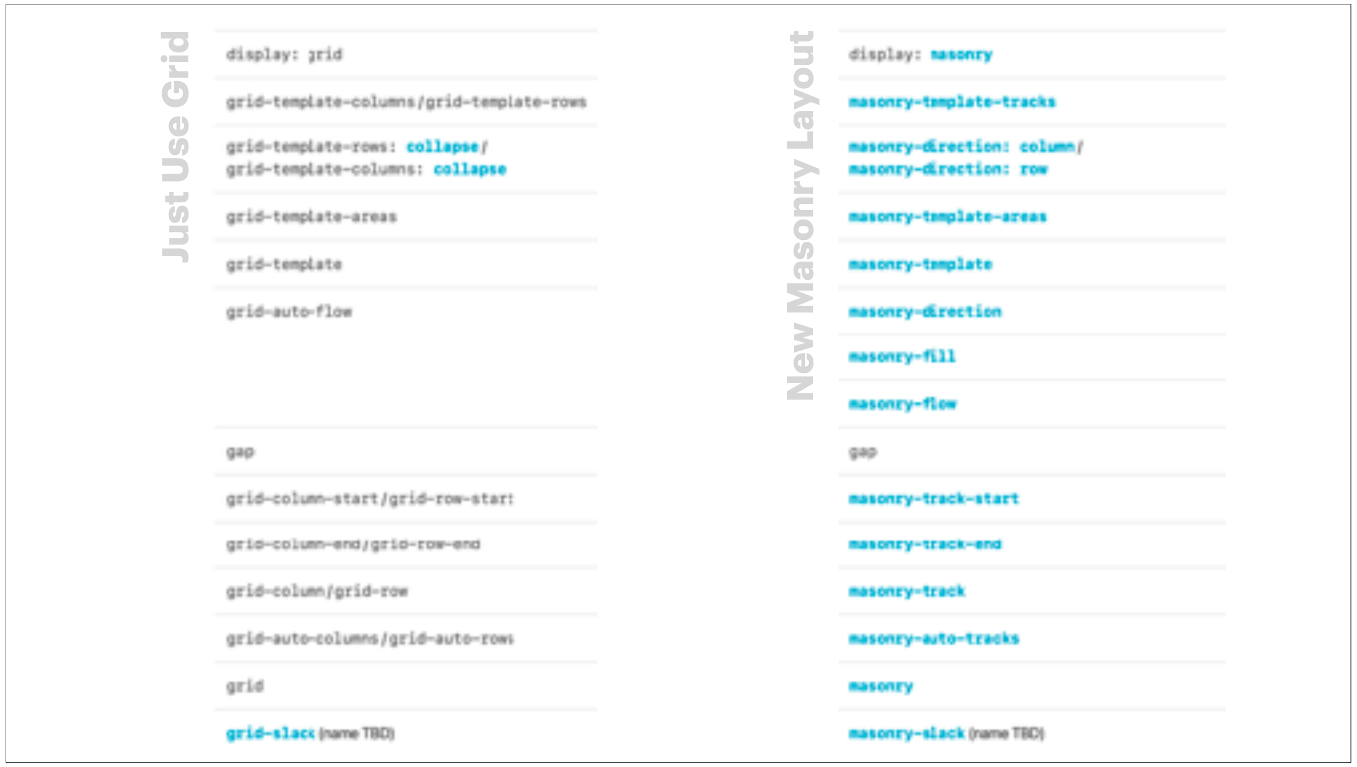
```css
main {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(14rem, 1fr));
  grid-template-rows: auto;   /* default value */
  gap: 1rem;
}
```

CSS Masonry

```css
main {
  display: masonry;
  masonry-template-tracks: repeat(auto-fit, minmax(14rem, 1fr));
  masonry-direction: column; /* default */
  gap: 1rem;
}
```

The choice to create a new layout type called Masonry is a choice to add a whole new layout mode — Grid, Flexbox, Block layout… and now Masonry.

It’s a separate tool, with new syntax that mimics what authors already know, but it’s not the same.



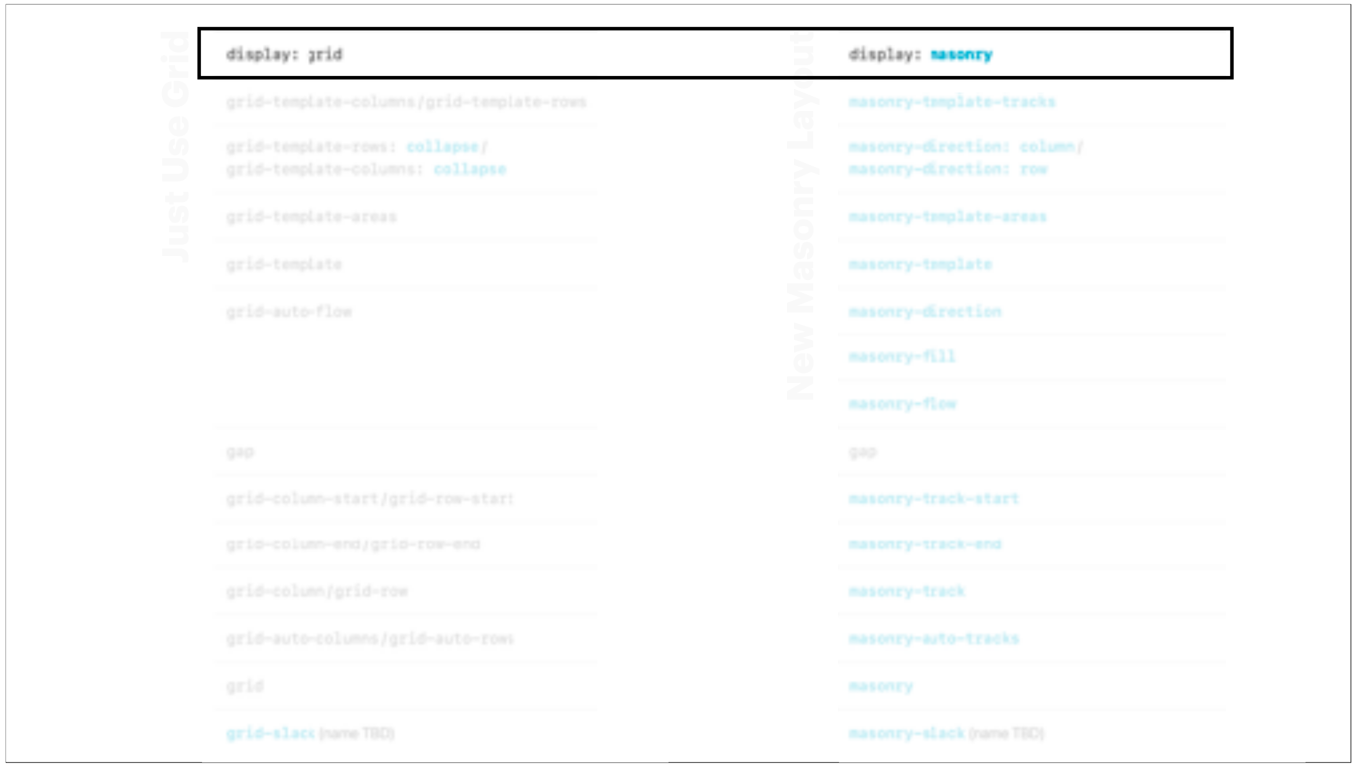
Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

If we just use Grid, we'll add 1 new value, and 1 new property.

(Anything new to CSS is marked here in blue.)

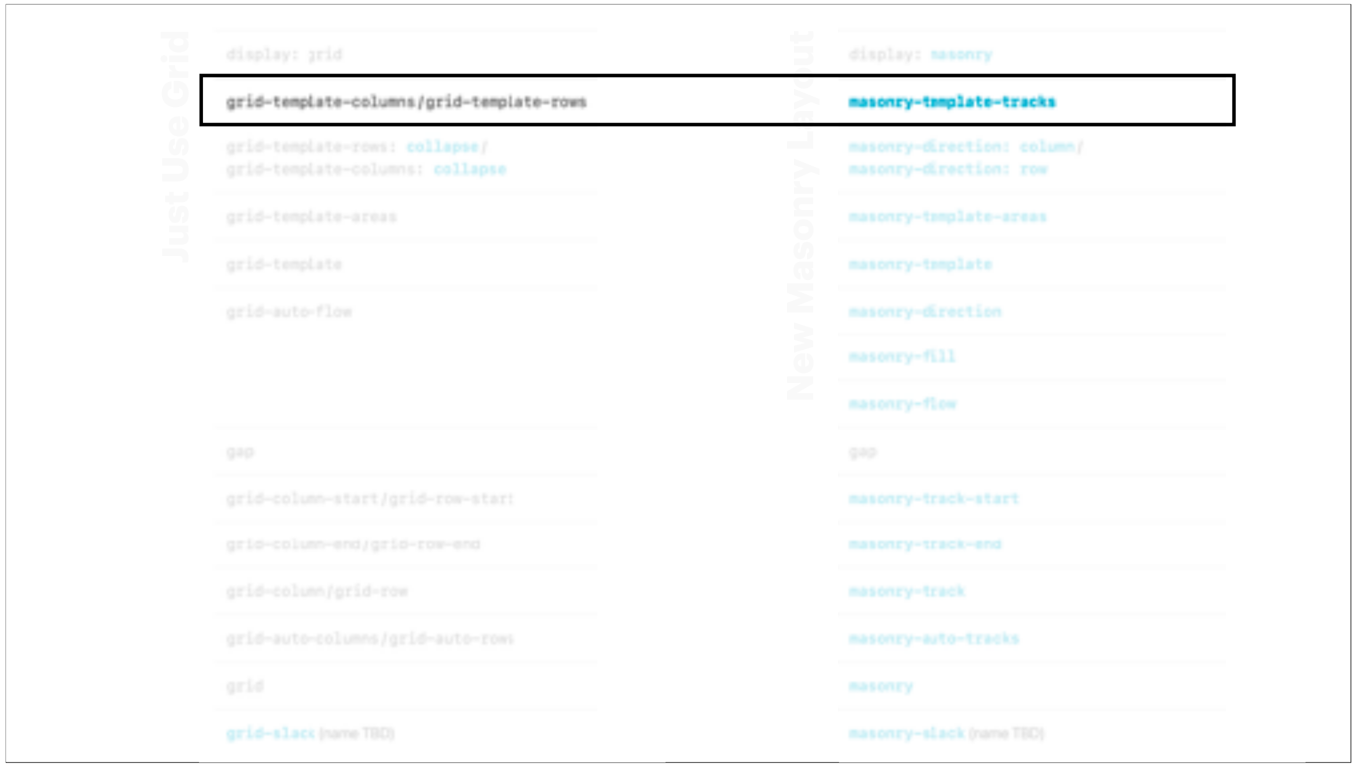
If we create a new Masonry Layout mode, we'll be adding 1 new value and 12 new properties to CSS.

We believe there's just not a compelling enough argument to warrant adding all of these new properties. But let's dig into it…



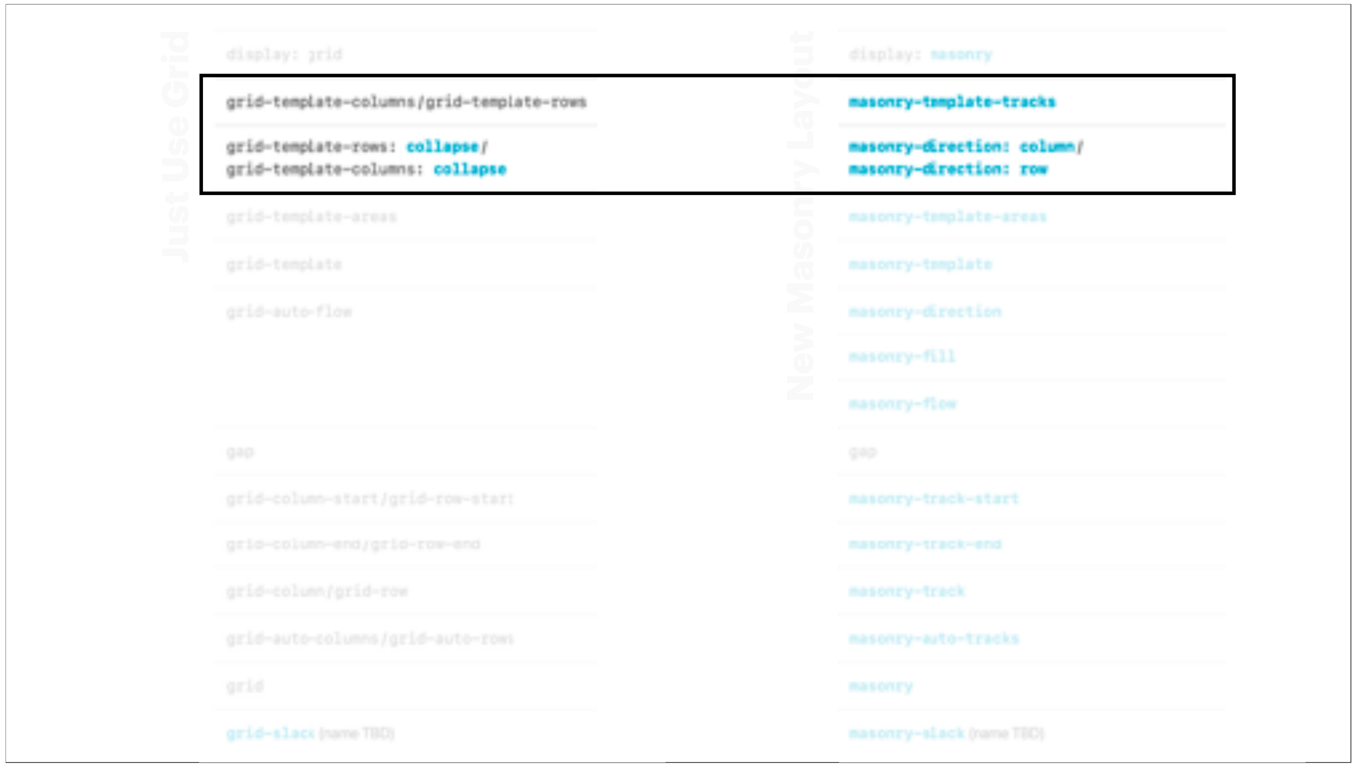
Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

Both use the `display` property to get things started.



Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

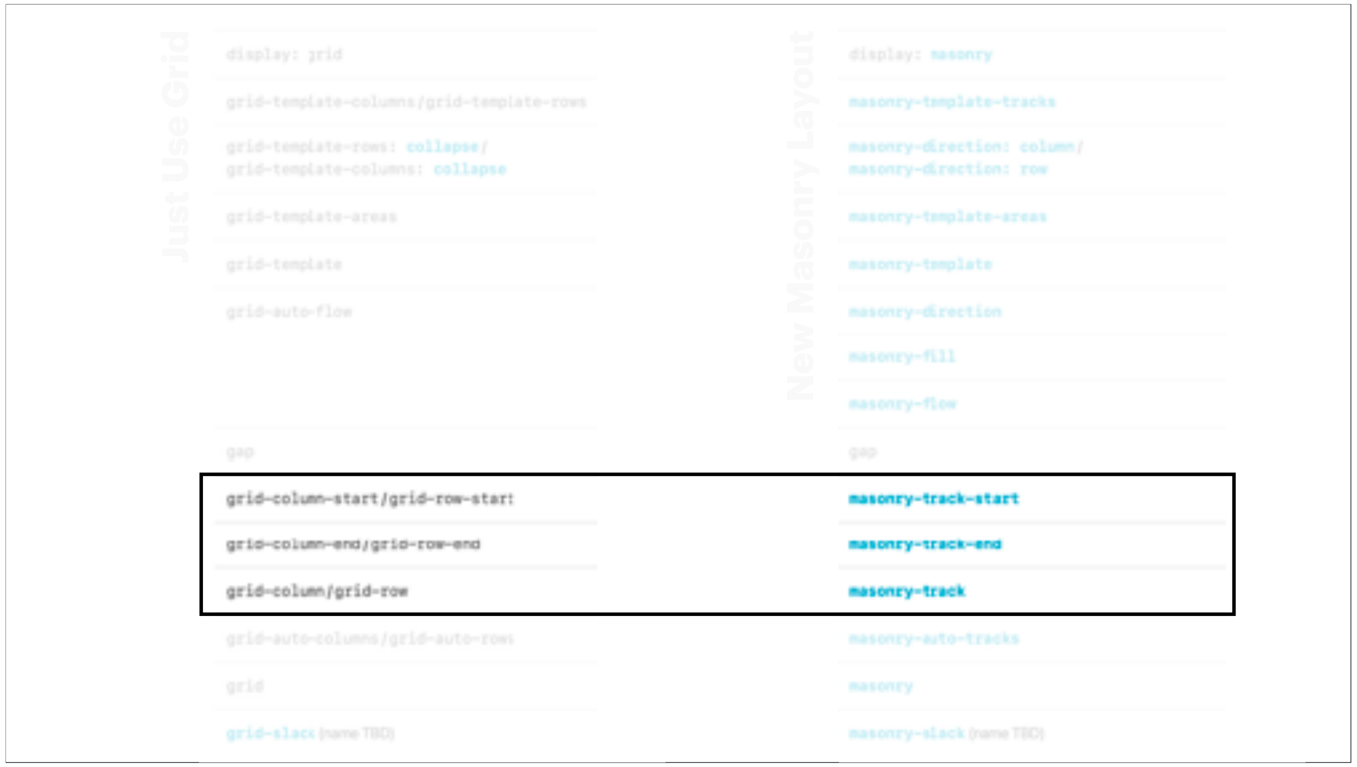
With the template properties you define track sizes.



Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

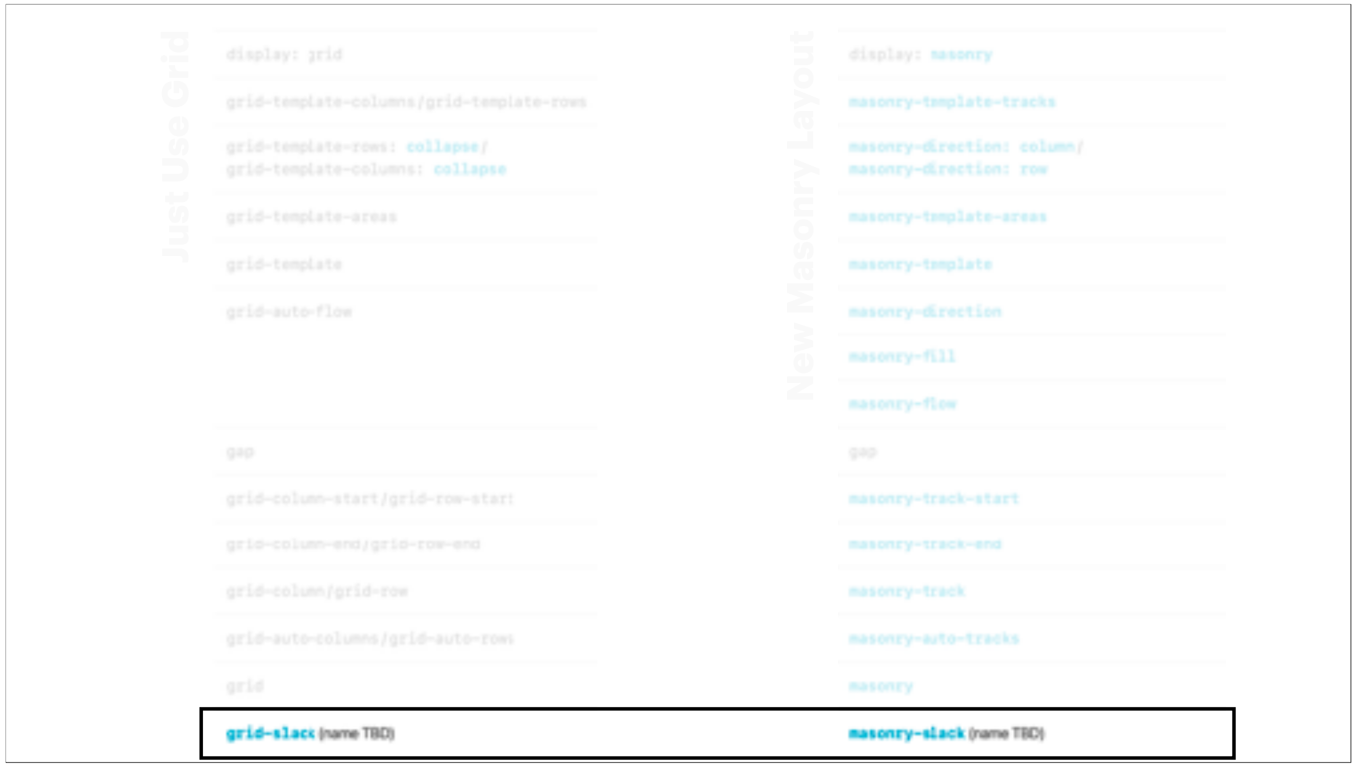
To specify which dimension gets packed masonry-style —
in Grid, you “collapse” the content together
by declaring so in the appropriate track definition.

In the new Masonry Layout, you instead use
`masonry-direction`.



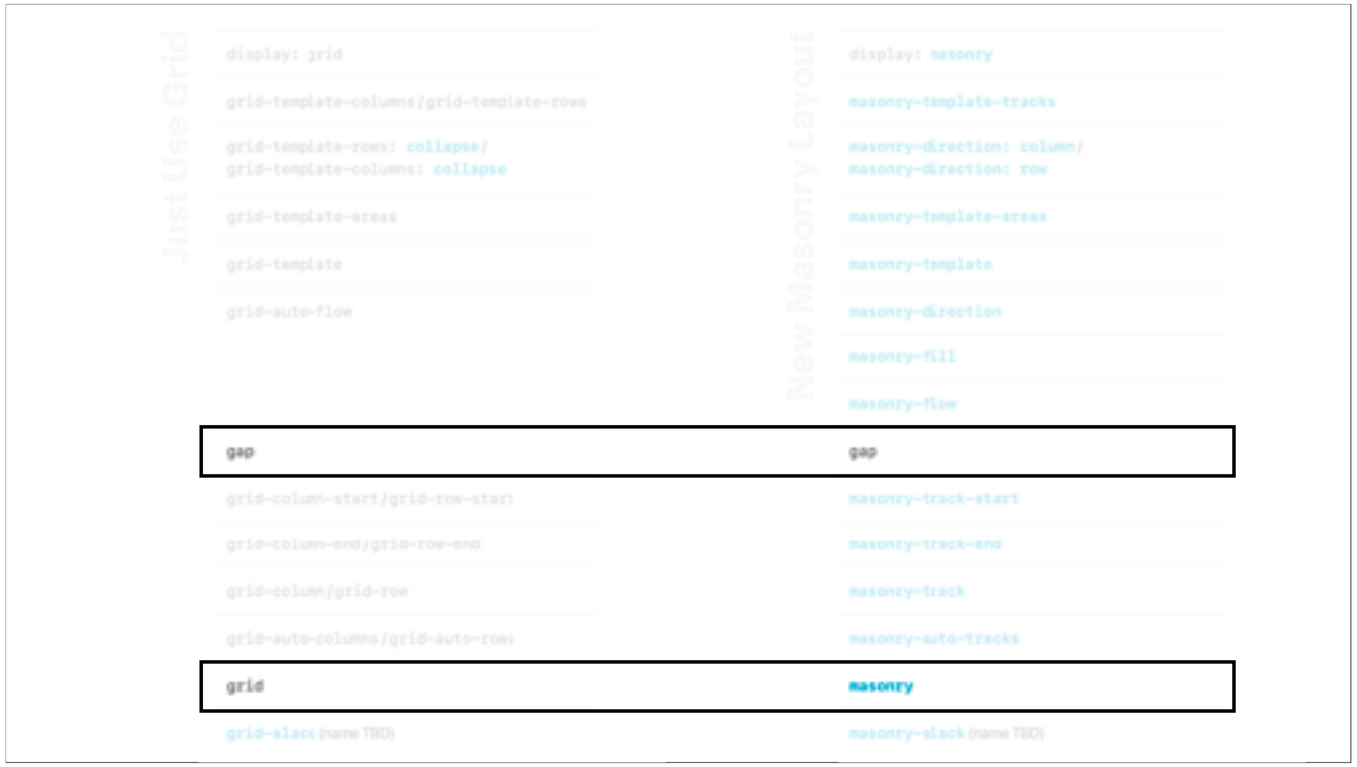
Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

They both have ways to define explicit placement of content…



Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

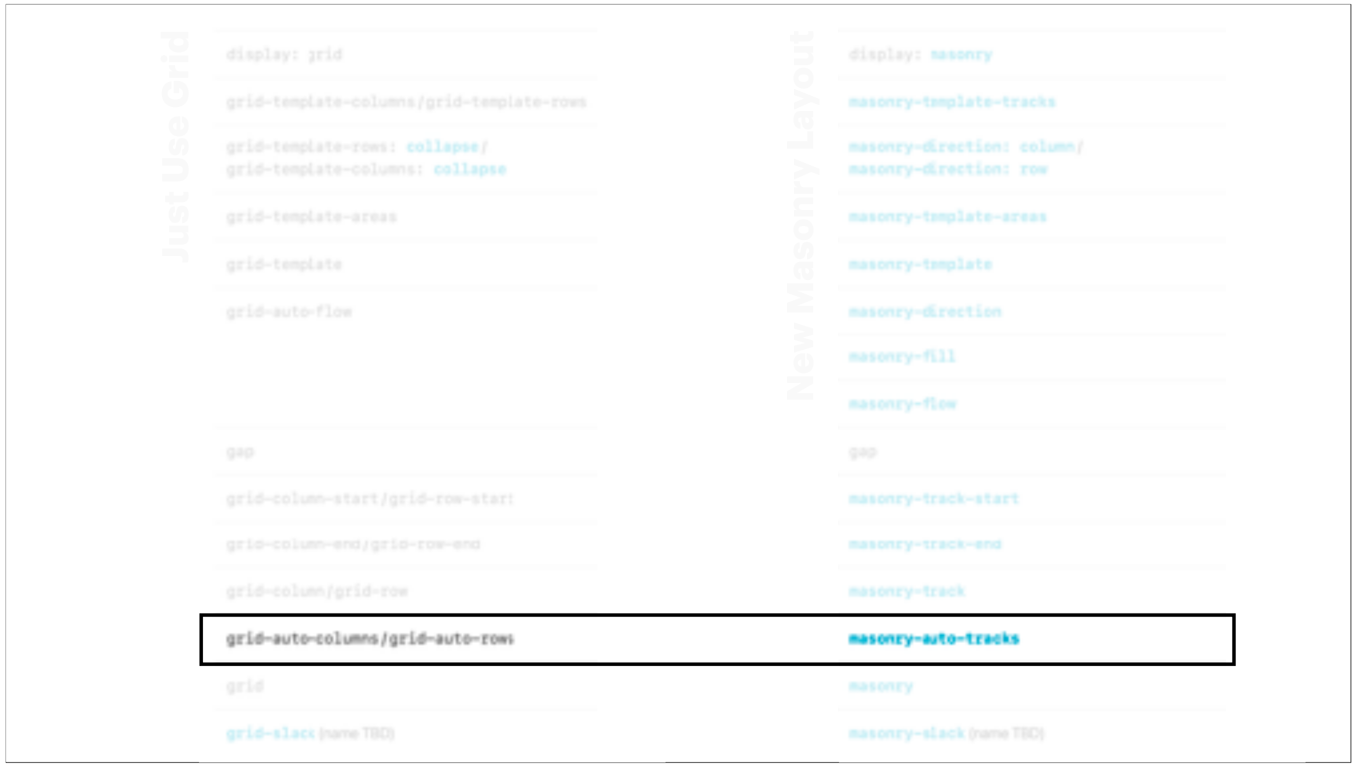
Both have a new `slack` property that adjusts how picky
the browser is when selecting the next space in which to place an item.



Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

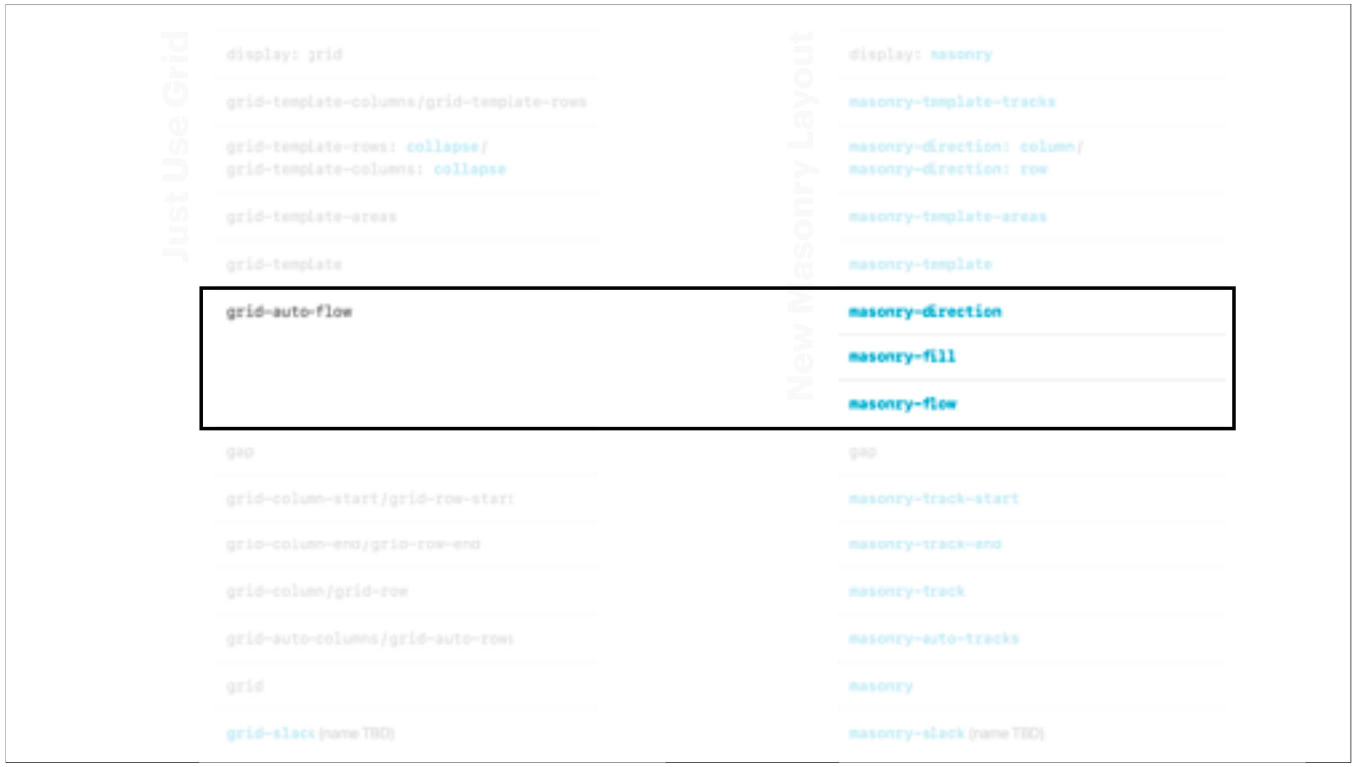
Both options support the `gap` property.

Both have an extra-short hand.



Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

Both have a way to size implicit tracks.

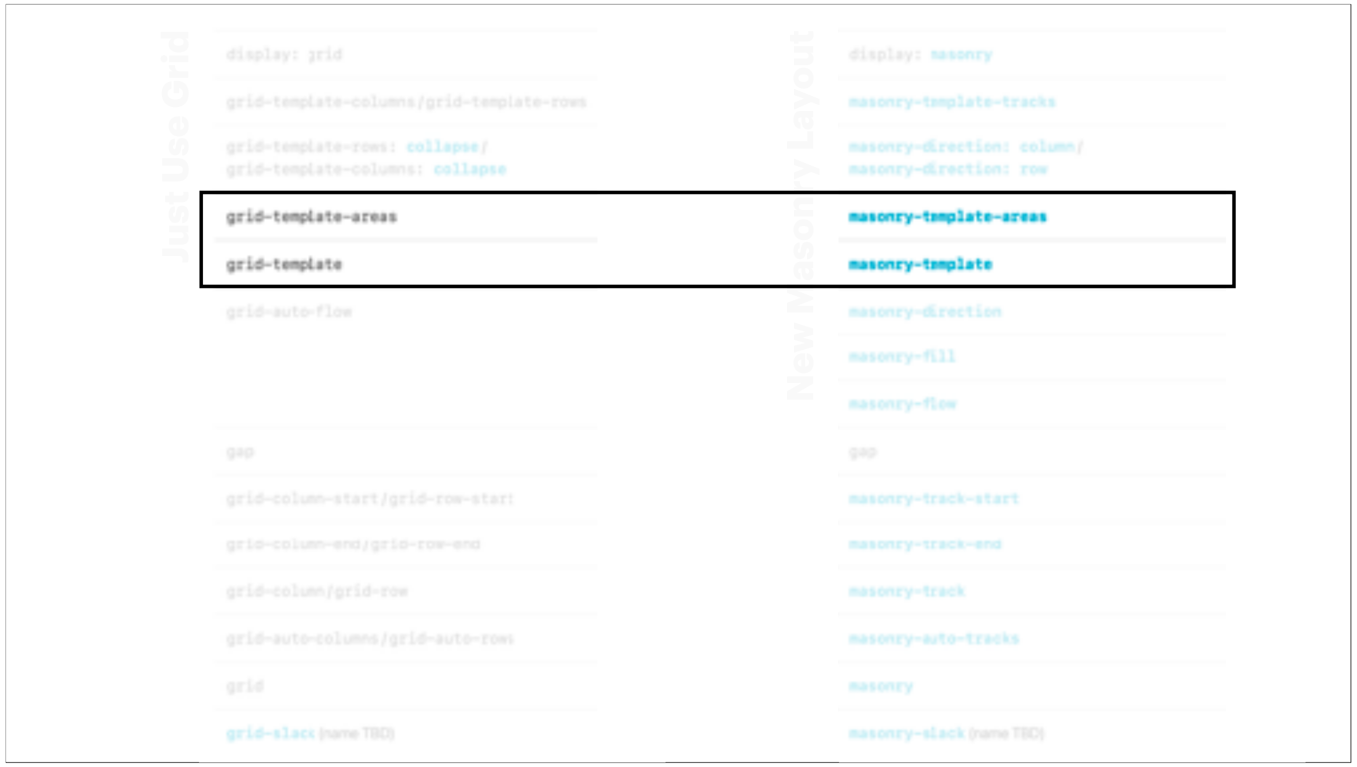


Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

Here’s a more significant difference… There are three new properties in Masonry — masonry-direction, fill and flow that cover the functionality that’s covered in CSS Grid by `grid-auto-flow` — to control how the auto-placement algorithm works.

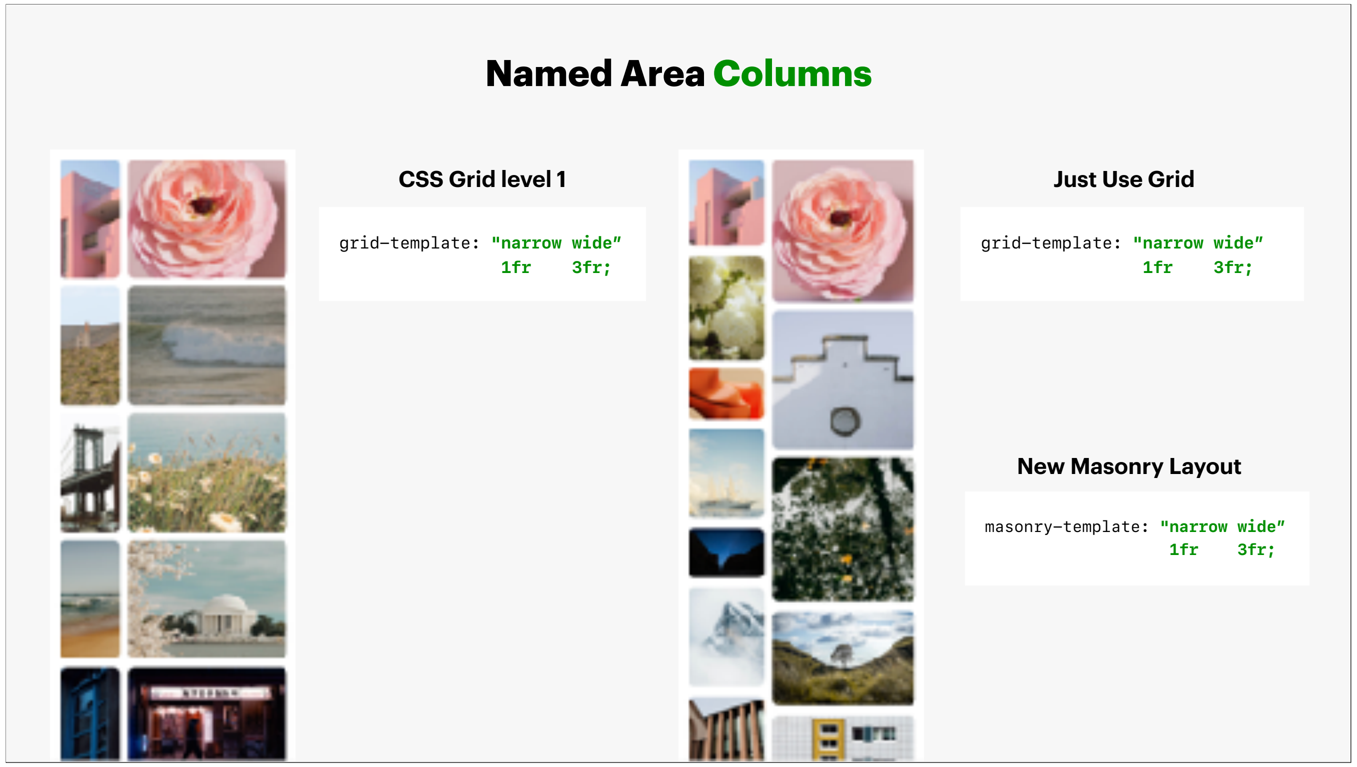
Chrome argues that masonry-direction, fill and flow will be more understandable than grid-auto-flow. That without making this change authors will be confused.

We don’t agree. We believe that since it does the same thing it does for CSS Grid, authors will understand it. If they learned Grid, they already know it.



Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns/grid-template-rows	masonry-template-tracks
grid-template-rows: collapse/ grid-template-columns: collapse	masonry-direction: column/ masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start/grid-row-start	masonry-track-start
grid-column-end/grid-row-end	masonry-track-end
grid-column/grid-row	masonry-track
grid-auto-columns/grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

Both have the ability to define named areas, including with the ‘ascii art’ technique. But there’s also a difference here.



Named Area Columns

CSS Grid level 1

```
grid-template: "narrow wide"
                1fr    3fr;
```

Just Use Grid

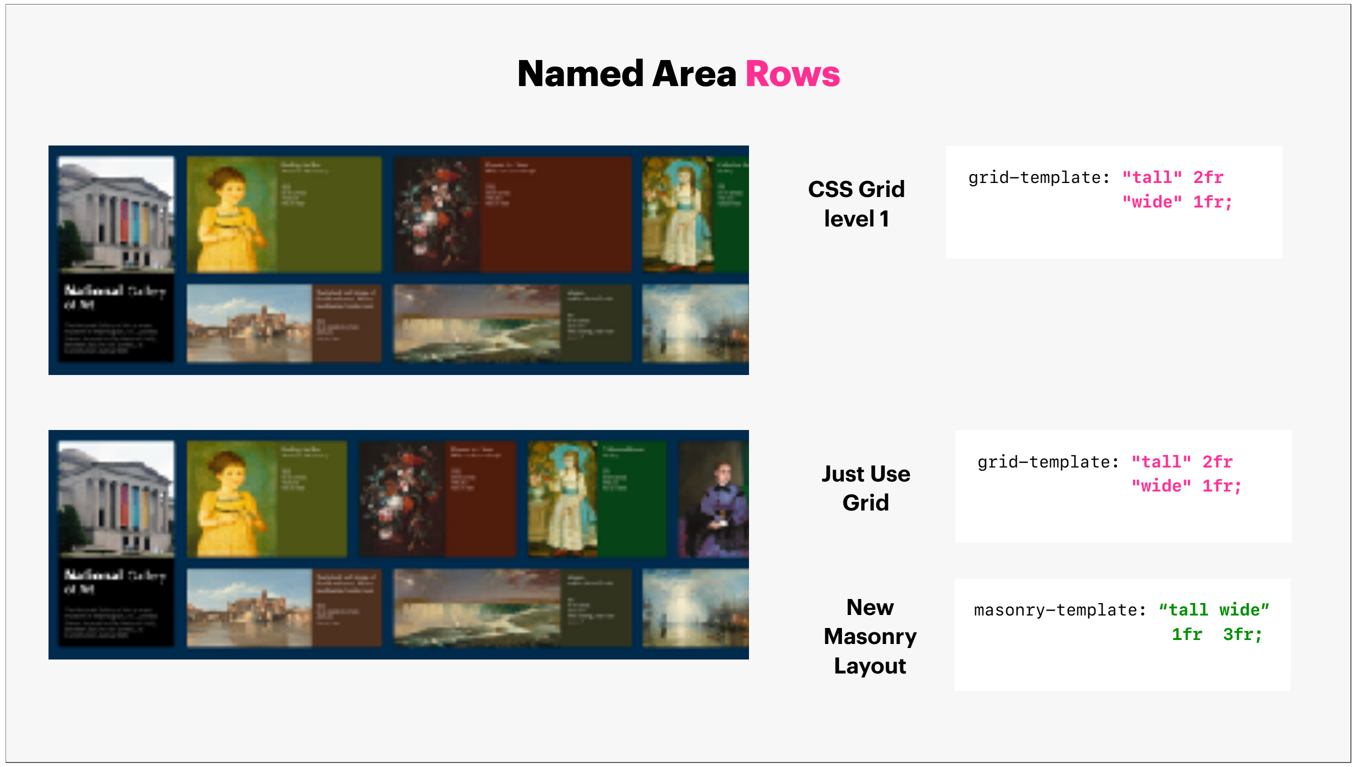
```
grid-template: "narrow wide"
                1fr    3fr;
```

New Masonry Layout

```
masonry-template: "narrow wide"
                   1fr    3fr;
```

In CSS Grid, if you want to create a “narrow” 1fr column &
a “wide” 3fr column — you can use this syntax, effectively diagramming the names and sizes of the columns.

⏯ For both the Just Use Grid and New Masonry Layout options, the syntax is identical to Grid — when defining columns.



Named Area Rows

CSS Grid level 1

```
grid-template: "tall" 2fr
               "wide" 1fr;
```

Just Use Grid

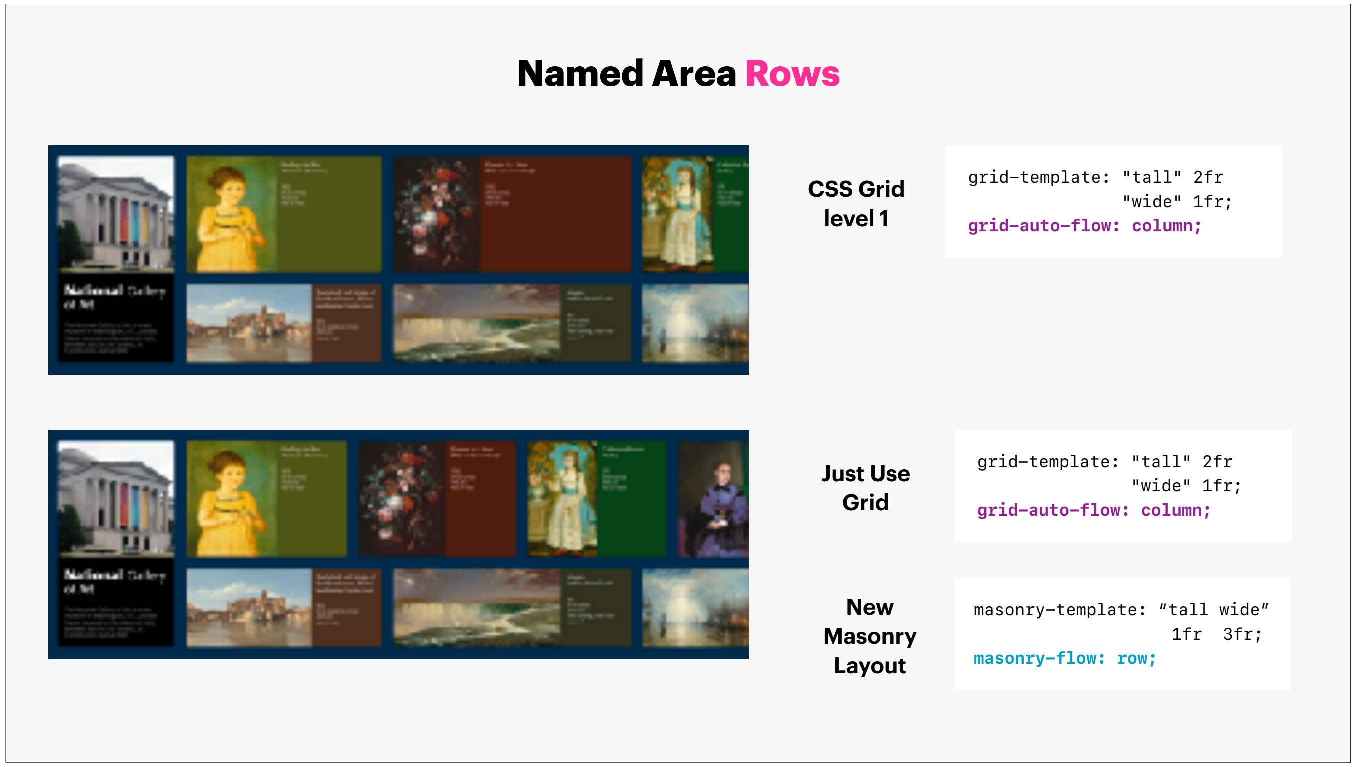
```
grid-template: "tall" 2fr
               "wide" 1fr;
```

New Masonry Layout

```
masonry-template: "tall wide"
                    1fr  3fr;
```

▶ For laying out rows, Grid's template syntax is a little different, stacking the area names to diagram the names and sizes of the rows. ▶ When switching to a masonry-style packing, Just Use Grid re-uses this syntax exactly.

▶ New Masonry Layout, however, sticks with the named columns template syntax. Chrome argues that using the same syntax in both axes is better. We believe matching the existing Grid template API is better.



Named Area Rows

CSS Grid level 1

```
grid-template: "tall" 2fr
               "wide" 1fr;
grid-auto-flow: column;
```

Just Use Grid

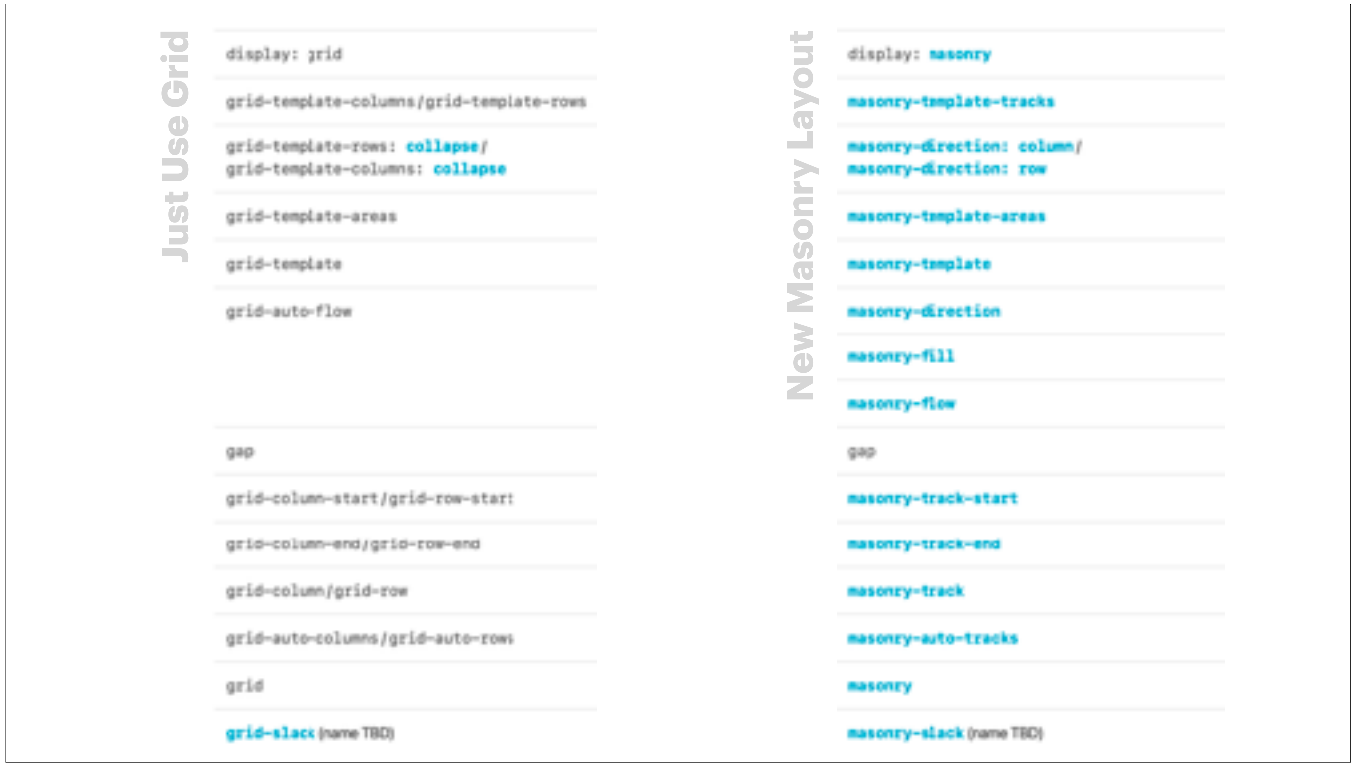
```
grid-template: "tall" 2fr
               "wide" 1fr;
grid-auto-flow: column;
```

New Masonry Layout

```
masonry-template: “tall wide”
                   1fr  3fr;
masonry-flow: row;
```

You can also see the difference between `grid-auto-flow` vs `masonry-flow`, (with it’s `masonry-fill` & `masonry-direction` longhands) in this example.

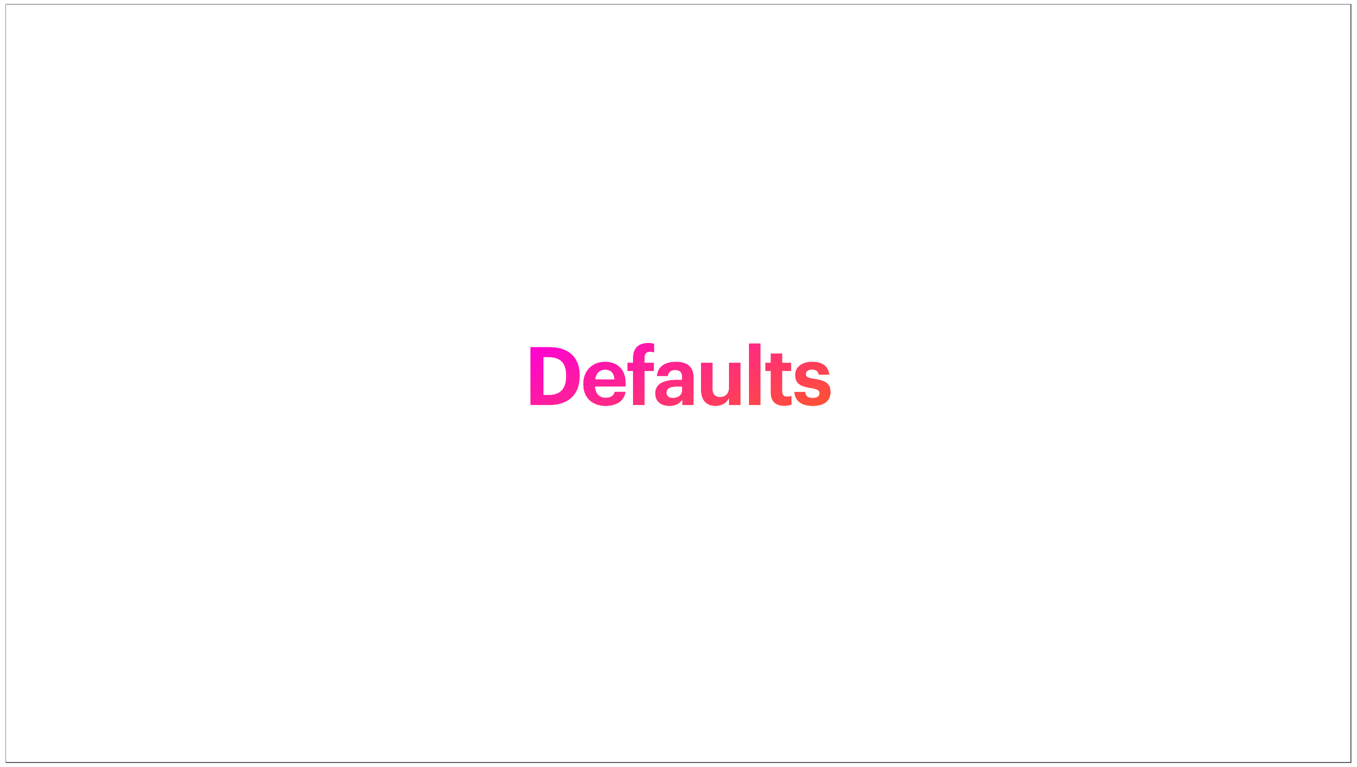
Flexbox’s `flex-flow` property indicates both the orientation of lines as well as the primary placement axis. But in Masonry, these two axes are different. Grid’s `grid-auto-flow` value indicates the primary fill direction; and Just Use Grid follows this convention. Chrome argues that the flow value should indicate the orientation of lines, rather than placement, so they’ve changed it to `masonry-flow: row`. This might not be an unreasonable idea, but is it worth creating an entire new layout system for such tweaks?



Just Use Grid	New Masonry Layout
display: grid	display: masonry
grid-template-columns / grid-template-rows	masonry-template-tracks
grid-template-rows: collapse / grid-template-columns: collapse	masonry-direction: column / masonry-direction: row
grid-template-areas	masonry-template-areas
grid-template	masonry-template
grid-auto-flow	masonry-direction
	masonry-fill
	masonry-flow
gap	gap
grid-column-start / grid-row-start	masonry-track-start
grid-column-end / grid-row-end	masonry-track-end
grid-column / grid-row	masonry-track
grid-auto-columns / grid-auto-rows	masonry-auto-tracks
grid	masonry
grid-slack (name TBD)	masonry-slack (name TBD)

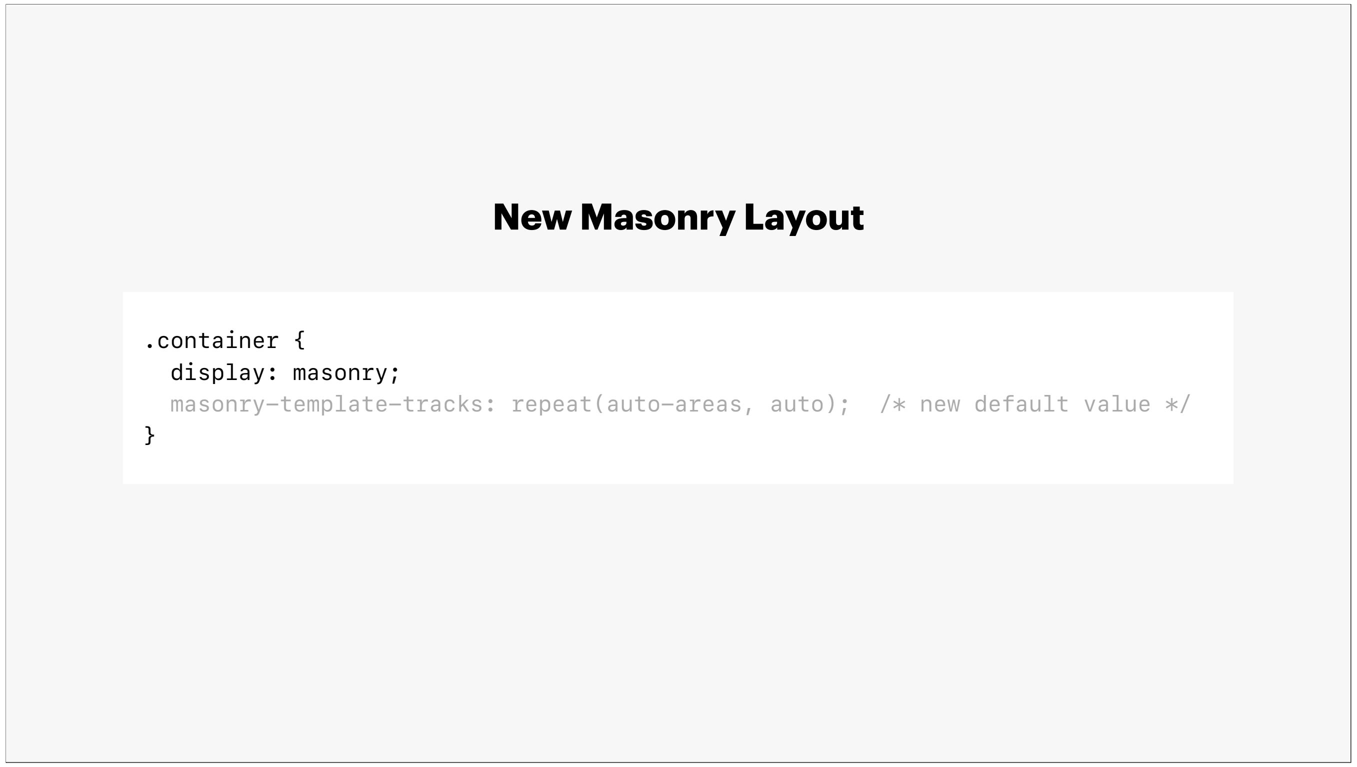
Overall — there's a lot of new syntax being proposed in New Masonry layout — most of which does the same thing. But some of which is subtly different in a way that's likely to trip people up.

Adding Masonry-style features to CSS is not an opportunity to fork Grid and rewrite past decisions. Authors will just get frustrated and confused.



Defaults

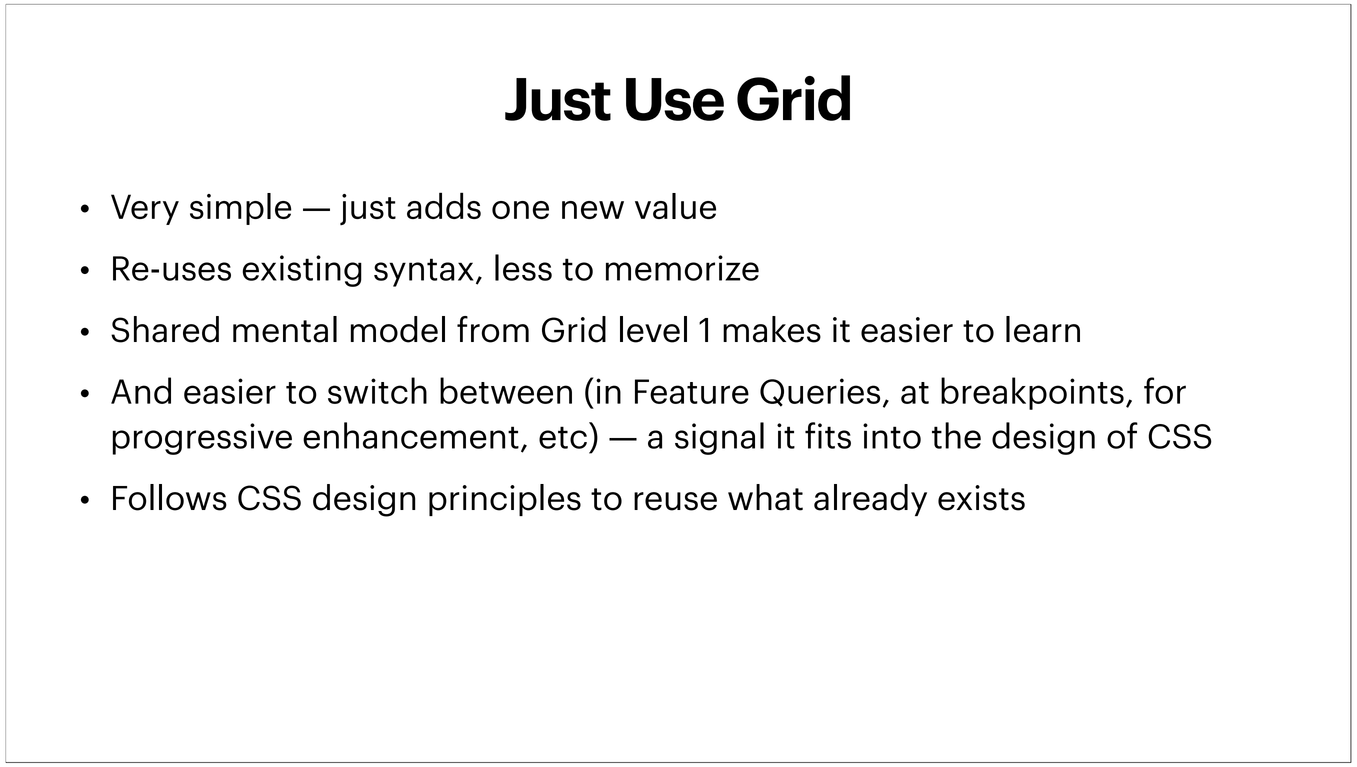
Chrome argues that the new syntax is worth the effort because it creates better defaults. We do not believe their proposed defaults are actually that useful. And that it’s not worth it to create a whole new display type to change the defaults.



New Masonry Layout

```
.container {
  display: masonry;
  masonry-template-tracks: repeat(auto-areas, auto);  /* new default value */
}
```

Specifically, there’s an idea that all an author would have to write is `display: masonry` and they would get a typical Pinterest style layout with just that one line of code. We wrote several thousand words in our article walking through how auto sizing would affect the layout and showing that it’s not actually a very useful default. It just moves the responsibility for sizing to the items, and often leaves authors in a very hard to understand state. Understanding `auto` sizing is the hardest thing in CSS, and authors do not understand it.



Just Use Grid

- Very simple — just adds one new value
- Re-uses existing syntax, less to memorize
- Shared mental model from Grid level 1 makes it easier to learn
- And easier to switch between (in Feature Queries, at breakpoints, for progressive enhancement, etc) — a signal it fits into the design of CSS
- Follows CSS design principles to reuse what already exists

We want to Just Use Grid. It’s very simple. It just adds one new value.
It re-uses existing syntax, making for far less for authors to memorize.
Using a shared mental model from Grid level 1 makes it easier to learn.
And using a shared mental model with Grid level 1 makes it easier to switch between Grid 1 and the new capabilities in Feature Queries, at breakpoints, for progressive enhancement. That’s a signal it fits into the design of CSS.
And it follows CSS design principles to reuse what already exists.