

Why 'display: masonry'?



CSSWG Meeting 12/4/2024

#1 Property/value differences

- Only applicable in Masonry:
 - `masonry-slack`
 - `masonry-auto-flow`
 - Intrinsic auto repeaters `repeat(auto-fill, auto)`
- Only applicable in Grid:
 - `grid-auto-flow`

#1 Property/value differences

- Behave differently in each, or have different valid values:
 - `grid-template-areas`
 - `grid-template`
 - Track sizing values for `grid-template-columns` and `grid-template-rows`
 - Grid has four placement properties, masonry only has two
 - Only grid can use all six `justify-*` and `align-*` properties.
- What happens in error cases?

#2 Better property defaults

- `masonry-template-tracks` defaults to `repeat(auto-areas, auto)`

```
display: masonry
```



```
display: grid  
grid-template-rows: masonry
```



A note on Grid as the default fallback

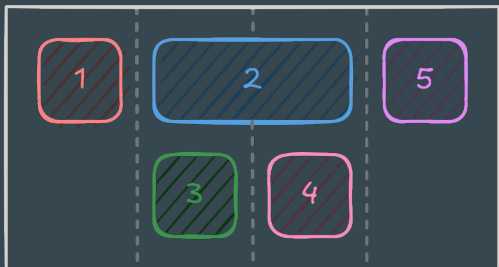
- Needing a fallback for Masonry is a temporary problem
- Grid may be a good fallback in some cases
- Authors should make the choice of how to fallback consciously to avoid potential surprises with an automatic fallback

#3 Positioning items is less complex

- Only allowed in Grid axis
- Since Grid is multi-dimensional, authors must carefully remember what direction placement is allowed
- If authors get it wrong with Grid, it will parse but not work
- With `display: masonry`, `masonry-track-*` applies to the Grid axis alone, which makes switching the Masonry direction seamless

#3 Positioning items is less complex

```
display: grid
grid-template-columns: repeat(4, 1fr)
grid-template-rows: masonry
```



```
grid-column: span 2
grid-column: 2
grid-column: 3
```

```
display: grid
grid-template-columns: masonry
grid-template-rows: repeat(4, 1fr)
```



```
grid-column: span 2
grid-column: 2
grid-column: 3
```

*Need to be redefined as grid-row

#4 Shorthands are more useful

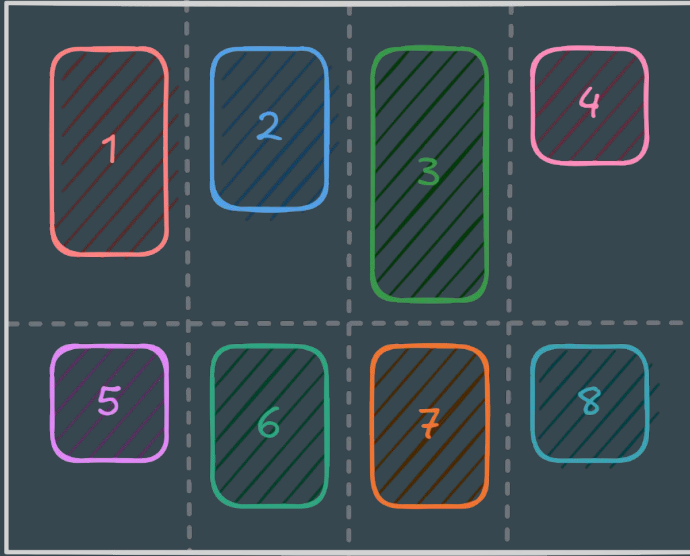
- Grid shorthand is complex given its 2 dimensional nature, and it is not widely used by authors
- With `display: masonry` the shorthand syntax is simpler, and thus, more likely to be utilized by authors

```
.masonry {  
  display: grid;  
  grid: repeat(3, 1fr) / masonry;  
}
```

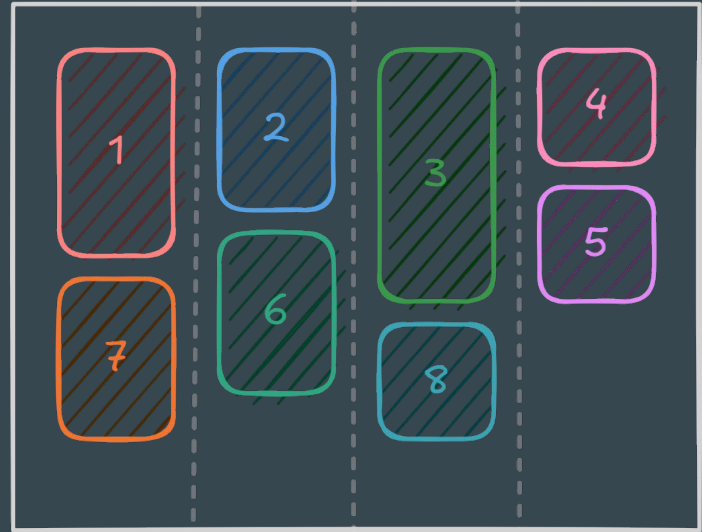
```
.masonry {  
  display: masonry;  
  masonry: repeat(3, 1fr) row;  
}
```


#5 Masonry placement != Grid placement

display: grid



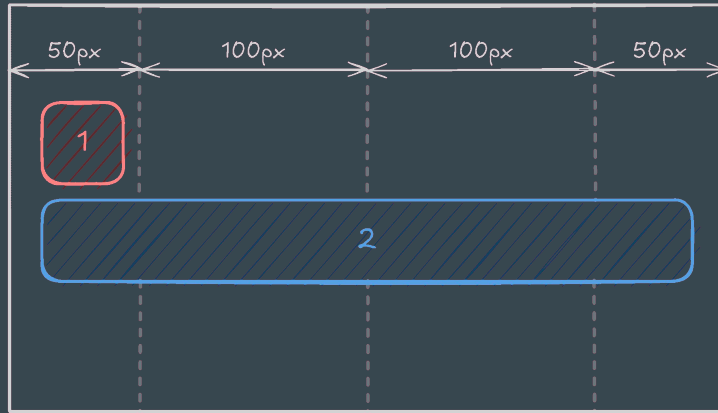
display: masonry



#5 Masonry placement != Grid placement

- Due to the interaction of spanners within a Masonry, Masonry will need to define how dense packing works in a different way than Grid

`grid-auto-flow: dense`



#6 Masonry implies different intrinsic sizing

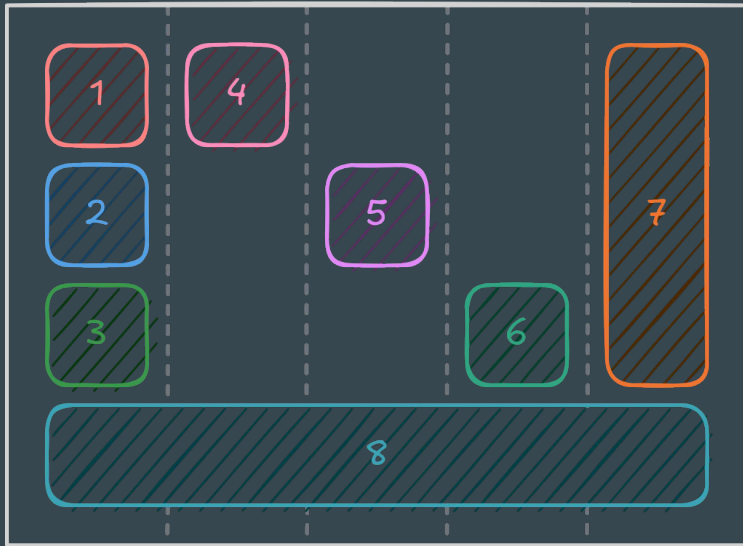
```
grid-template-columns: repeat(3, auto)  
grid-template-rows: none
```



```
grid-template-columns: repeat(3, auto)  
grid-template-rows: masonry
```



#7 Alignment needs to be defined differently



`align-self: start`

`align-self: center`

`align-self: end`

`align-self: stretch`

`masonry-track: span 3`

#8 Subgrid needs to be defined differently

- In Masonry layout, auto-placed subgrids don't inherit any line names from their parent grid, because that would require the subgrid to have a single definite placement, which is not possible in the Masonry placement algorithm since final placement is determined after track sizing is completed

#8 Subgrid needs to be defined differently

- As Submasonry gets further fleshed out, we expect to find more areas of divergence that will need to be explicitly defined in the spec
 - E.g., how to account for auto-placed items in a Submasonry within a Grid ancestor? Likely we will need to extend the concept of the virtual item groups further than just in Masonry's track sizing

#9 Reduced spec complexity & improved teachability

All of these differences between Grid & Masonry from points 1 to 8 lead to the following if Masonry is incorporated into CSS Grid Level 3:

- Spec bloat
- Reduced teachability
- Developer confusion

Our conclusion:

Masonry should be a separate display type

Further resources

[BlinkOn talk discussing support for 'display: masonry'](#)

Ethan Jimenez

[“How Should we define CSS Masonry”](#)

Rachel Andrew, Ian Kilpatrick, Tab Atkins

[Github 'display: masonry' proposal and discussion](#)

CSSWG issue #9041