

Does it work?

Rachel Andrew @ East Bay WP Meetup



CSS IS
AWESOME

CSS is too trivial for real developers to worry about, yet too hard for them to understand.

What I learned from Twitter ...

CSS is unique, and amazing.

We can't break the web



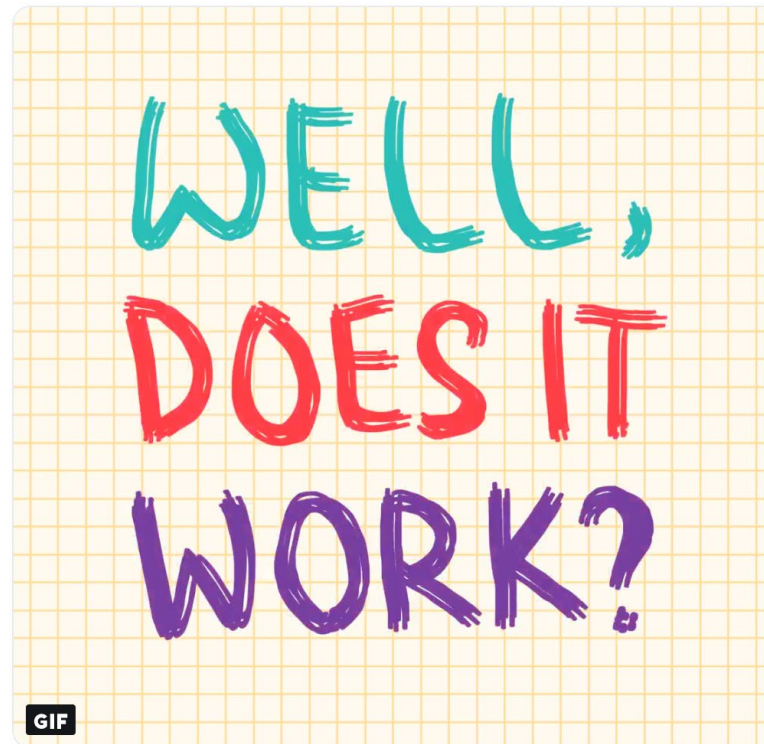
Meowlivia_

@meowlivia_

Follow



A quote by [@rachelandrew](#) during her workshop. This is her response when someone expressed concern about whether they're using CSS Grid correctly. I'm so happy about this! I always feel like I'm not utilizing CSS Grid "the right way" but it works... so... 🙄 😊



12:59 PM - 12 May 2019

Modern CSS can be **minimal**

So you *really* need to understand what those few lines do.

Normal Flow

The no-layout layout.

Just some HTML and content

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

Source order matters

Source order matters

- Ite¹ ne
- Ite² wo
- Ite³ ree

Veggies es bonus vobis, proinde vos postulo essum may⁴ hrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet⁵ reens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Tab stop visualization with <https://accessibilityinsights.io/>

Get your source ordered

Work with document flow and CSS is much simpler.

“Grid or Flexbox”

This is a terrible question.

Columns

Absolutely
positioned
sidebar.

Past layout

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

“He then employed the little-used CSS
“float” property to float the content to the
desired width”

February 2001: <https://alistapart.com/article/journey/>

“ It uses float:left, suffers from needing percentage widths for each column, and from potential column wrapping when the browser window is narrowed”

April 2001: <https://web.archive.org/web/20010413091021/http://glish.com/css/>

Multi-column Layout

```
.columns {  
  column-count: 3;  
}
```

I want columns

Section 1

Veggies es bonus vobis,
proinde vos postulo
essum magis kohlrabi
welsh onion daikon
amaranth tatsoi tomatillo
melon azuki bean garlic.

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 2

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 3

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 4

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 5

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

```
section {  
  break-inside: avoid;  
}
```

I want columns

Section 1

Veggies es bonus vobis,
proinde vos postulo
essum magis kohlrabi
welsh onion daikon
amaranth tatsoi tomatillo
melon azuki bean garlic.

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 2

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 3

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 4

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 5

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

```
h2 {  
  column-span : all;  
}
```

I want

columns

Section 1

Veggies es bonus vobis,
proinde vos postulo
essum magis kohlrabi
welsh onion daikon
amaranth tatsoi tomatillo
melon azuki bean garlic.

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.

Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 2

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea

sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber

earthnut pea peanut soko
zucchini.

Section 3

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea

sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber

earthnut pea peanut soko
zucchini.

Section 4

Gumbo beet greens corn
soko endive gumbo

sprouts fava bean collard
greens dandelion okra

earthnut pea peanut soko
zucchini.

```
.columns {  
  column-width: 250px;  
}
```

I want columns

Section 1

Veggies es bonus vobis,
proinde vos postulo
essum magis kohlrabi
welsh onion daikon
amaranth tatsoi tomatillo
melon azuki bean garlic.

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 2

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 3

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 4

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Section 5

Gumbo beet greens corn
soko endive gumbo
gourd. Parsley shallot
courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra
wakame tomato.
Dandelion cucumber
earthnut pea peanut soko
zucchini.

Width is a minimum

Column boxes are flexible by default.

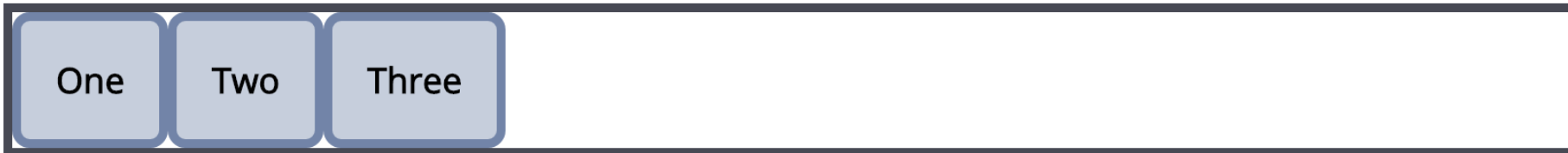
display

One

Two

Three

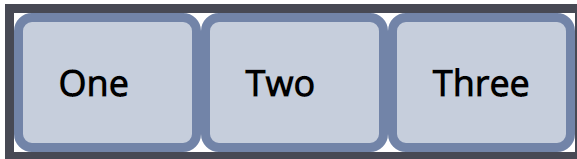
```
.flex {  
  display: flex;  
}
```



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



```
.grid {  
  display: inline-grid;  
}
```



Content following the grid.

Old & new values for display

Old Value	New Value(s)
block	block flow
flow-root	block flow-root
inline	inline flow
inline-block	inline flow-root
flex	block flex
inline-flex	inline flex
grid	block grid
inline-grid	inline grid

* Full list at <https://developer.mozilla.org/en-US/docs/Web/CSS/display>

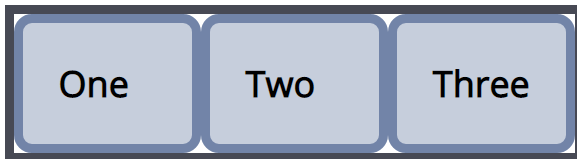
```
.flex {  
  display: block flex;  
}
```



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



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



Content following the grid.


```
.grid {  
  display: block flow;  
}
```



display: block flow-root

Creating a new Block Formatting Context (BFC)



Gumbo beet greens corn soko
endive gumbo gourd. Parsley
shallot courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra wakame
tomato. Dandelion cucumber
earthnut pea peanut soko
zucchini.

```
.container {  
  display: block flow-root;  
}
```



Gumbo beet greens corn soko
endive gumbo gourd. Parsley
shallot courgette tatsoi pea
sprouts fava bean collard
greens dandelion okra wakame
tomato. Dandelion cucumber
earthnut pea peanut soko
zucchini.

`display: inline-block`

```
<div>
  <span class="inline">
    I am an inline thing
  </span>
  I follow the inline thing.
  <br>I am below the inline thing.
</div>
```

```
.inline {  
  background-  
color: rgb(71, 73, 84);  
  color: #fff;  
  padding: 1em;  
}
```

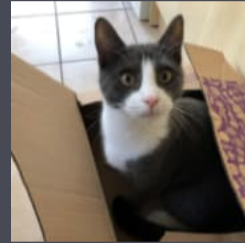
```
.inline img {  
  float: left;  
}
```



I am an inline thing I follow the inline thing.
I am below the inline thing.

```
.inline {  
  background-  
color: rgb(71, 73, 84);  
  color: #fff;  
  padding: 1em;  
  display: inline-block ;  
}
```

```
.inline img {  
  float: left;  
}
```



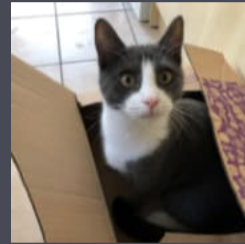
I am an inline thing

I follow the inline thing.

I am below the inline thing.


```
.inline {  
  background-  
color: rgb(71, 73, 84);  
  color: #fff;  
  padding: 1em;  
  display: inline flow-root;  
}
```

```
.inline img {  
  float: left;  
}
```



I am an inline thing

I follow the inline thing.

I am below the inline thing.

Two-value syntax of display

Implemented in Firefox 70

`display: flex`

```
.flex {  
  display: flex;  
}
```

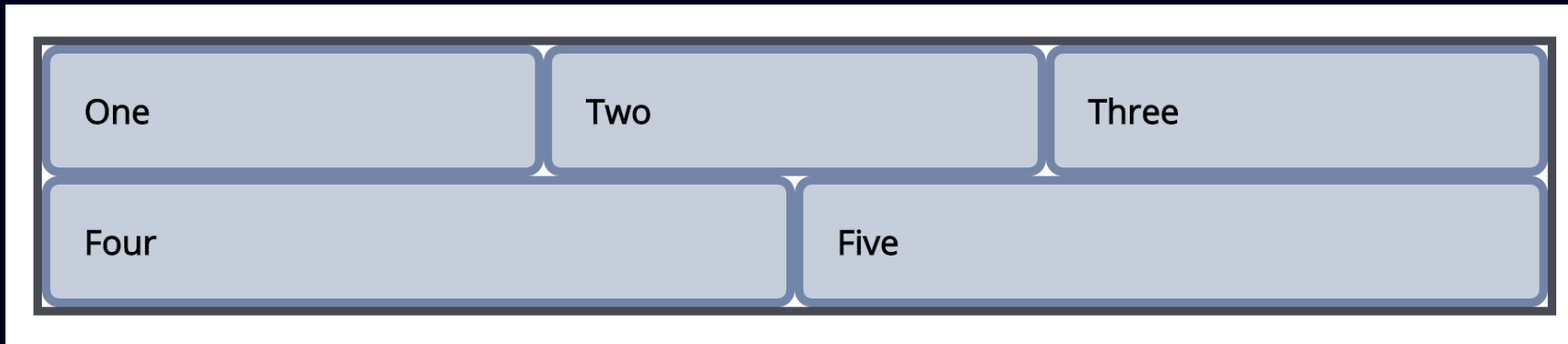
```
.flex > div {  
  flex: 1;  
}
```



A flex-basis of 0 means items
contribute no size.

```
.flex {  
  display: flex;  
  flex-wrap: wrap;  
}
```

```
.flex > div {  
  flex: 1 1 200px;  
}
```



```
.flex {  
  display: flex;  
  flex-wrap: wrap;  
}
```

```
.flex > div {  
  flex: 0 0 33.3333%;  
}
```



Flexbox is more than floats with alignment.

Which we discover when we stop trying to make it a grid.

`flex: auto`

auto

Lay the items out at max-content size,
then remove or add space in proportion.

```
.flex {  
  display: flex;  
  justify-content: space-between;  
}
```

Home

About us

A committee chose this nav item

Contact

Item One	2
Item Two	11
Item Three	4
Item Four	5

```
<ul class="list-group">
  <li>Item One
    <span class="badge">2</span>
  </li>
  <li>Item Two
    <span class="badge">11</span>
  </li>
  <li>Item Three
    <span class="badge">4</span>
  </li>
  <li>Item Four
    <span class="badge">5</span>
  </li>
</ul>
```

```
<li>Item One  
  <span class="badge">2</span>  
</li>
```

Item One 2

Item Two 11

Item Three 4

Item Four 5

```
.list-group li {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
<li>Item One  
  <span class="badge">2</span>  
</li>
```

Item One	2
Item Two	11
Item Three	4
Item Four	5

Home

About us

A committee chose this nav item

Contact

Home

About us

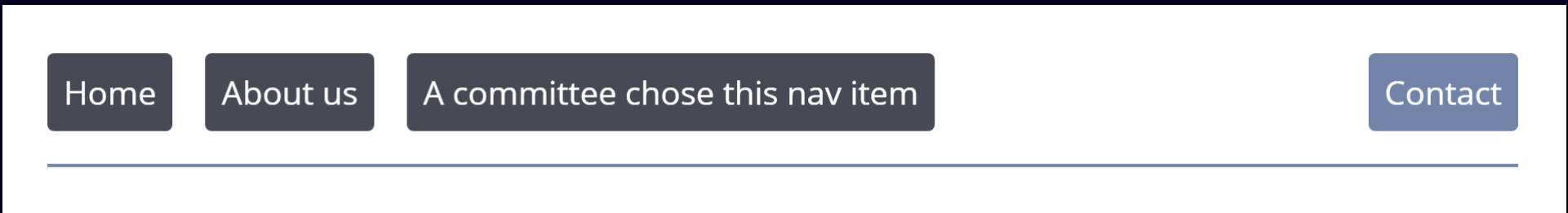
A committee chose this nav item

Contact

“Prior to alignment via justify-content and align-self, any positive free space is distributed to auto margins in that dimension.”

<https://www.w3.org/TR/css-flexbox-1/#auto-margins>

```
nav li:last-child {  
    margin-left: auto;  
}
```



“The specification describes a CSS box model optimized for user interface design.”

<https://www.w3.org/TR/css-flexbox-1/>

Writing Modes

```
writing-mode: horizontal-tb;
```

Inline Dimension



Block Dimension



Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify bunya nuts nori azuki bean chickweed potato bell pepper artichoke.

writing-mode: vertical-rl;

Block Dimension



Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi
welsh onion daikon amaranth tatsoi tomatillo melon azuki bean
garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot
courgette tatsoi pea sprouts fava bean collard greens dandelion
okra wakame tomato. Dandelion cucumber earthenut pea peanut
soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea
lettuce kohlrabi amaranth water spinach avocado daikon napa
cabbage asparagus winter purslane kale.

Celery potato scallion desert raisin horseradish spinach carrot
soko. Lotus root water spinach fennel kombu maize bamboo shoot
green bean swiss chard seakale pumpkin onion chickpea gram corn
pea. Brussels sprout coriander water chestnut gourd swiss chard
wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify
bunya nuts nori azuki bean chickweed potato bell pepper artichoke.



Inline Dimension

Block Start

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify bunya nuts nori azuki bean chickweed potato bell pepper artichoke.

Block End

Block Start

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify bunya nuts nori azuki bean chickweed potato bell pepper artichoke.

Block End

Inline Start

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber eartrnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify bunya nuts nori azuki bean chickweed potato bell pepper artichoke.

Inline End

Inline End

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber eartrnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

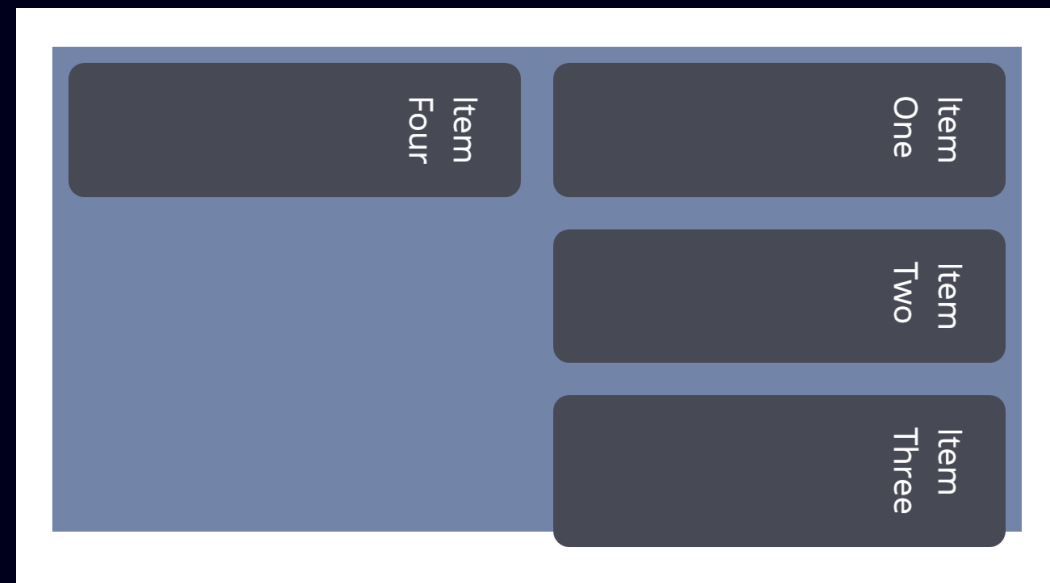
Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify bunya nuts nori azuki bean chickweed potato bell pepper artichoke.

Inline Start

Web layout was tied to physical dimensions.

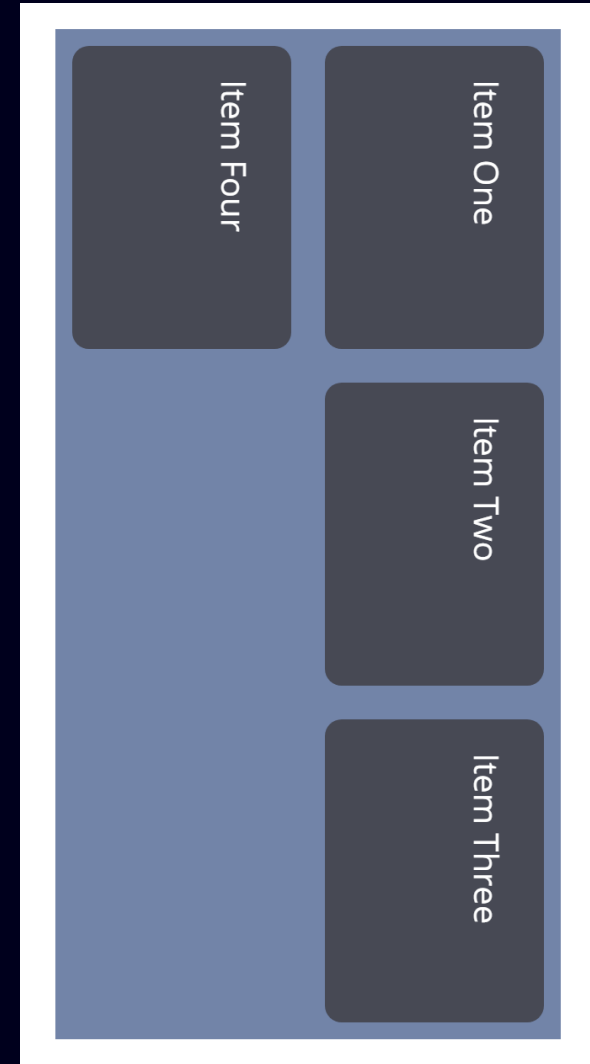
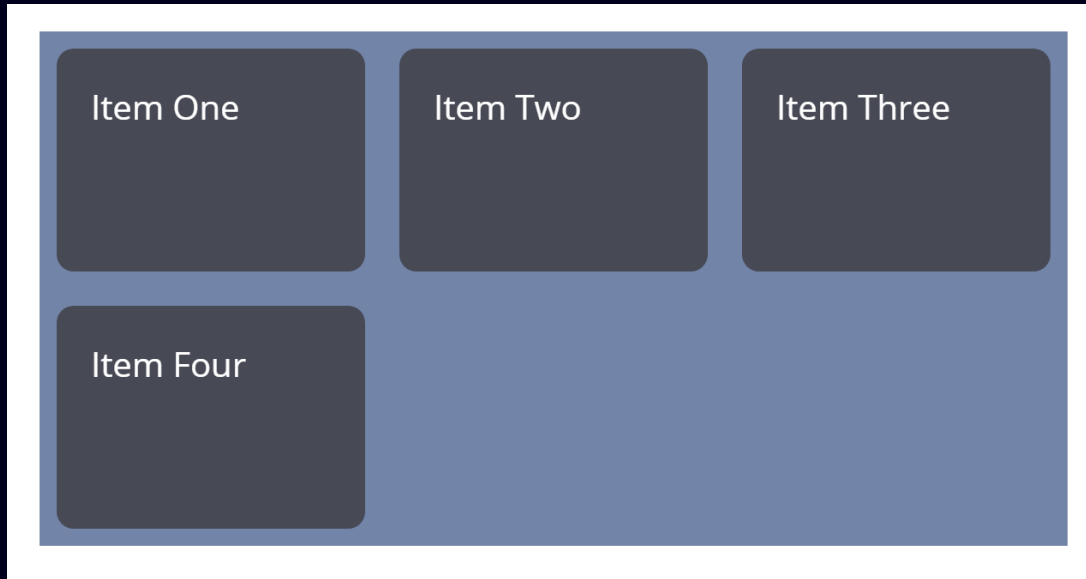
We think in top, right, bottom left, width and height.

```
.example {  
  width: 600px;  
  height: 300px;  
}
```



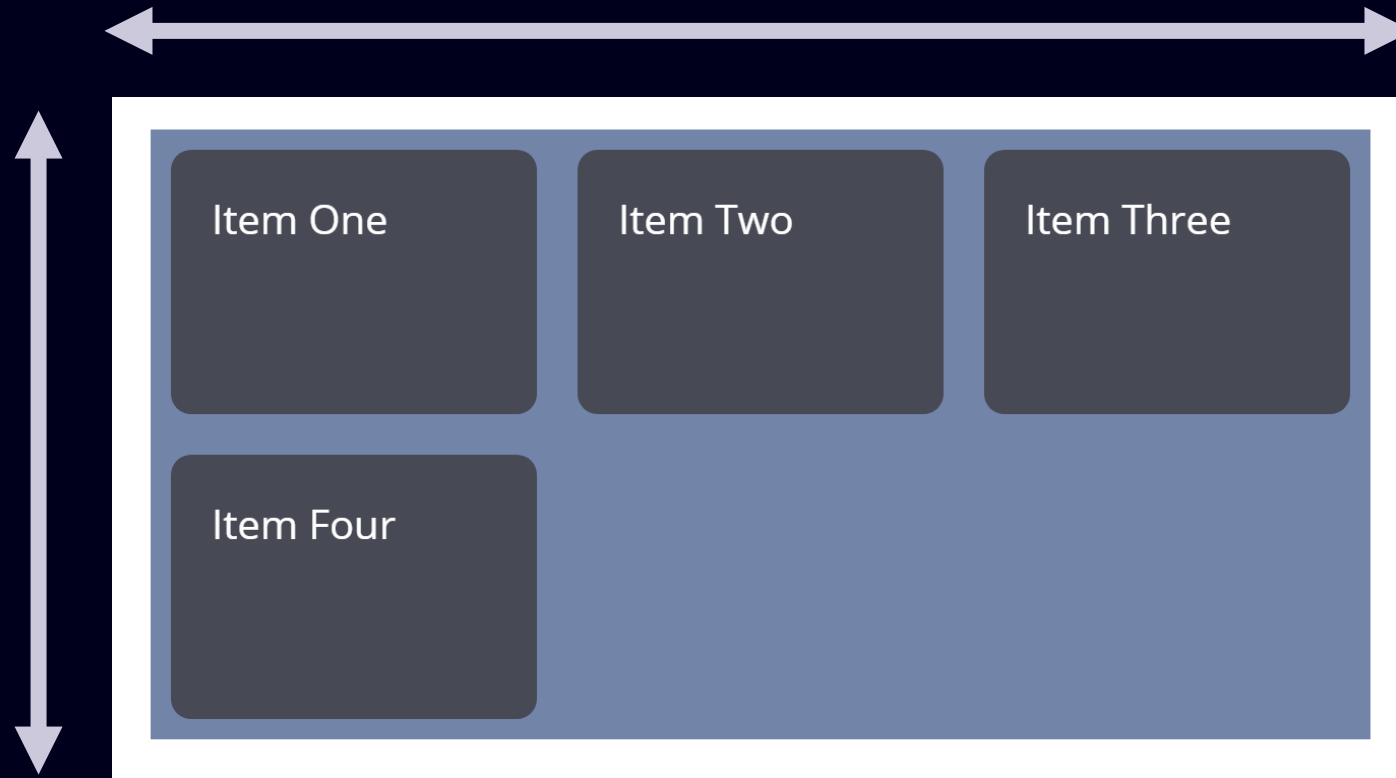
Logical Properties and Values

```
.example {  
  inline-size: 600px;  
  block-size: 300px;  
}
```

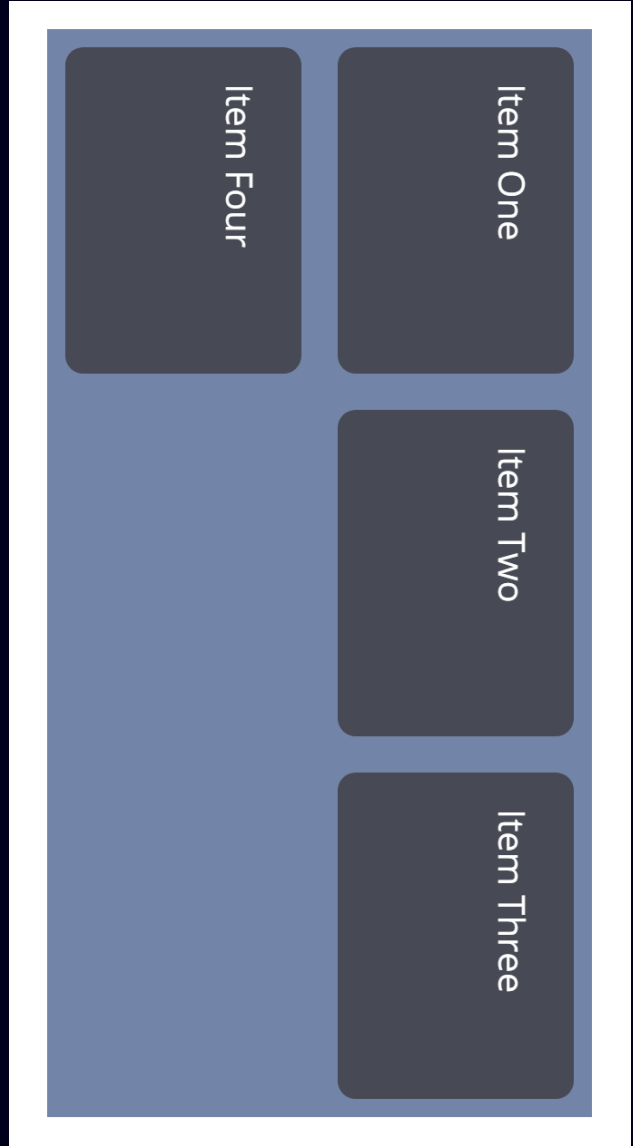


inline-size = width

block-size = height



block-size = width



inline-size = height



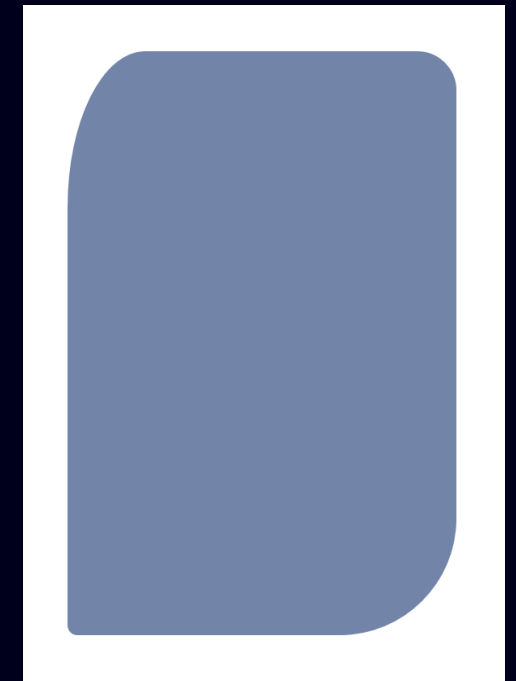
Physical v. Logical

```
.example {  
  padding-top: 10px;  
  padding-right: 2em;  
  margin-bottom: 2em;  
}
```

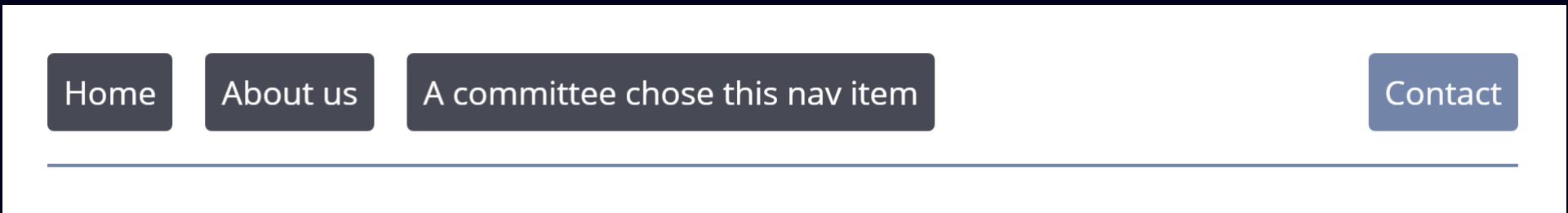
```
.example {  
  padding-block-start: 10px;  
  padding-inline-end: 2em;  
  margin-block-end: 2em;  
  margin-inline: 1em;  
}
```



```
.example {  
  border-start-start-radius: 20px;  
  border-start-end-radius: 3em;  
  border-end-start-radius: 2em 4em;  
  border-end-end-radius: 5px;  
}
```



```
nav li:last-child {  
    margin-left: auto;  
}
```



Naming things

A CSS Grid superpower



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vehicula vitae ligula sit amet maximus. Nunc auctor neque ipsum, ac porttitor elit lobortis ac. Vivamus ultrices sodales tellus et aliquam. Pellentesque porta sit amet nulla vitae luctus. Praesent quis risus id dolor venenatis condimentum.

```
.media {
  display: grid;
  grid-template-columns: fit-content(200px) 1fr;
  grid-template-areas:
    "image content";
  grid-gap: 20px;
}

.img {
  grid-area: image;
}

.content {
  grid-area: content;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vehicula vitae ligula sit amet maximus. Nunc auctor neque ipsum, ac porttitor elit lobortis ac. Vivamus ultrices sodales tellus et aliquam. Pellentesque porta sit amet nulla vitae luctus. Praesent quis risus id dolor venenatis condimentum.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vehicula vitae ligula sit amet maximus. Nunc auctor neque ipsum, ac porttitor elit lobortis ac. Vivamus ultrices sodales tellus et aliquam. Pellentesque porta sit amet nulla vitae luctus. Praesent quis risus id dolor venenatis condimentum.

```
.media-flip {  
  grid-template-columns: 1fr fit-content(200px);  
  grid-template-areas:  
    "content image";  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vehicula vitae ligula sit amet maximus. Nunc auctor neque ipsum, ac porttitor elit lobortis ac. Vivamus ultrices sodales tellus et aliquam. Pellentesque porta sit amet nulla vitae luctus. Praesent quis risus id dolor venenatis condimentum.



No 'right' way.

I like using grid for this. You might use flex. It's *all good*.

“Named Columns”

Not a thing. Kind of a thing.

In grid we name lines
and we name areas.

We don't name tracks.

```
.grid {  
  display: grid;  
  gap: 20px;  
  grid-template-columns:  
    [full-start] minmax(0,1fr)  
    [content-start start-half-start] minmax(0,2fr)  
    [start-half-end end-half-start] minmax(0,2fr)  
    [end-half-end content-end] minmax(0,1fr)  
    [full-end];  
}
```

```
h1 {  
  grid-column: content-start / content-end;  
}  
  
.one {  
  grid-row: 2;  
  grid-column: start-half-start / start-half-end;  
}
```

```
h1 {  
  grid-column: content;  
}
```

```
.one {  
  grid-row: 2;  
  grid-column: start-half;  
}
```

Naming columns

start-half

end-half

full

Named lines make a named area

Using the main name, with `-start` and `-end` removed.

Naming columns

start-half

end-half

full

Named areas can be used as named lines

The name resolves to the start or end edge of the area.

`grid-column: full`

`grid-column: full / full`

`grid-column: full-start / full-end`

Now add subgrid

Naming columns

start-half

start-half

end-half

end-half

```
.two {  
  grid-row: 3;  
  grid-column: full;  
  display: grid;  
  grid-template-columns: subgrid;  
}
```


Naming columns

start-half

start-half

end-half

end-half

Naming columns

start-half

start-half

end-half

end-half

Line names from the
parent grid are inherited
by the subgrid.

```
.two-a {  
  grid-column: start-half;  
}
```

```
.two-b {  
  grid-column: end-half;  
}
```

Naming columns

start-half

start-half

end-half

end-half

More CSS. Faster
implementations.

CSS WORKING GROUP WIKI

[Home](#) [Specs](#) [Ideas](#) [Testing](#) [About](#)

You are here: [CSS Working Group Wiki](#) » [Ideas and Resolutions](#) » [Incomplete List of Mistakes in the Design of CSS](#)

Incomplete List of Mistakes in the Design of CSS

That should be corrected if anyone invents a time machine. :P

- `white-space: nowrap` should be `white-space: no-wrap`
 - and line wrapping behavior should not have been added to `white-space`
- `vertical-align` should not apply to table cells. Instead the CSS3 alignment properties should exist in Level 1.
- `vertical-align: middle` should be `text-middle` or `x-middle` because it's not really in the middle, and such a name would better describes what it does.
- Percentage heights should be calculated against `fill-available` rather than being undefined in auto situations.
- Table layout should be sane.
- Box-sizing should be `border-box` by default.
- `background-size` with one value should duplicate its value, not default the second one to `auto`. Ditto `translate()`.
- `background-position` and `border-spacing` (all 2-axis properties) should take `*vertical*` first, to match with the 4-direction properties like `margin`.
- The 4-value shorthands like `margin` should go counter-clockwise (so that the inline-start value is before the block-start value).
- `z-index` should be called `z-order` or `depth` and should Just Work on all elements (like it does on flex items).
- `word-wrap/overflow-wrap` should not exist. Instead, `overflow-wrap` should be a keyword on 'white-space', like `nowrap` (`no-wrap`).
- The top and bottom margins of a single box should never have been allowed to collapse together automatically as this is the **root of all margin-collapsing evil**.
- Partial collapsing of margins instead of weird rules to handle min/max-heights?
- Tables (like other non-blocks, e.g. flex containers) should form pseudo-stacking contexts.
- The `currentcolor` keyword should have a dash, `current-color`. Likewise all other color multi-word keyword names.
- There should have been a predictable color naming system instead of arbitrary X11 names.
- `border-radius` should have been `corner-radius`.

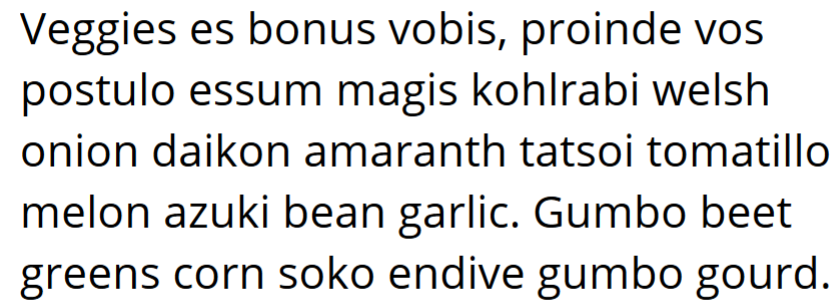
Interoperability

Things working the same way across browsers.

Standardizing

CSS Scrollbars

```
.scroller {  
  overflow-y: scroll;  
  scrollbar-width: thin;  
  scrollbar-color: rgb(255,255,255) rgb(158,120,153);  
}
```



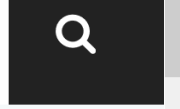
Veggies es bonus vobis, proinde vos
postulo essum magis kohlrabi welsh
onion daikon amaranth tatsoi tomatillo
melon azuki bean garlic. Gumbo beet
greens corn soko endive gumbo gourd.

Better defaults

Scroll anchoring

Ask why things work

Avoid always copying and pasting from a cheatsheet.



Subgrid

Jump to: [Introduction to subgrid](#) [Subgrid for columns](#) [Subgrid for rows](#) [A subgrid in both dimensions](#) [The gap properties and subgrid](#) [Named grid lines](#)

[Using subgrids](#) [Specification](#)

- [Web technology for developers](#) >
- [CSS: Cascading Style Sheets](#) >
- [CSS Grid Layout](#) > [Subgrid](#)

Related Topics

CSS

CSS Reference

CSS Grid Layout

Guides

- [Auto-placement in CSS Grid Layout](#)
- [Basic Concepts of grid layout](#)
- [Box alignment in CSS Grid Layout](#)
- [CSS Grid Layout and Accessibility](#)
- [CSS Grid Layout and Progressive Enhancement](#)

Level 2 of the CSS Grid Layout specification includes a `subgrid` value for `grid-template-columns` and `grid-template-rows`. This guide details what subgrid does, and gives some use cases and design patterns that are solved by the feature.

Important: This feature is currently available in Firefox Nightly to allow for testing. Download Firefox Nightly in order to try out the examples shown below.

Introduction to subgrid

When you add `display: grid` to a grid container, only the direct children become grid items and can then be placed on the grid that you have created. The children of these items display in normal flow.

You can "nest" grids by making a grid item a grid container. These grids however are independent of the parent grid and of each other, meaning that they do not take their track

We're working out how to use
all this new stuff together.

Thank you

@rachelandrew | <https://noti.st/rachelandrew/>