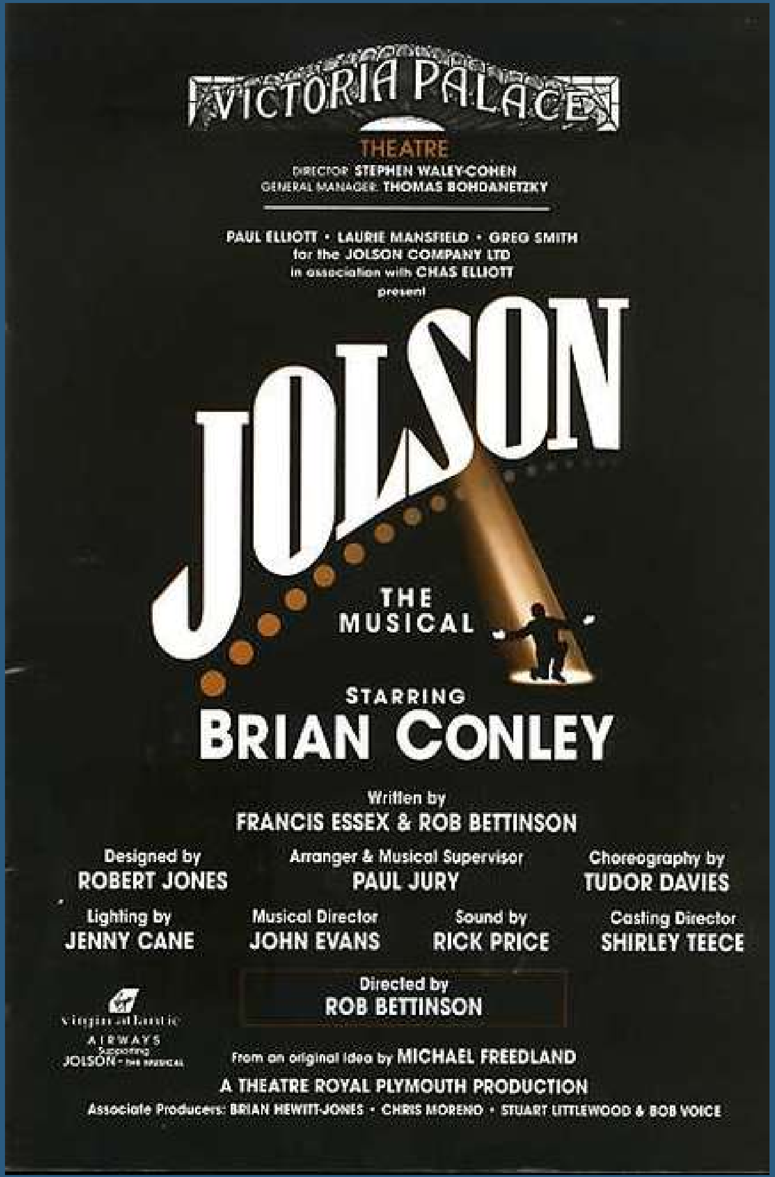
Learning to love CSS.

Rachel Andrew @ c’t webdev

Doing things on the web since 1996

Co-founder **Perch CMS** & **Notist**. Editor in Chief **Smashing Magazine**. Writer of many books. **CSS Working Group** Member representing **Fronteers**. Spec editor **Multicol** and Page Floats. **MDN** tech writer.

```
Amstrad 64K Microcomputer  (v1)
©1984 Amstrad Consumer Electronics plc
          and Locomotive Software Ltd.
BASIC 1.0
Ready
```



VICTORIA PALACE
THEATRE
DIRECTOR STEPHEN WALEY-COHEN
GENERAL MANAGER THOMAS BOHDANETZKY

PAUL ELLIOTT • LAURIE MANSFIELD • GREG SMITH
for the JOLSON COMPANY LTD
in association with CHAS ELLIOTT
present

JOLSON
THE MUSICAL

STARRING
BRIAN CONLEY

Written by
FRANCIS ESSEX & ROB BETTINSON

Designed by
ROBERT JONES

Arranger & Musical Supervisor
PAUL JURY

Choreography by
TUDOR DAVIES

Lighting by
JENNY CANE

Musical Director
JOHN EVANS

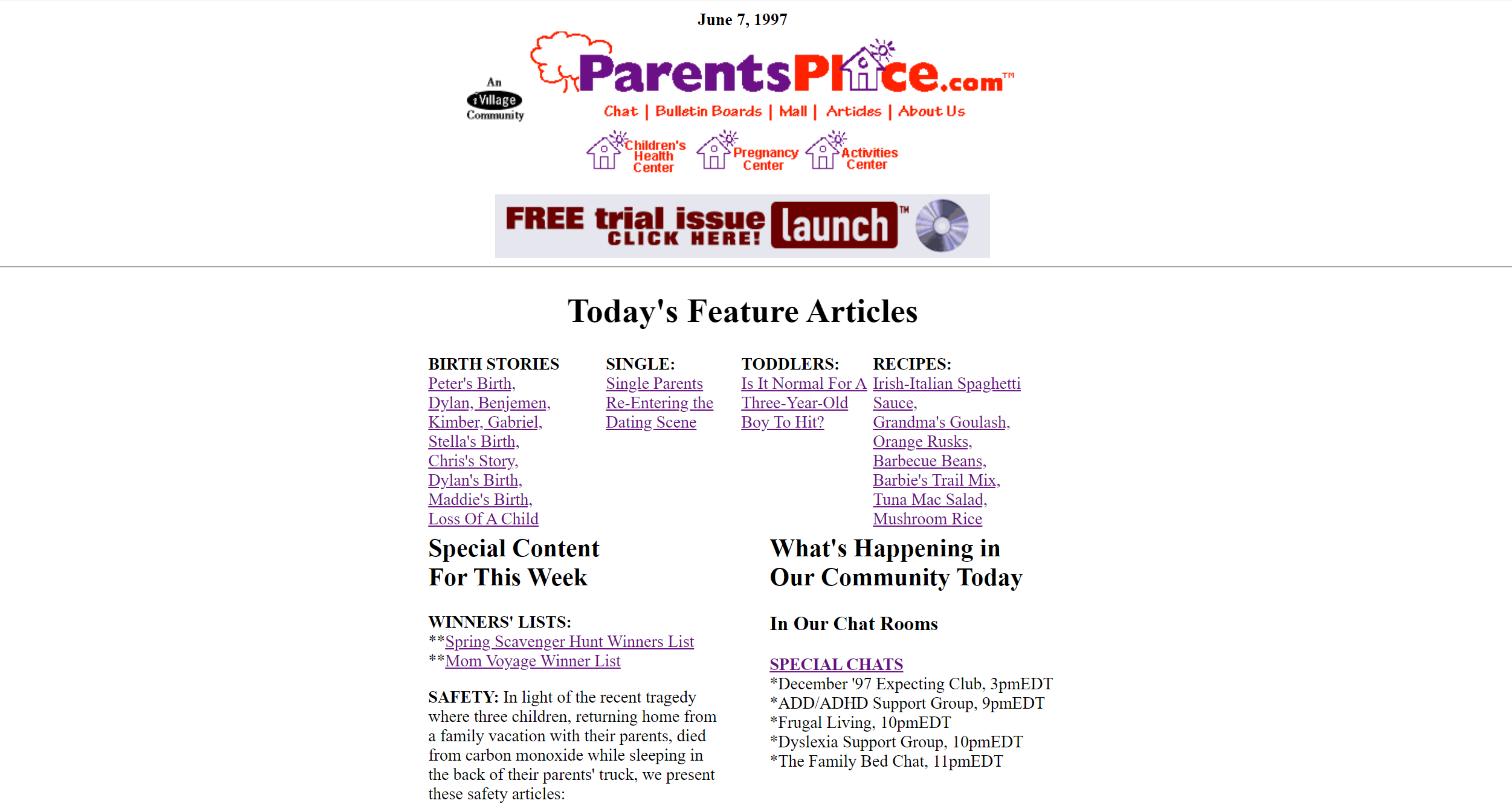
Sound by
RICK PRICE

Casting Director
SHIRLEY TEECE

Directed by
ROB BETTINSON

virgin atlantic
AIRWAYS
Supporting
JOLSON - THE MUSICAL

From an original idea by MICHAEL FREEDLAND
A THEATRE ROYAL PLYMOUTH PRODUCTION
Associate Producers: BRIAN HEWITT-JONES • CHRIS MORENO • STUART LITTLEWOOD & BOB VOICE



June 7, 1997

An iVillage Community

ParentsPlace.com™

Chat | Bulletin Boards | Mall | Articles | About Us

Children's Health Center

Pregnancy Center

Activities Center

Today's Feature Articles

BIRTH STORIES
Peter's Birth,
Dylan, Benjemen,
Kimber, Gabriel,
Stella's Birth,
Chris's Story,
Dylan's Birth,
Maddie's Birth,
Loss Of A Child

SINGLE:
Single Parents Re-Entering the Dating Scene

TODDLERS:
Is It Normal For A Three-Year-Old Boy To Hit?

RECIPES:
Irish-Italian Spaghetti Sauce,
Grandma's Goulash,
Orange Rusks,
Barbecue Beans,
Barbie's Trail Mix,
Tuna Mac Salad,
Mushroom Rice

Special Content For This Week

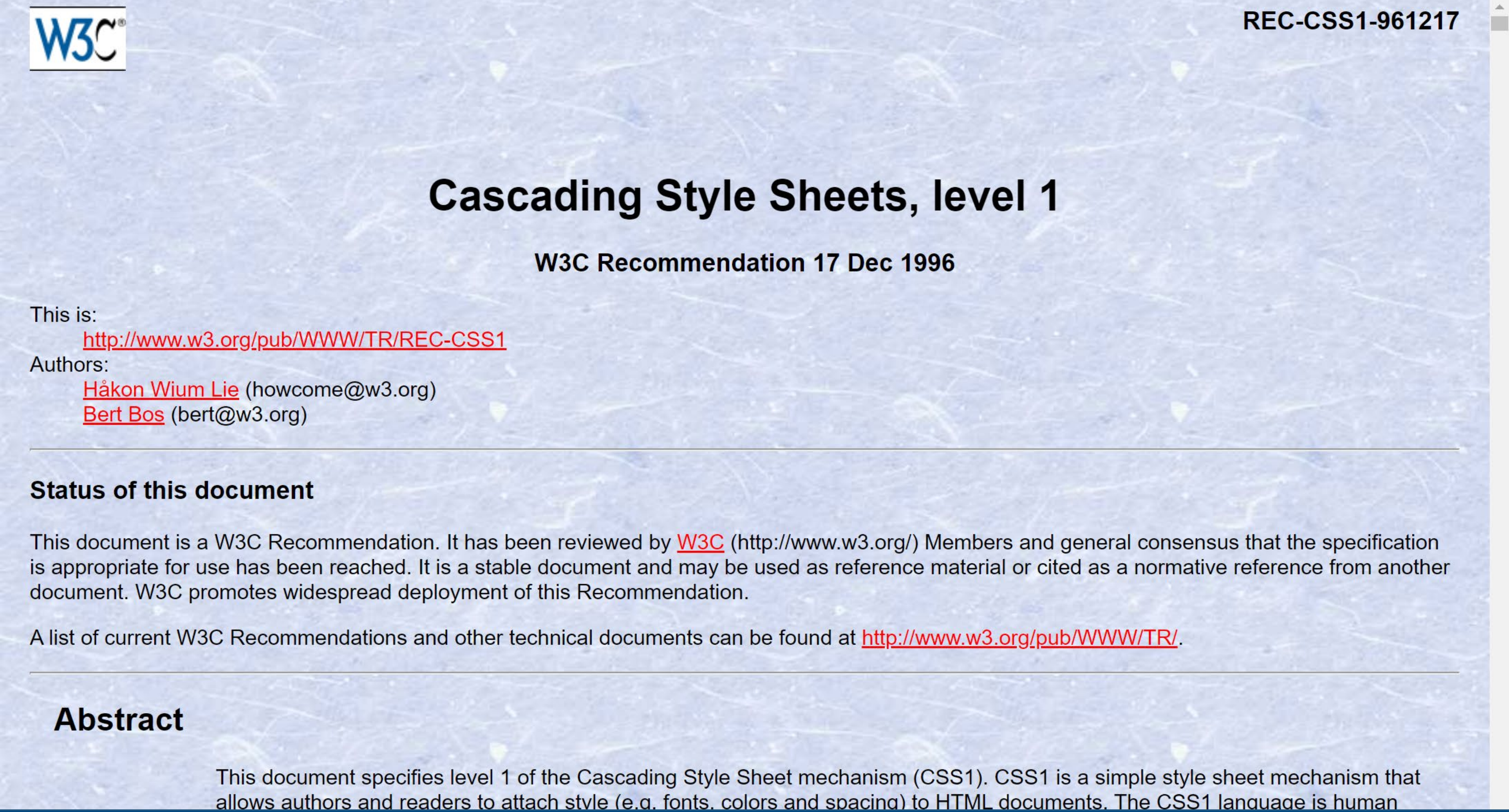
WINNERS' LISTS:
**Spring Scavenger Hunt Winners List
**Mom Voyage Winner List

SAFETY: In light of the recent tragedy where three children, returning home from a family vacation with their parents, died from carbon monoxide while sleeping in the back of their parents' truck, we present these safety articles:

What's Happening in Our Community Today

In Our Chat Rooms

SPECIAL CHATS
*December '97 Expecting Club, 3pmEDT
*ADD/ADHD Support Group, 9pmEDT
*Frugal Living, 10pmEDT
*Dyslexia Support Group, 10pmEDT
*The Family Bed Chat, 11pmEDT



REC-CSS1-961217

Cascading Style Sheets, level 1

W3C Recommendation 17 Dec 1996

This is:
    http://www.w3.org/pub/WWW/TR/REC-CSS1
Authors:
    Håkon Wium Lie (howcome@w3.org)
    Bert Bos (bert@w3.org)

Status of this document

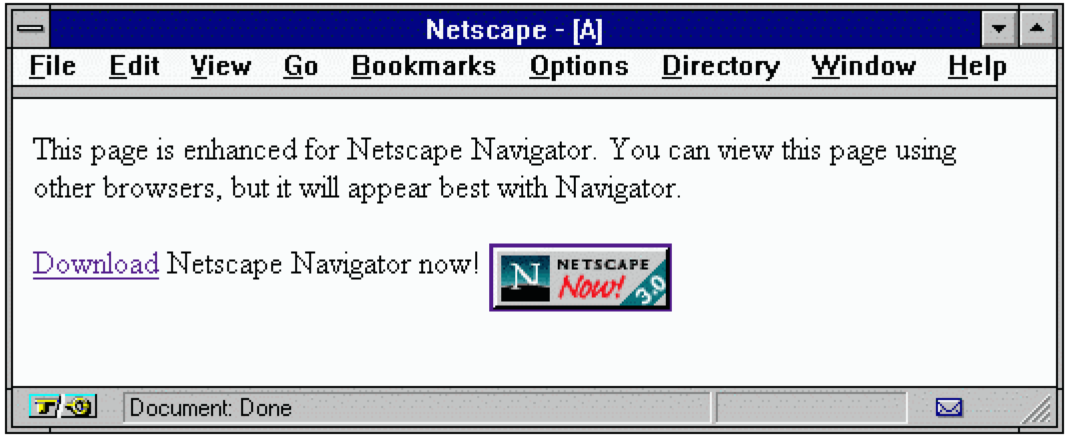
This document is a W3C Recommendation. It has been reviewed by W3C (http://www.w3.org/) Members and general consensus that the specification is appropriate for use has been reached. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C promotes widespread deployment of this Recommendation.

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/pub/WWW/TR/.

Abstract

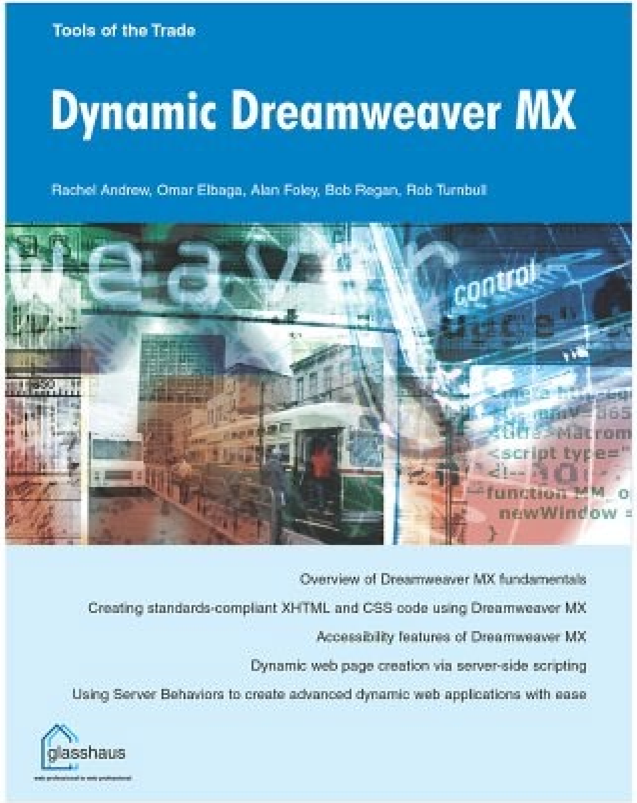
This document specifies level 1 of the Cascading Style Sheet mechanism (CSS1). CSS1 is a simple style sheet mechanism that allows authors and readers to attach style (e.g. fonts, colors and spacing) to HTML documents. The CSS1 language is human

this site best viewed in
Microsoft
Internet
Explorer 4.0



This page is enhanced for Netscape Navigator. You can view this page using other browsers, but it will appear best with Navigator.

Download Netscape Navigator now!



Tools of the Trade

Dynamic Dreamweaver MX

Rachel Andrew, Omar Elbaga, Alan Foley, Bob Regan, Rob Turnbull

Overview of Dreamweaver MX fundamentals

Creating standards-compliant XHTML and CSS code using Dreamweaver MX

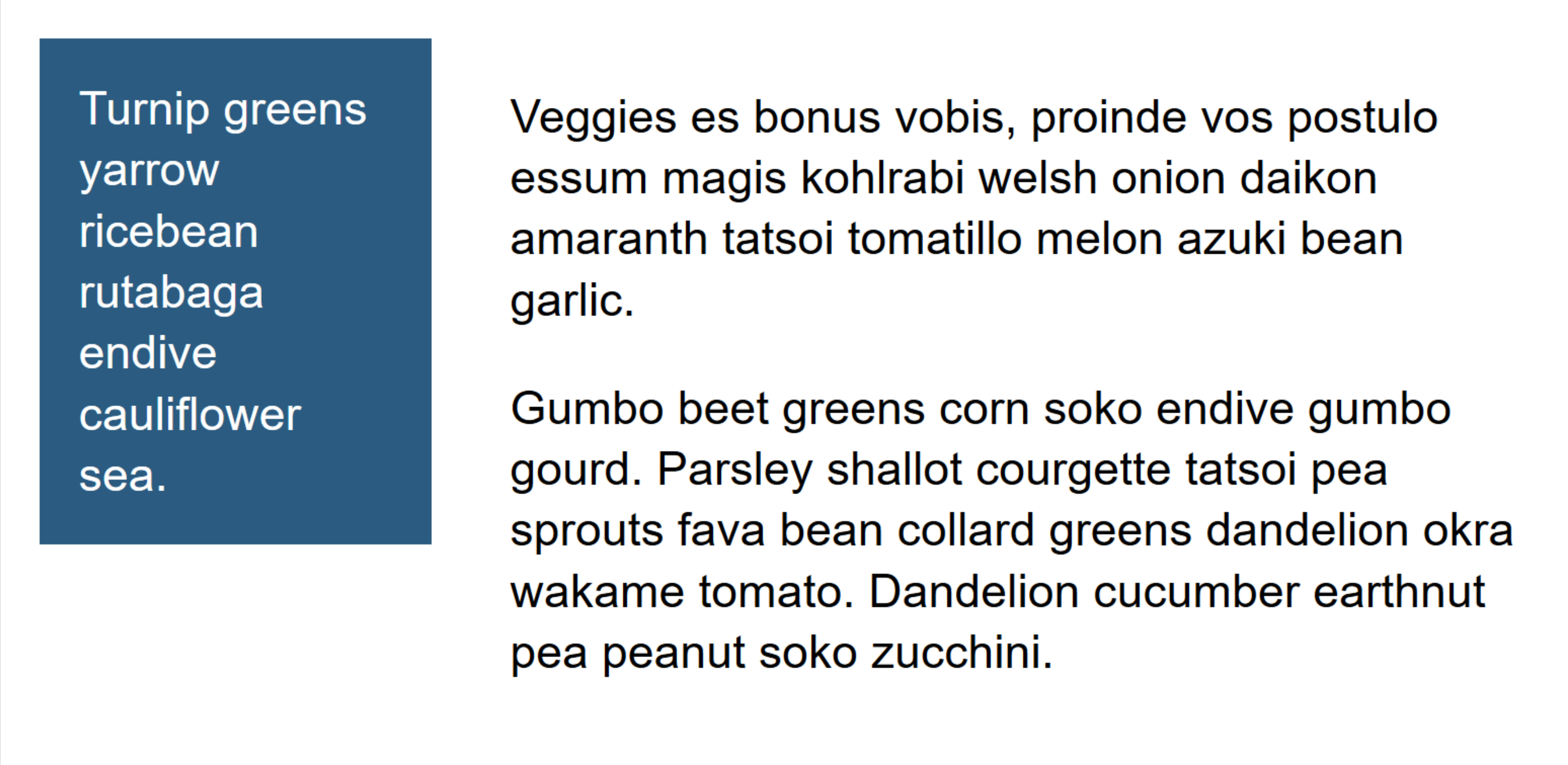
Accessibility features of Dreamweaver MX

Dynamic web page creation via server-side scripting

Using Server Behaviors to create advanced dynamic web applications with ease

glasshaus

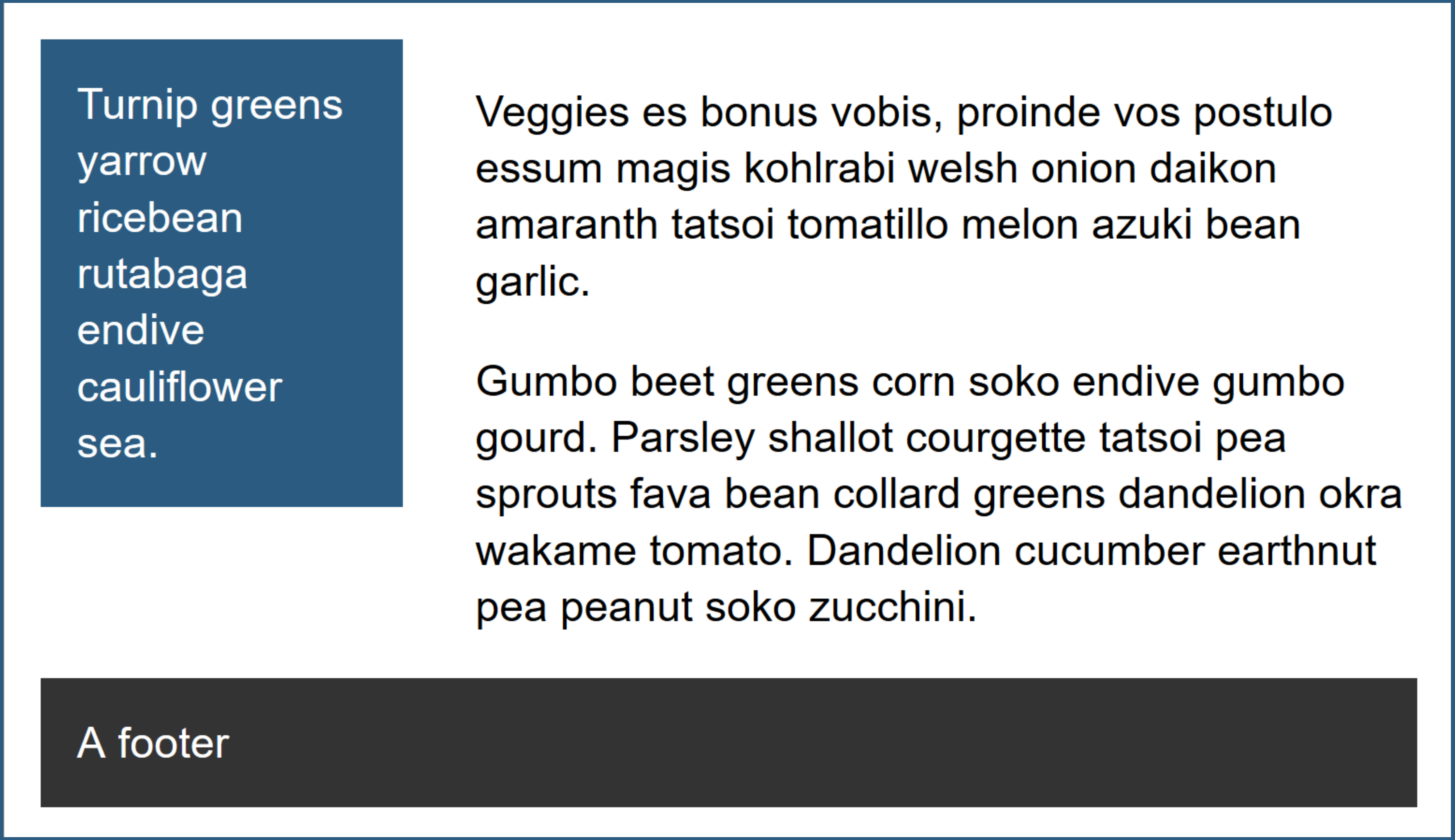
From <table> to CSS



Turnip greens yarrow ricebean rutabaga endive cauliflower sea.

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.

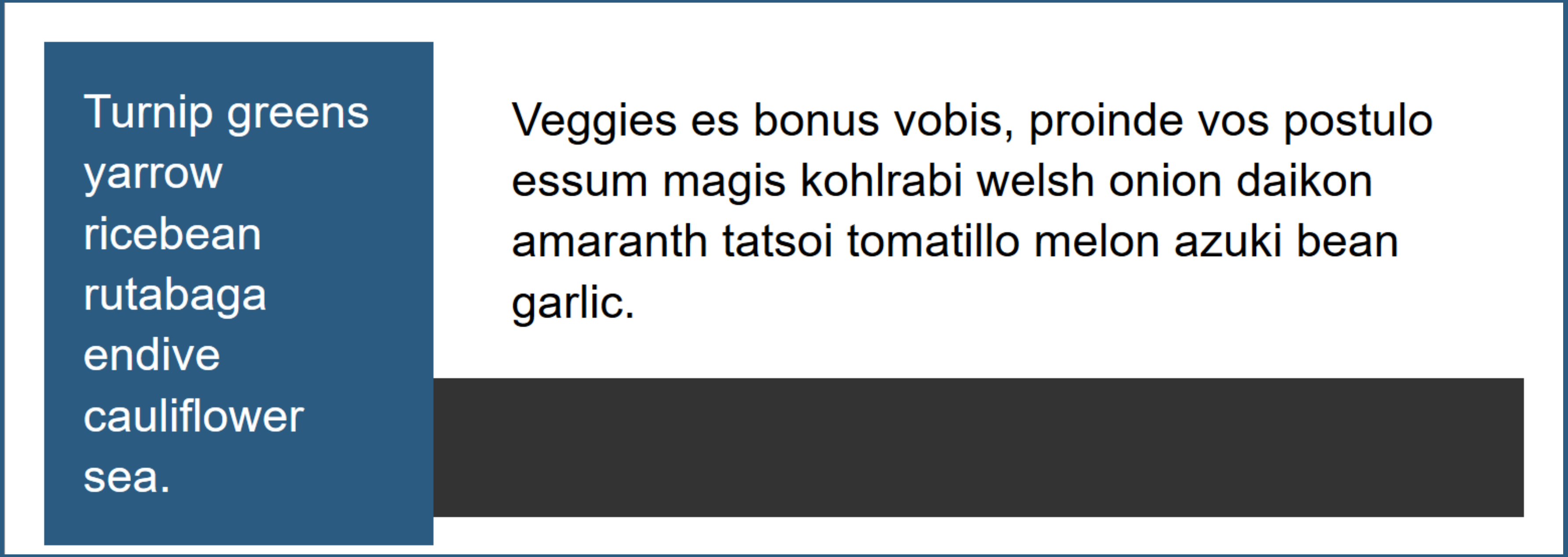


Turnip greens yarrow ricebean rutabaga endive cauliflower sea.

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.

A footer



Turnip greens yarrow ricebean rutabaga endive cauliflower sea.

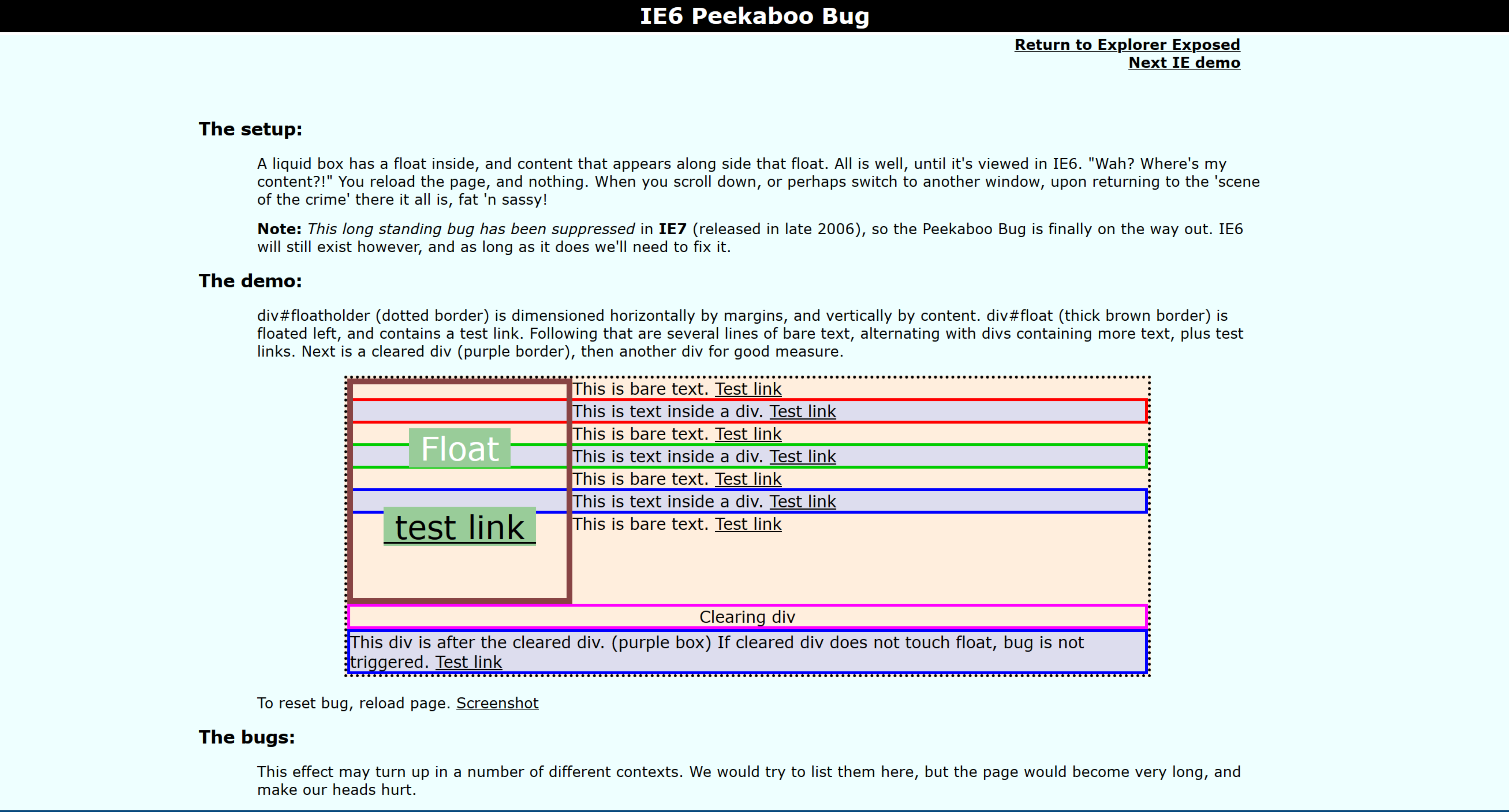
Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

The ‘Netscape Resize Fix’

Reloaded the browser window on resize, otherwise all positioned elements stacked up top left.

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

https://alistapart.com/article/journey/



IE6 Peekaboo Bug

Return to Explorer Exposed
Next IE demo

The setup:

A liquid box has a float inside, and content that appears along side that float. All is well, until it's viewed in IE6. "Wah? Where's my content?!" You reload the page, and nothing. When you scroll down, or perhaps switch to another window, upon returning to the 'scene of the crime' there it all is, fat 'n sassy!

Note: *This long standing bug has been suppressed* in **IE7** (released in late 2006), so the Peekaboo Bug is finally on the way out. IE6 will still exist however, and as long as it does we'll need to fix it.

The demo:

div#floatholder (dotted border) is dimensioned horizontally by margins, and vertically by content. div#float (thick brown border) is floated left, and contains a test link. Following that are several lines of bare text, alternating with divs containing more text, plus test links. Next is a cleared div (purple border), then another div for good measure.

This is bare text. Test link
This is text inside a div. Test link
This is bare text. Test link
This is text inside a div. Test link
This is bare text. Test link
This is text inside a div. Test link
This is bare text. Test link

Clearing div

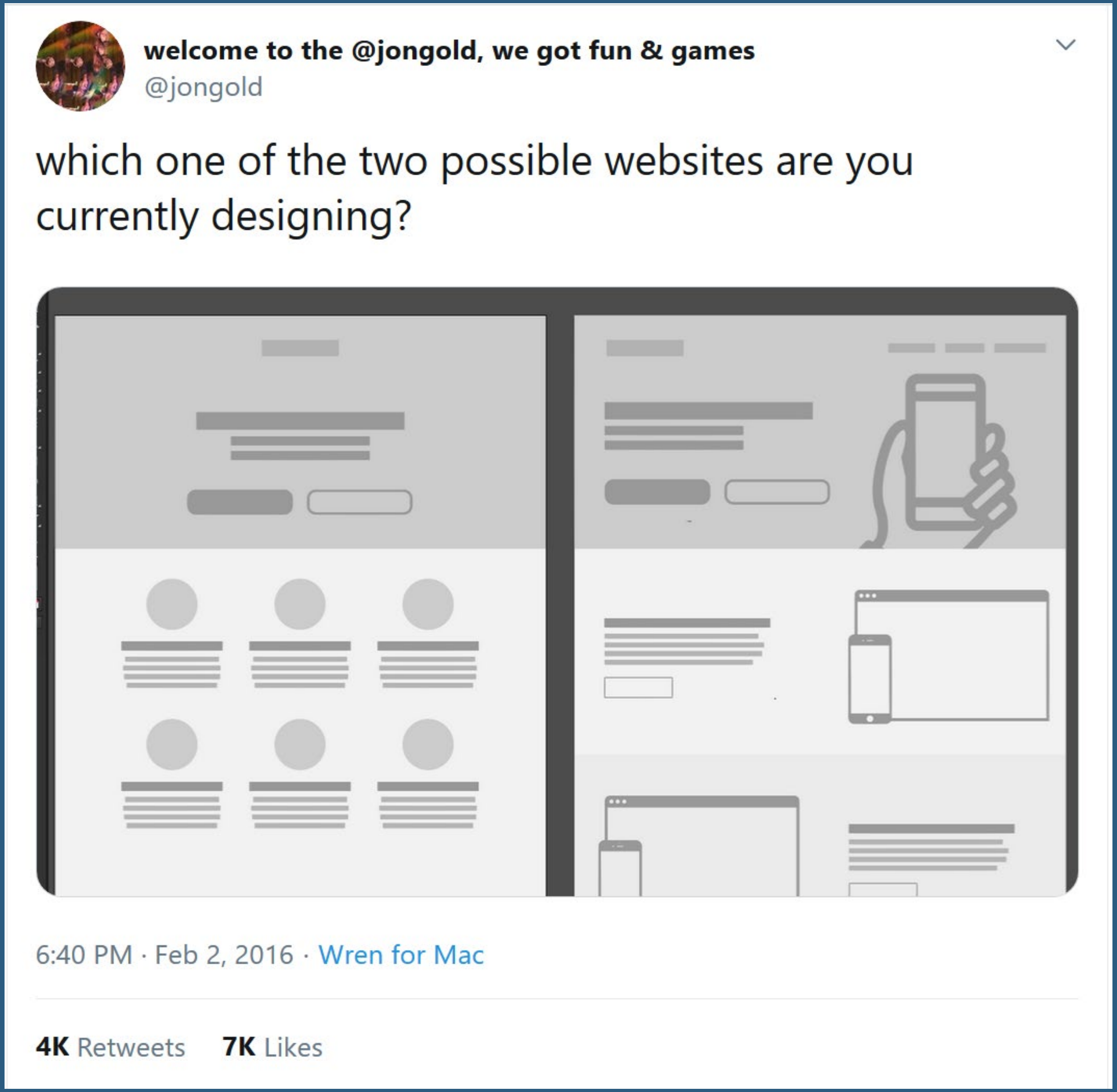
This div is after the cleared div. (purple box) If cleared div does not touch float, bug is not triggered. Test link

To reset bug, reload page. Screenshot

The bugs:

This effect may turn up in a number of different contexts. We would try to list them here, but the page would become very long, and make our heads hurt.

The job of a web developer was as browser bug wrangler.



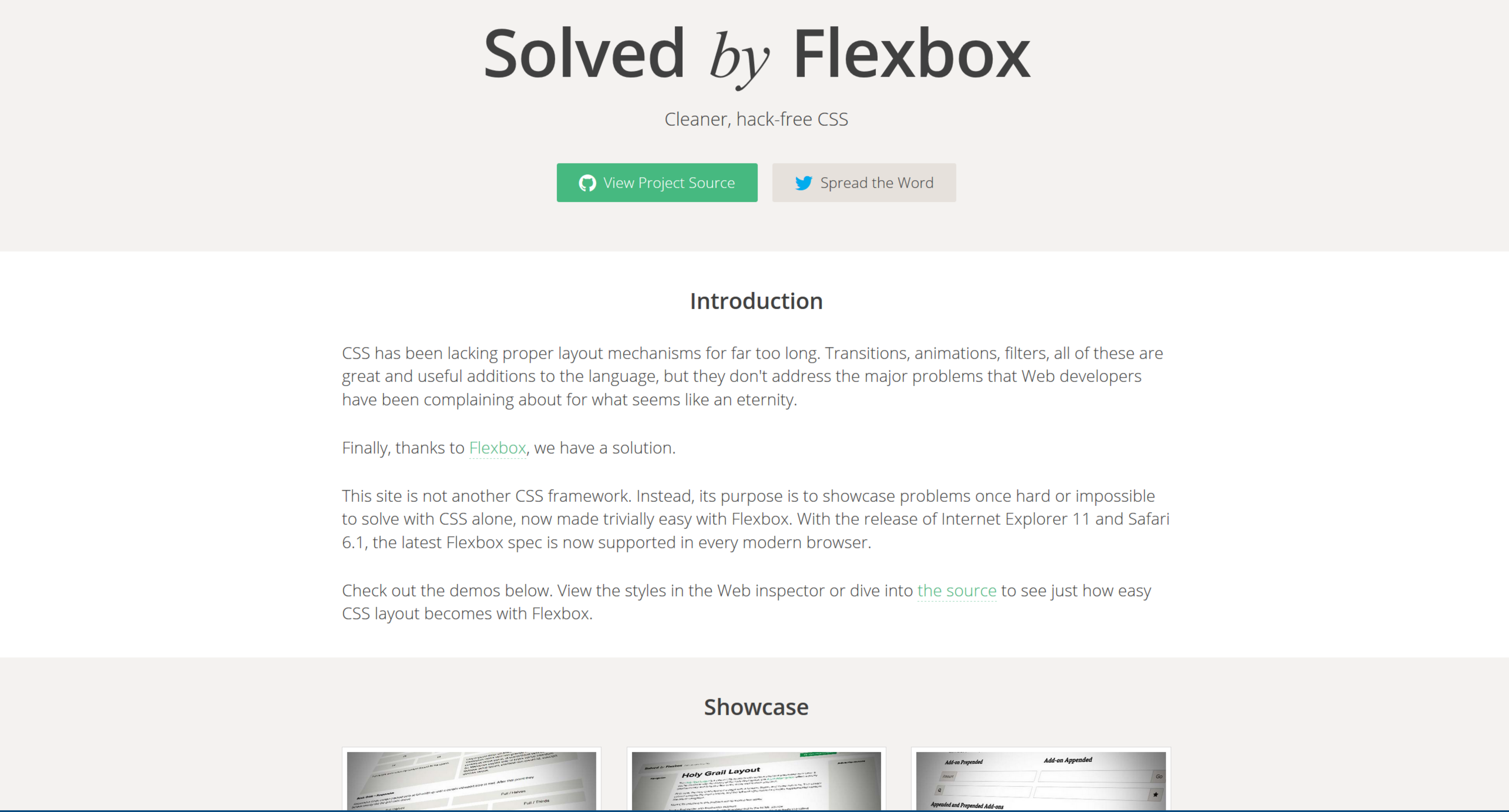
welcome to the @jongold, we got fun & games
@jongold

which one of the two possible websites are you currently designing?

6:40 PM · Feb 2, 2016 · Wren for Mac

4K Retweets **7K** Likes

Frameworks helped busy teams deal with fragile CSS layout methods.



Solved *by* Flexbox

Cleaner, hack-free CSS

Introduction

CSS has been lacking proper layout mechanisms for far too long. Transitions, animations, filters, all of these are great and useful additions to the language, but they don't address the major problems that Web developers have been complaining about for what seems like an eternity.

Finally, thanks to Flexbox, we have a solution.

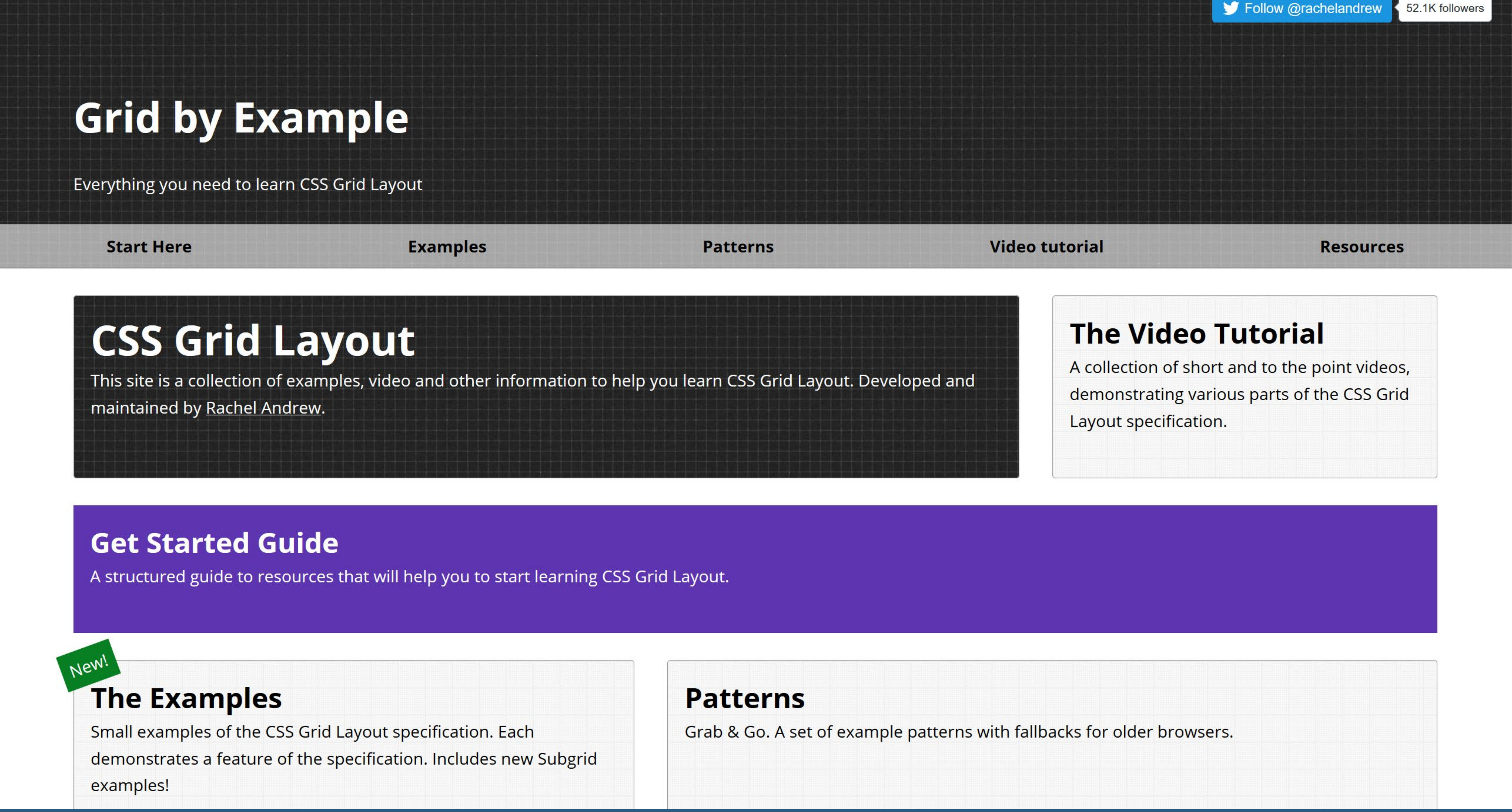
This site is not another CSS framework. Instead, its purpose is to showcase problems once hard or impossible to solve with CSS alone, now made trivially easy with Flexbox. With the release of Internet Explorer 11 and Safari 6.1, the latest Flexbox spec is now supported in every modern browser.

Check out the demos below. View the styles in the Web inspector or dive into the source to see just how easy CSS layout becomes with Flexbox.

Showcase

The Grid Layout spec

Initial implementation in **Internet Explorer 10**.



Grid by Example

Everything you need to learn CSS Grid Layout

Start Here | Examples | Patterns | Video tutorial | Resources

CSS Grid Layout

This site is a collection of examples, video and other information to help you learn CSS Grid Layout. Developed and maintained by Rachel Andrew.

The Video Tutorial

A collection of short and to the point videos, demonstrating various parts of the CSS Grid Layout specification.

Get Started Guide

A structured guide to resources that will help you to start learning CSS Grid Layout.

New!

The Examples

Small examples of the CSS Grid Layout specification. Each demonstrates a feature of the specification. Includes new Subgrid examples!

Patterns

Grab & Go. A set of example patterns with fallbacks for older browsers.

A real layout system at the heart of CSS

It’s more than just **grid** and **flex**.

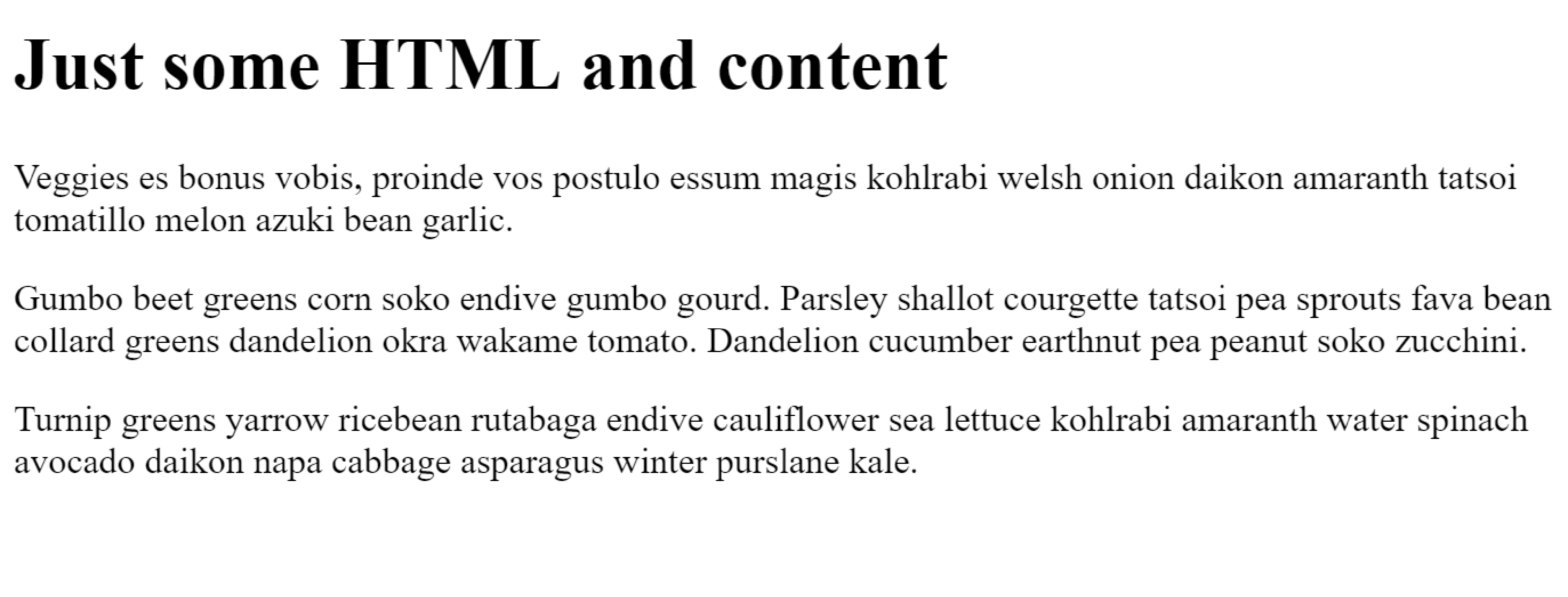
Talking about CSS as a layout system

- Flow Layout
- Changing the value of display
- Out of flow elements
- Block Formatting Contexts
- Writing Modes
- Logical, flow-relative properties and values
- Alignment
- Sizing
- Media & Feature Queries

Understanding display

Normal Flow

Block and Inline Layout



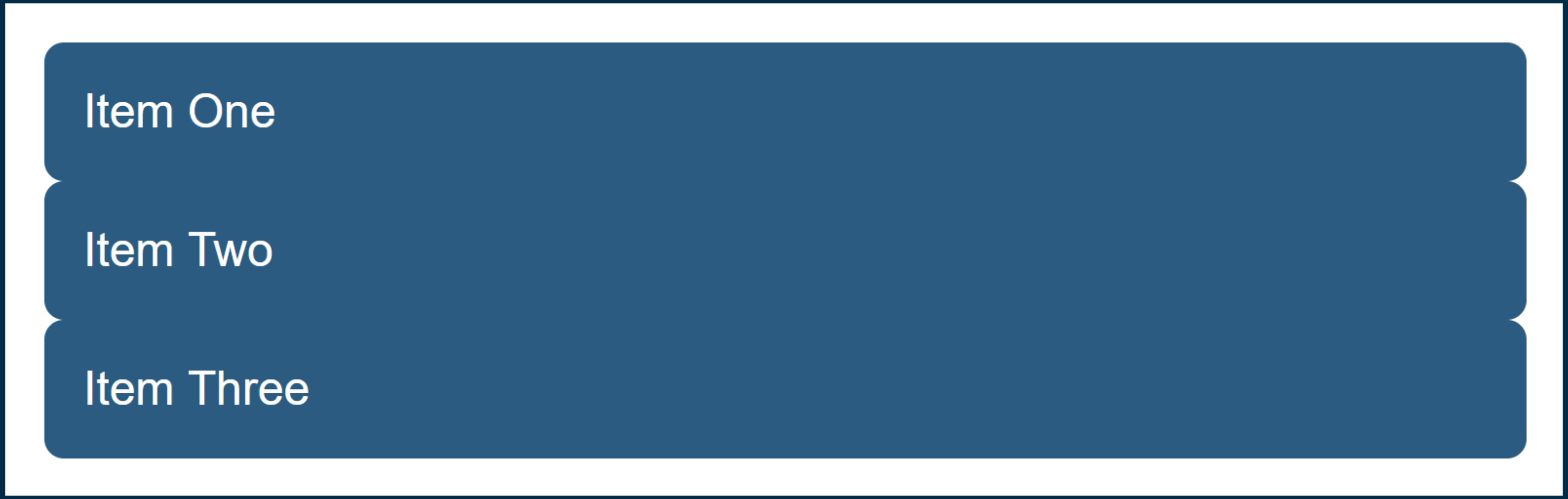
Just some HTML and content

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.

CSS is doing work for us, before we write any CSS.

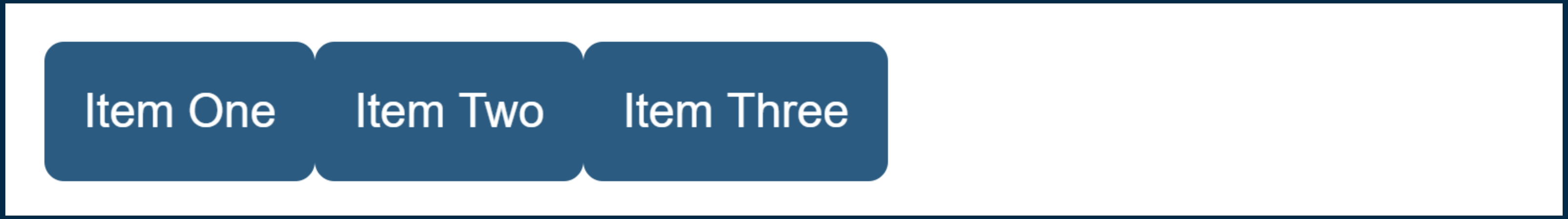


Item One

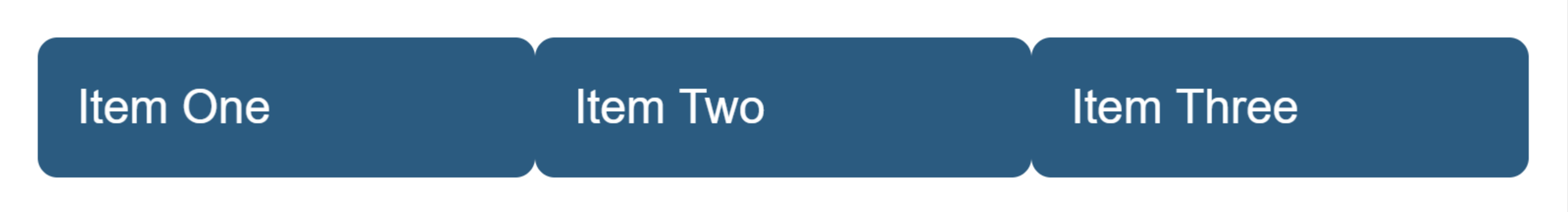
Item Two

Item Three

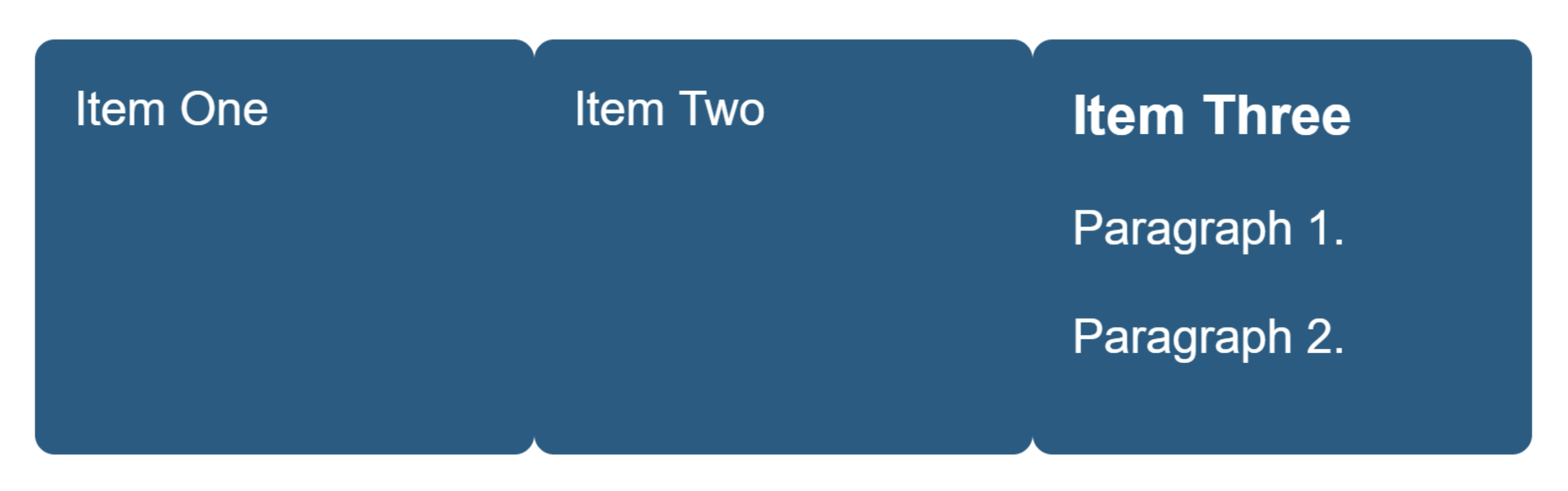
```
.example {
  display: flex;
}
```



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



Changing the value of display changes that element and its direct children.



Item One

Item Two

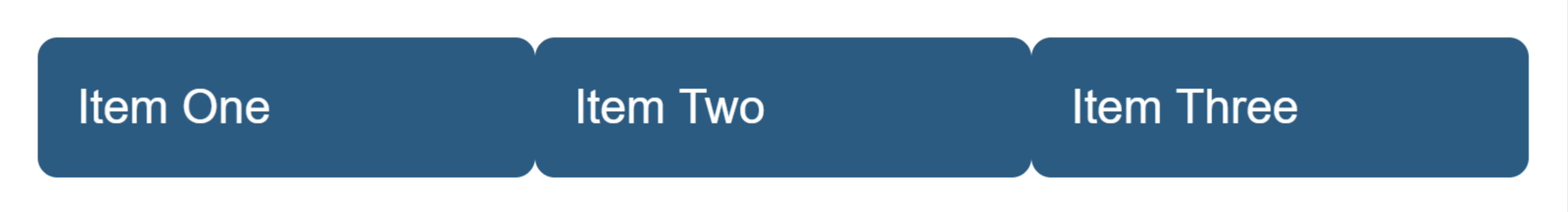
Item Three

Paragraph 1.

Paragraph 2.

The two values of display

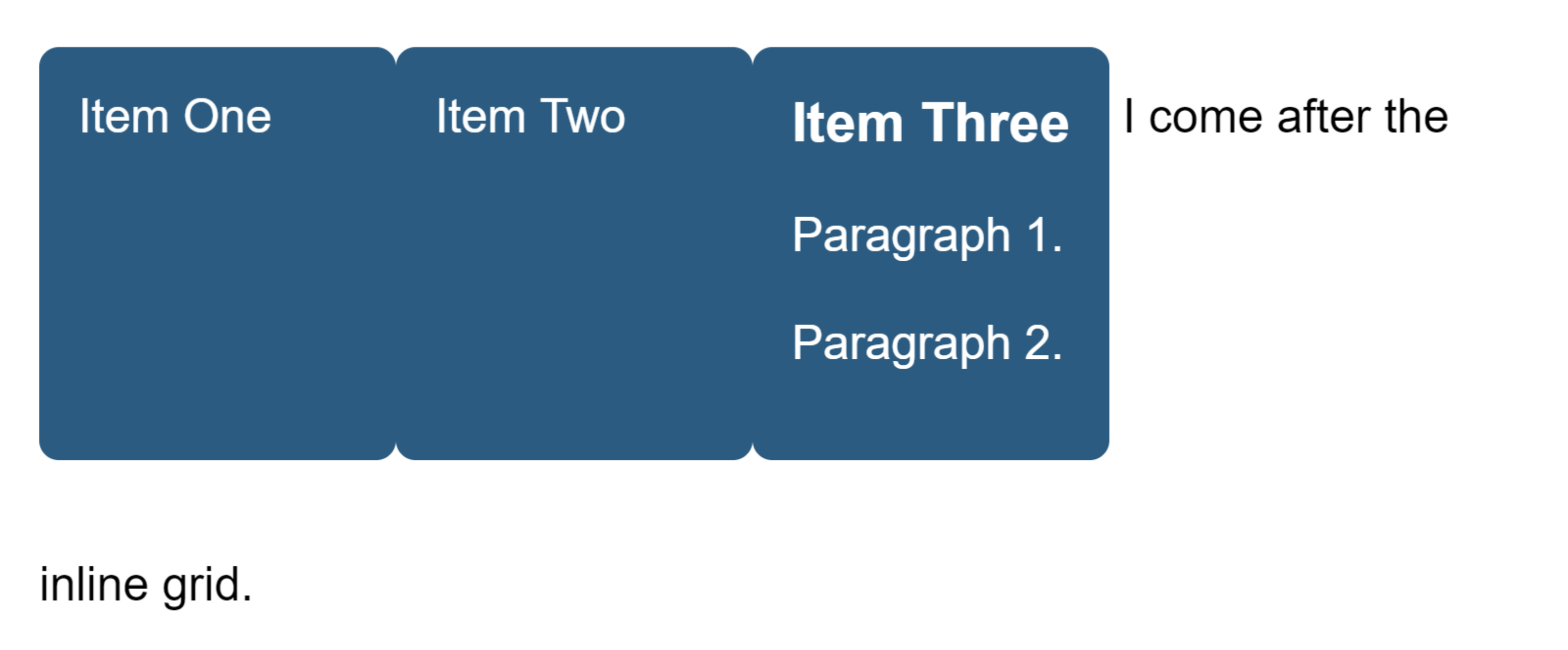
```
.example {
  display: block grid;
  grid-template-columns: 1fr 1fr 1fr;
}
```


Item One

Item Two

Item Three

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



Item One

Item Two

Item Three

Paragraph 1.

Paragraph 2.

I come after the inline grid.

The outer display type

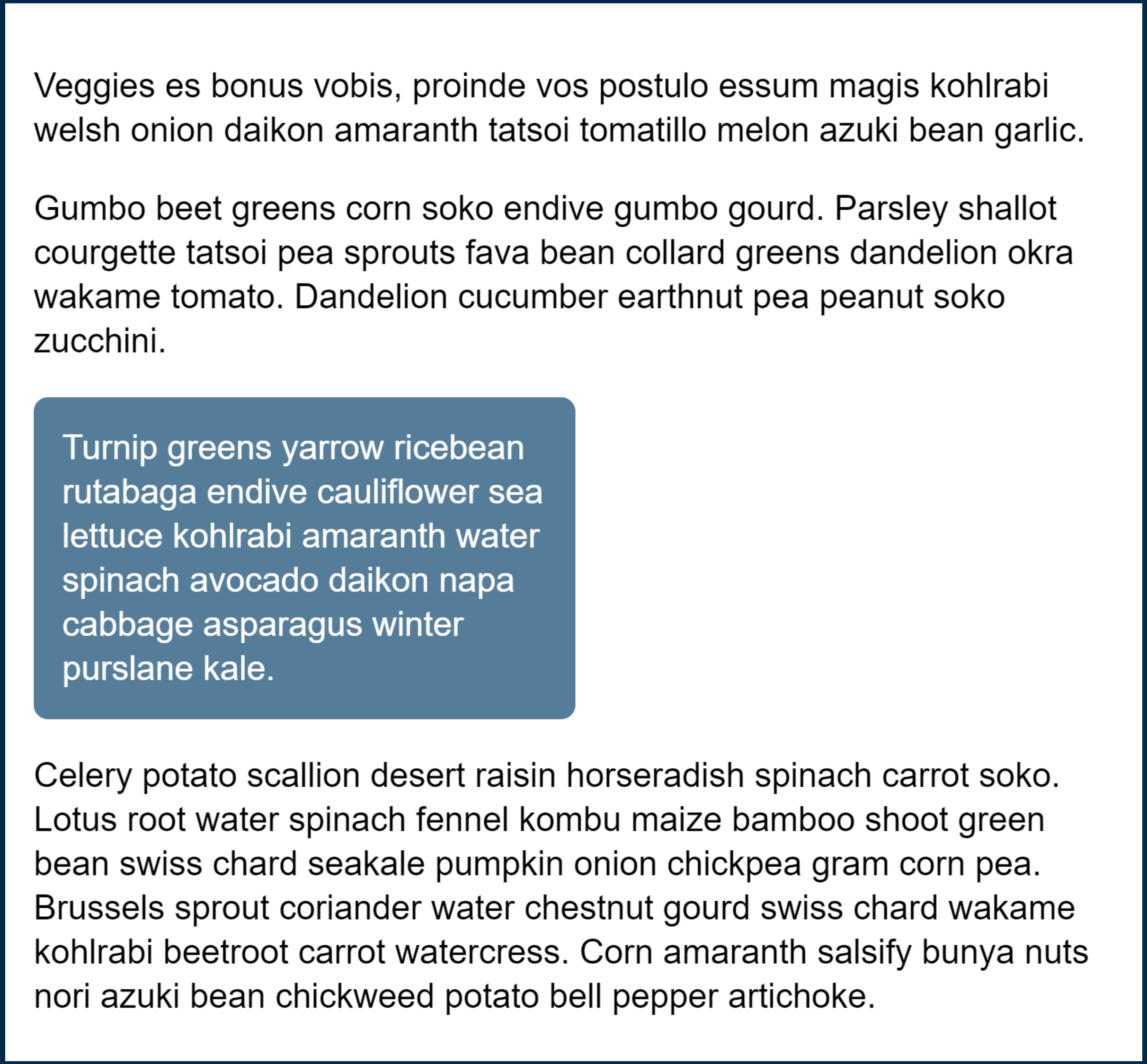
How the box behaves in the layout - block or inline

The inner display type

The formatting context of the direct children – grid, flex etc.

Busting out of flow

position



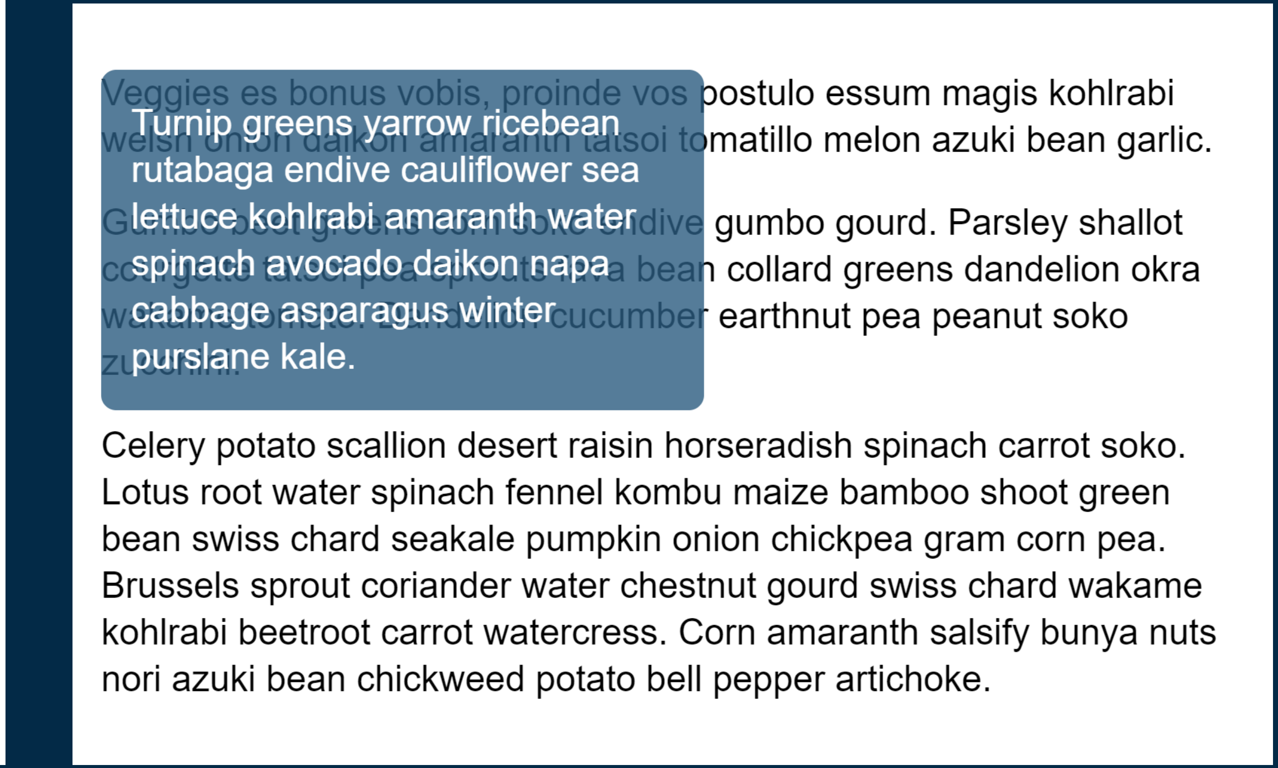
Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

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.

```css
.position {
  position: absolute;
}
```



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

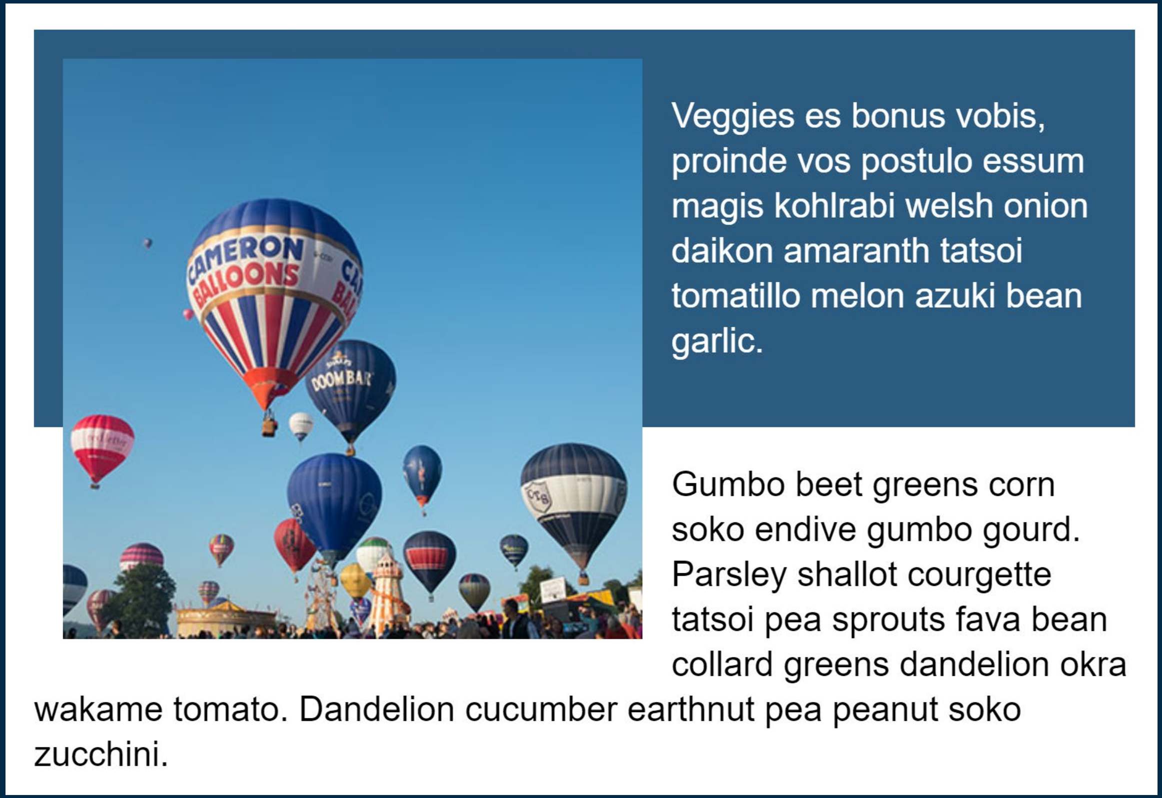
Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

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.

float

```css
.box {
  background-color: rgb(43,91,128);
}
```



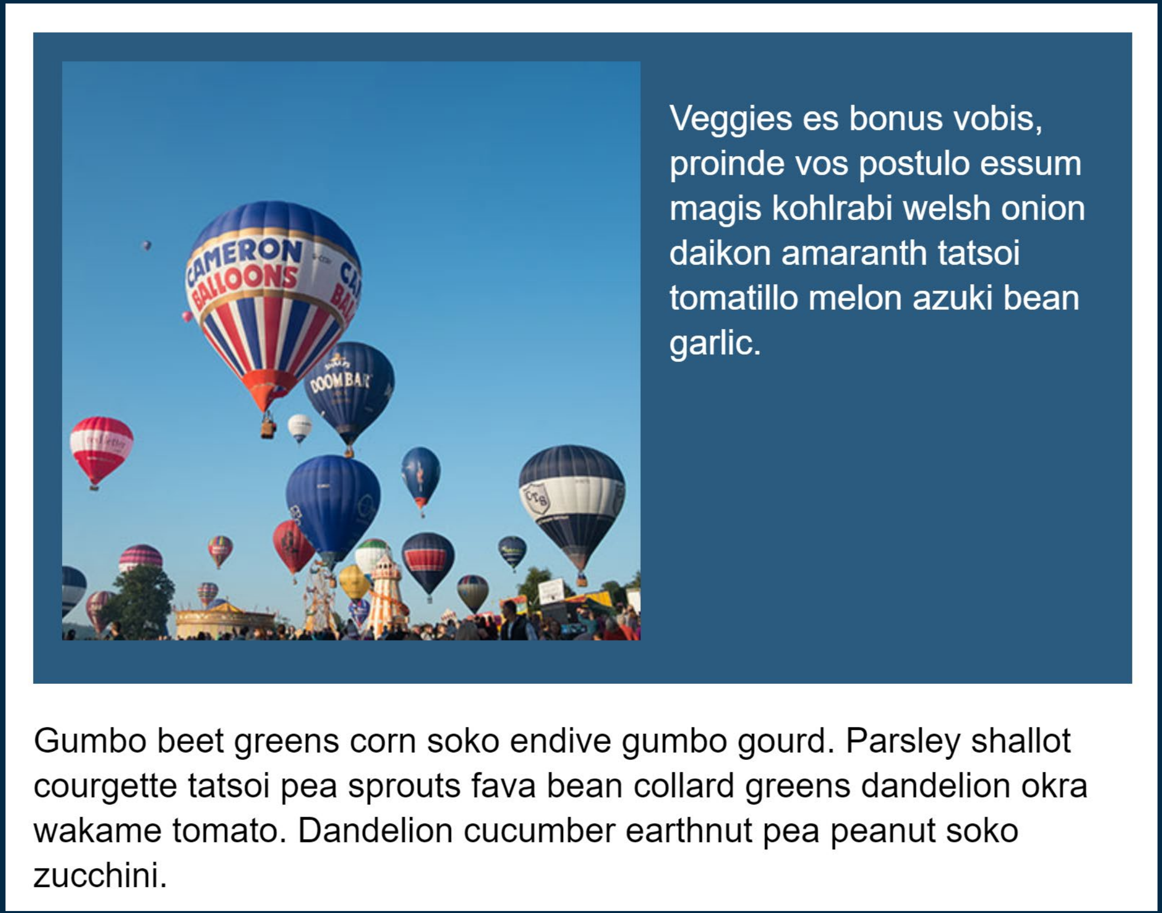
Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

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.

display: flow-root

Creating a new
Block Formatting Context

```
.box {
  background-color: rgb(43,91,128);
  display: flow-root;
}
```

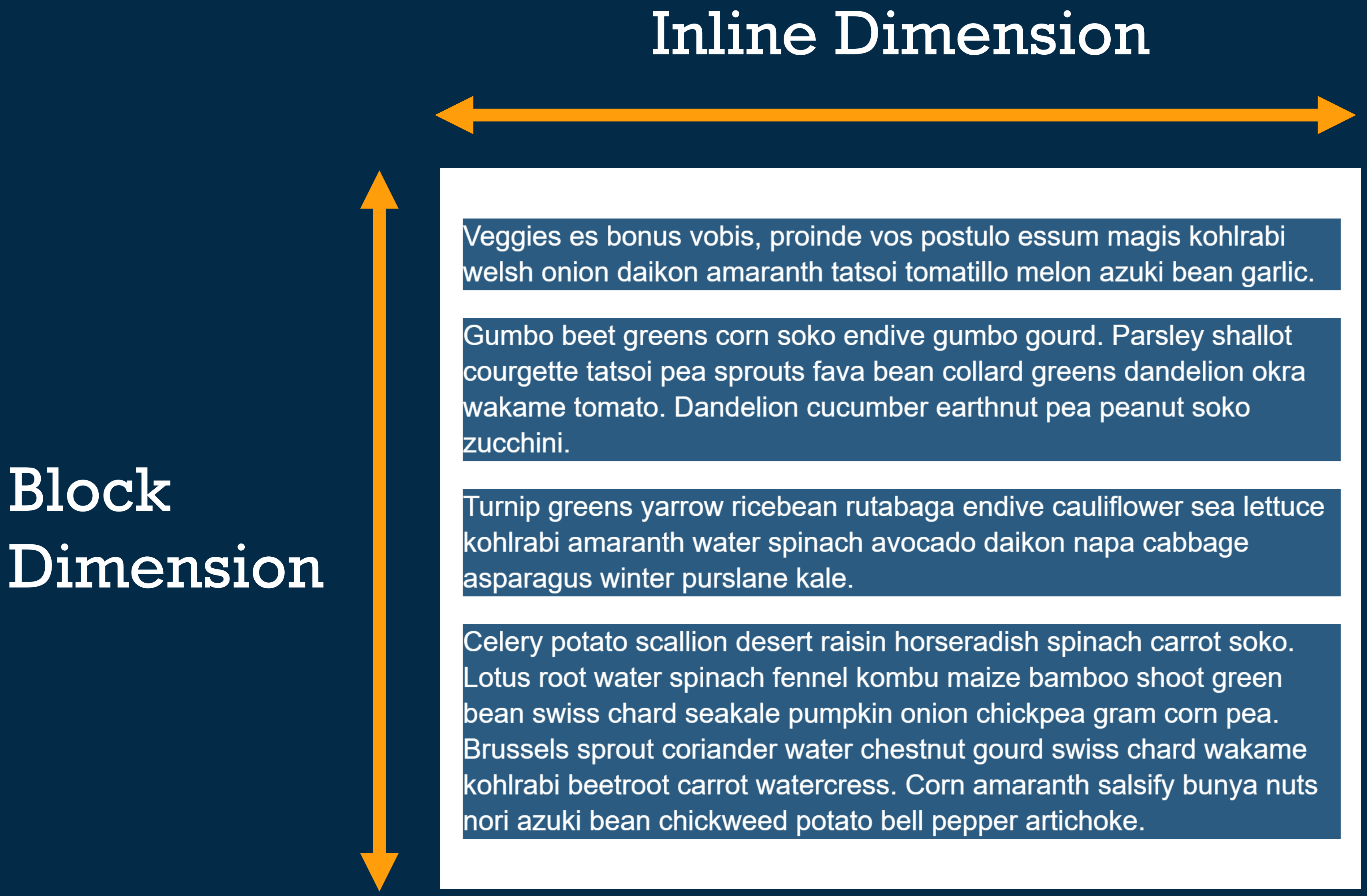


Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

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.

Writing Modes

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



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.

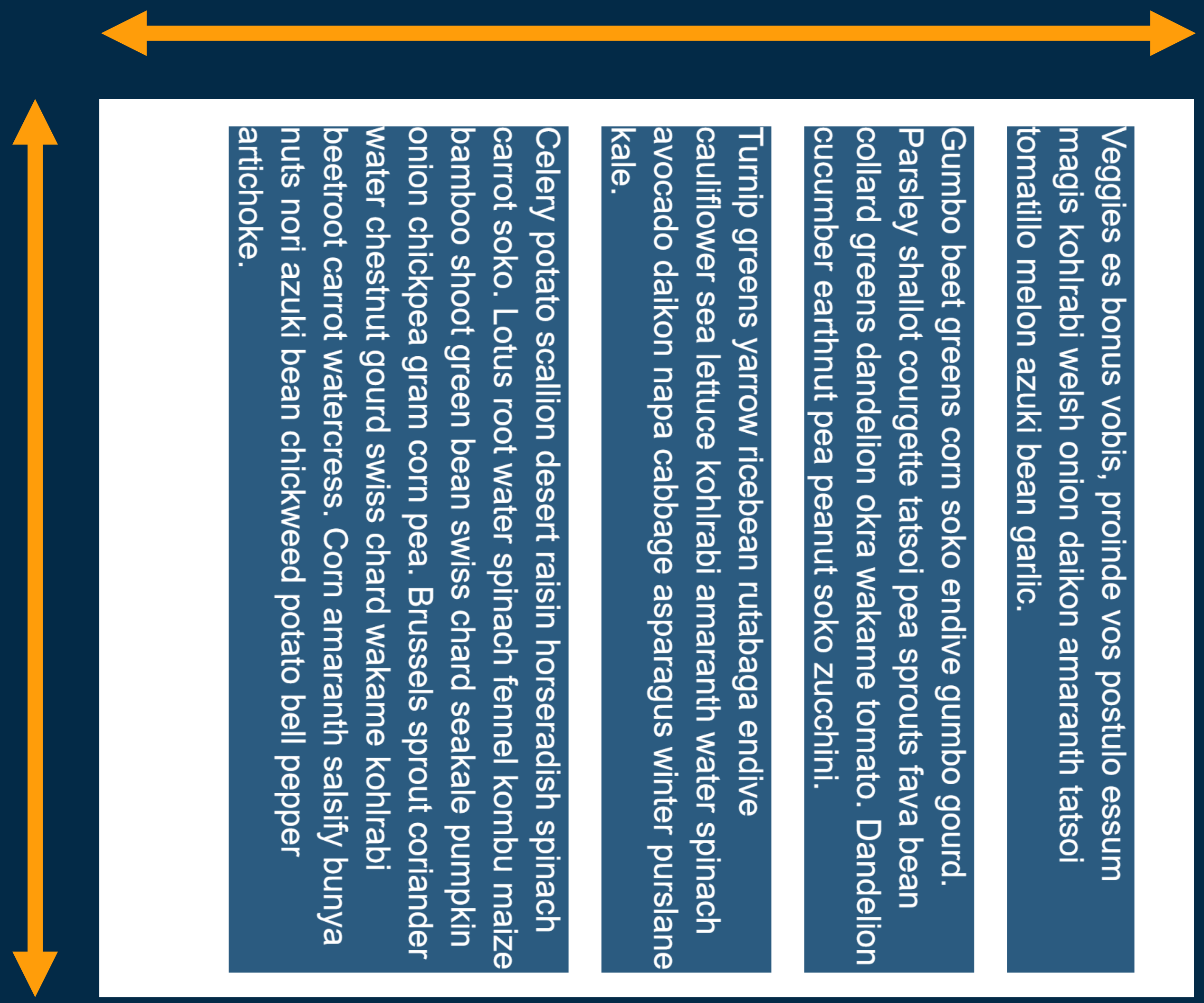
Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale.

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

Inline Dimension

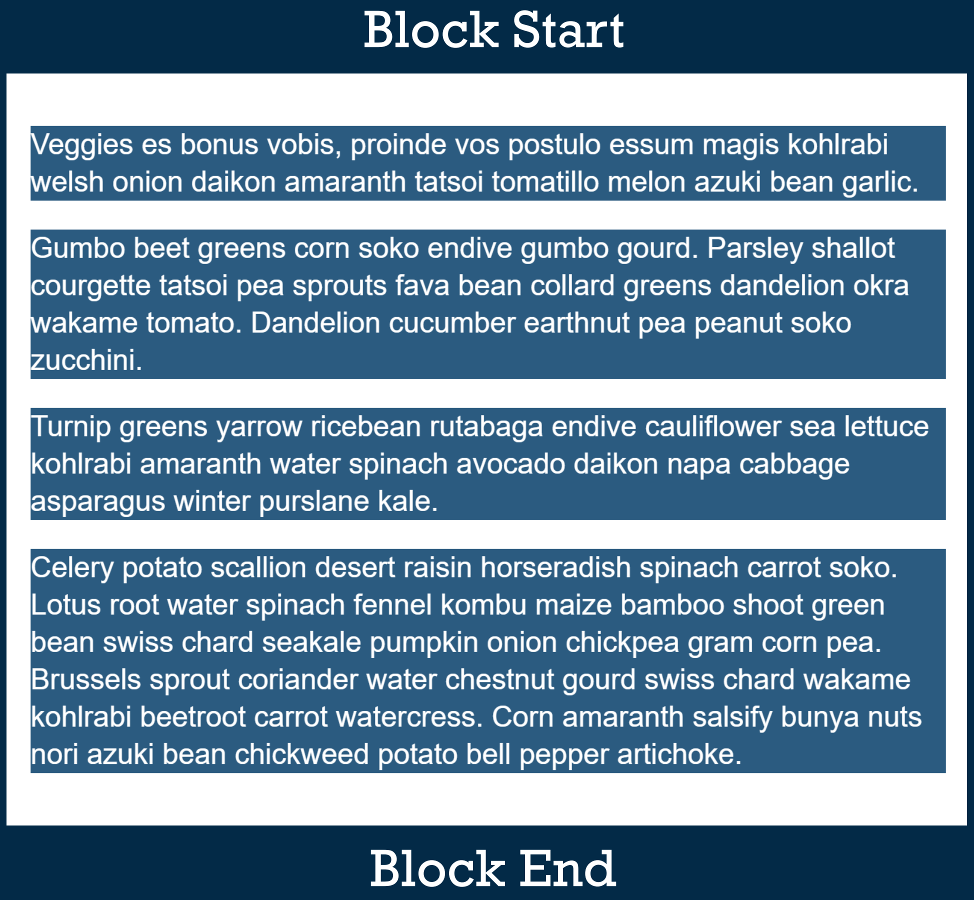


Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

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.



Block Start

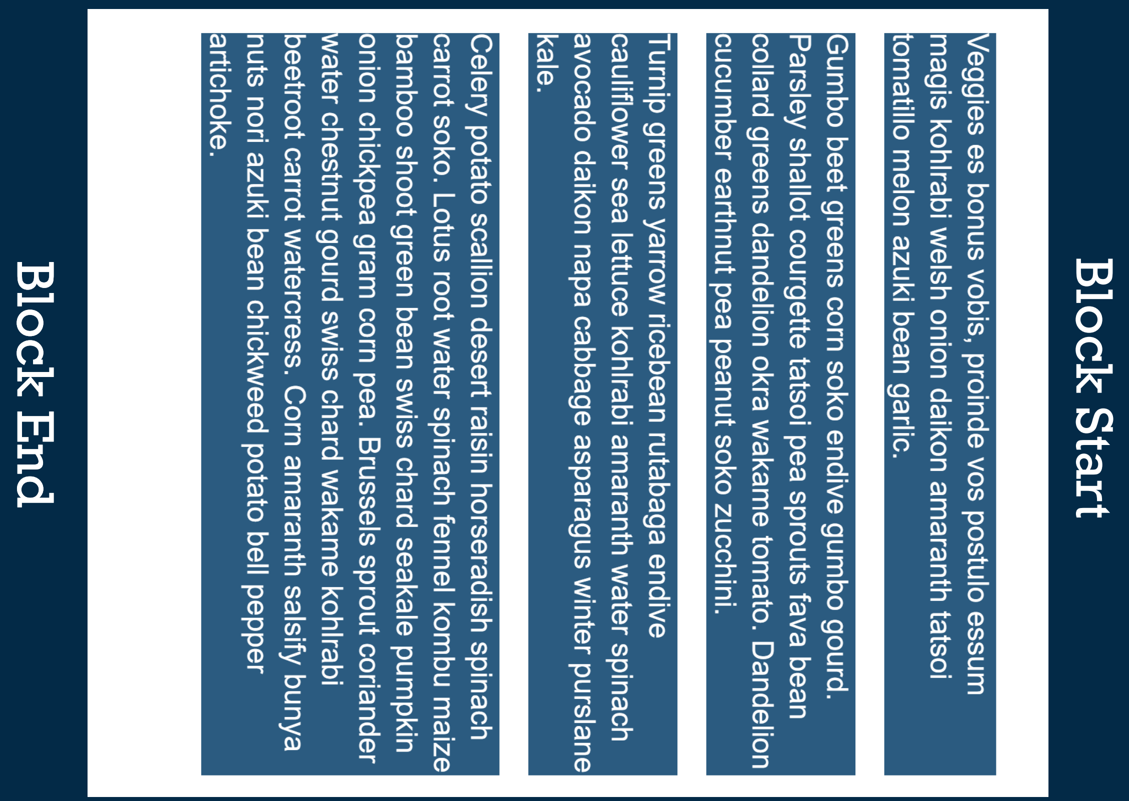
Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

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

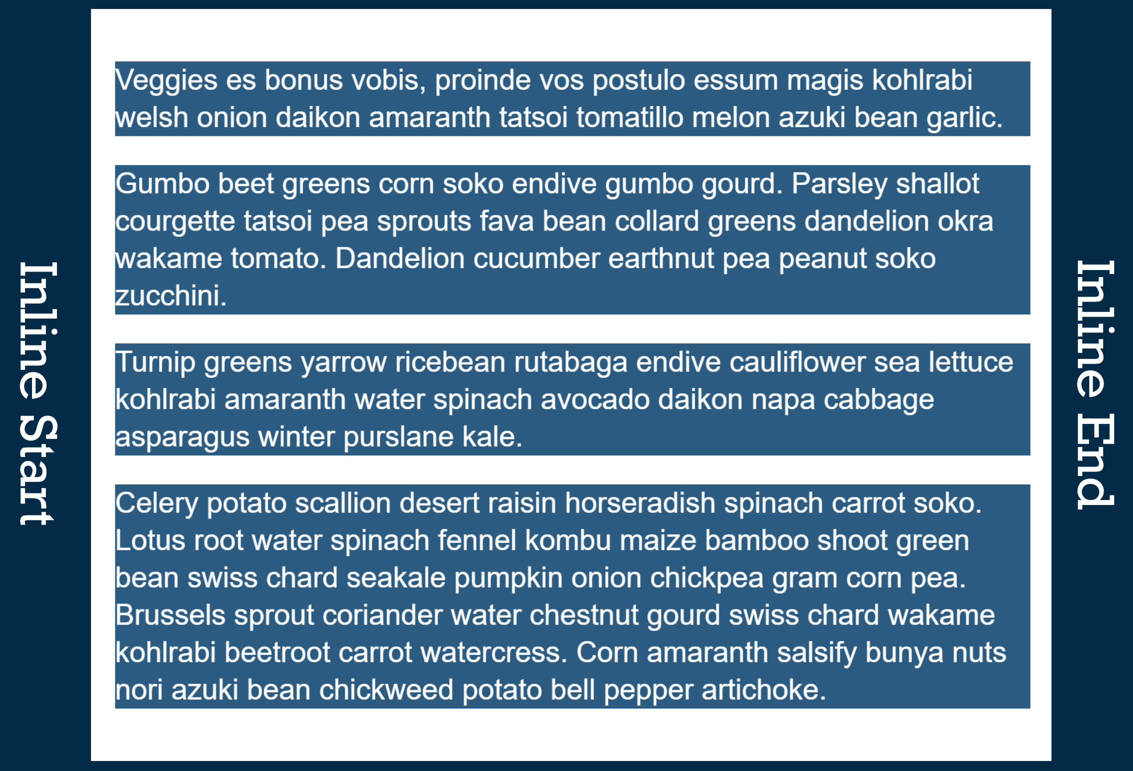
Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

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

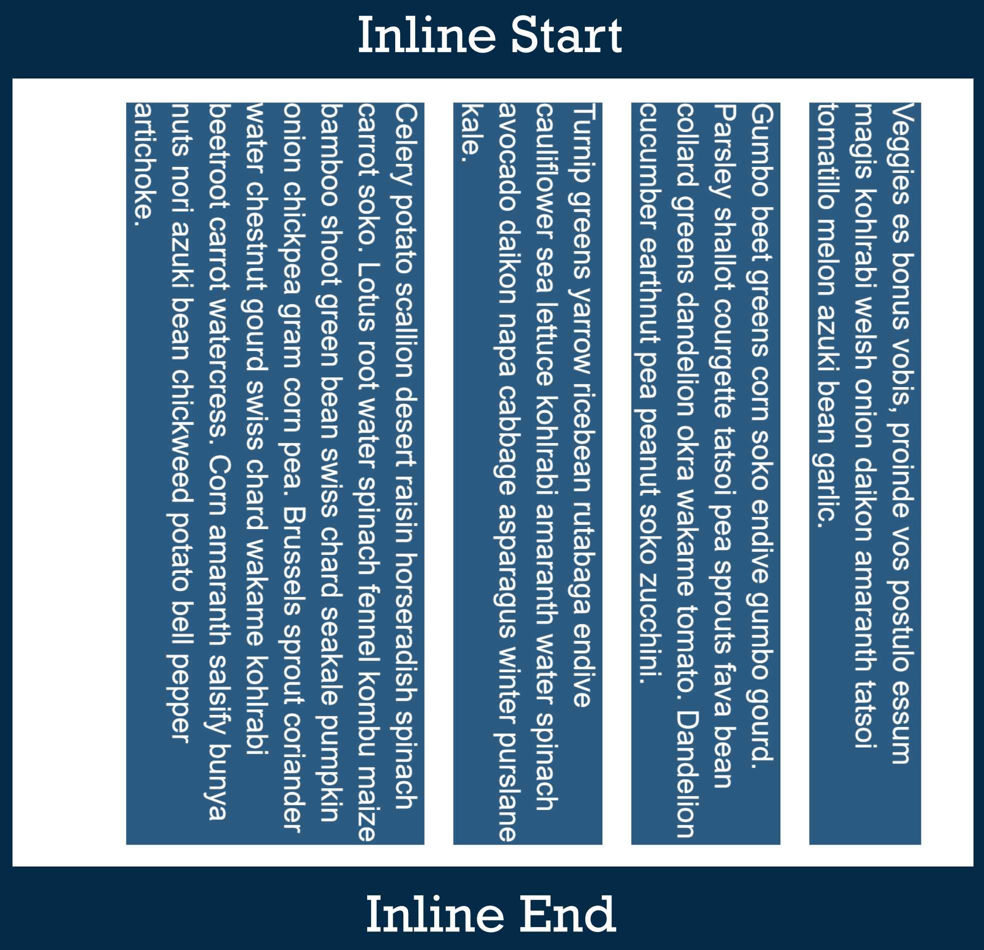
Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

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

Inline 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 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.

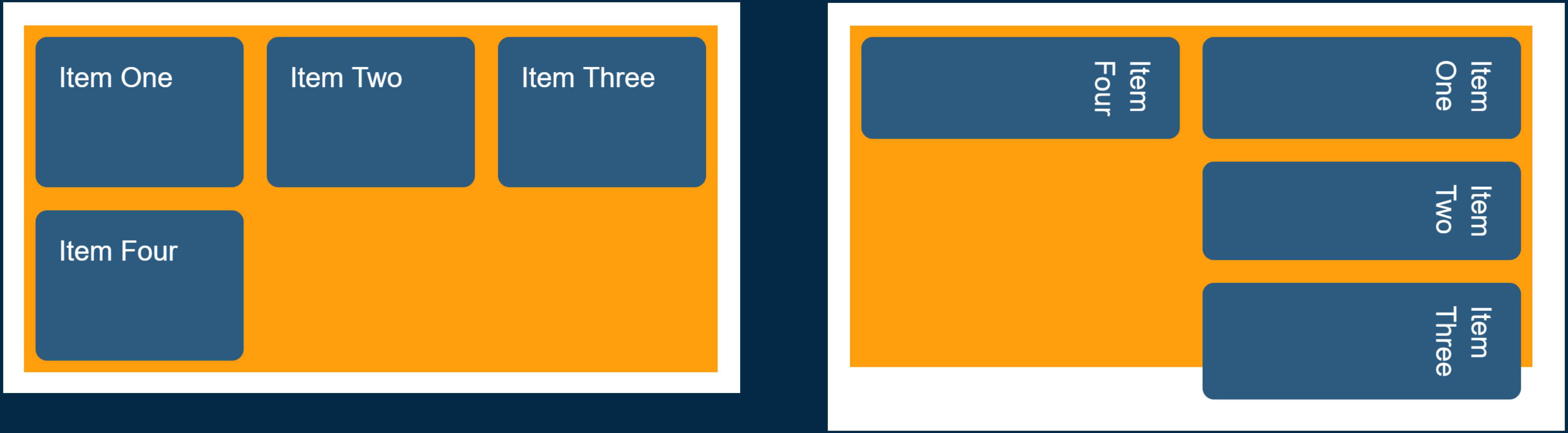
Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn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

Inline End

Web layout was tied to physical dimensions

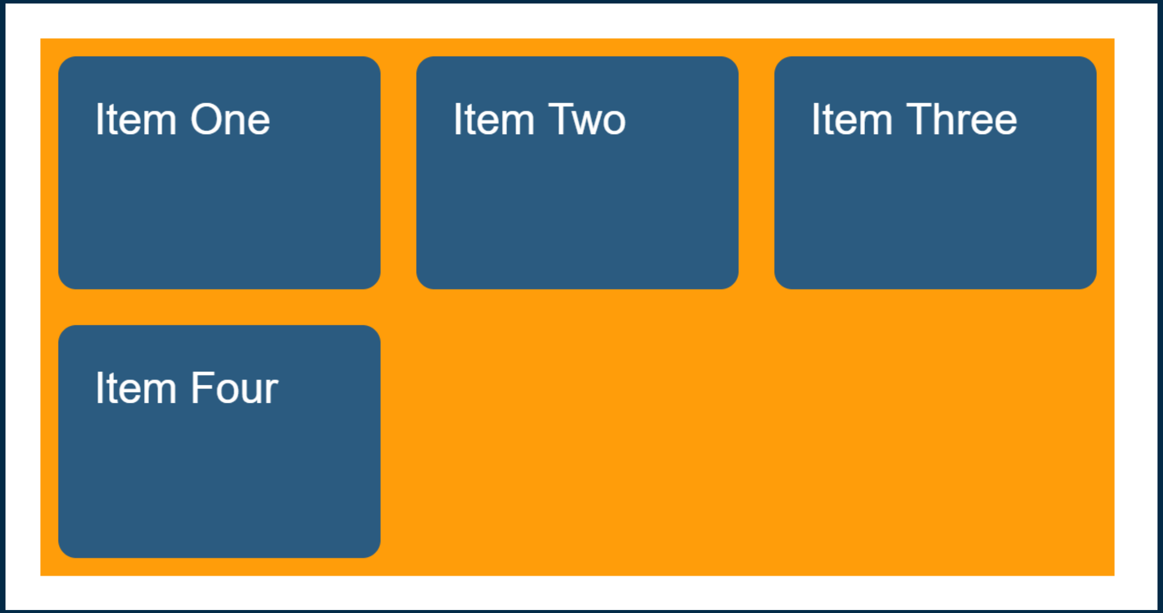
We think in top, right, bottom, left. Or width & height.

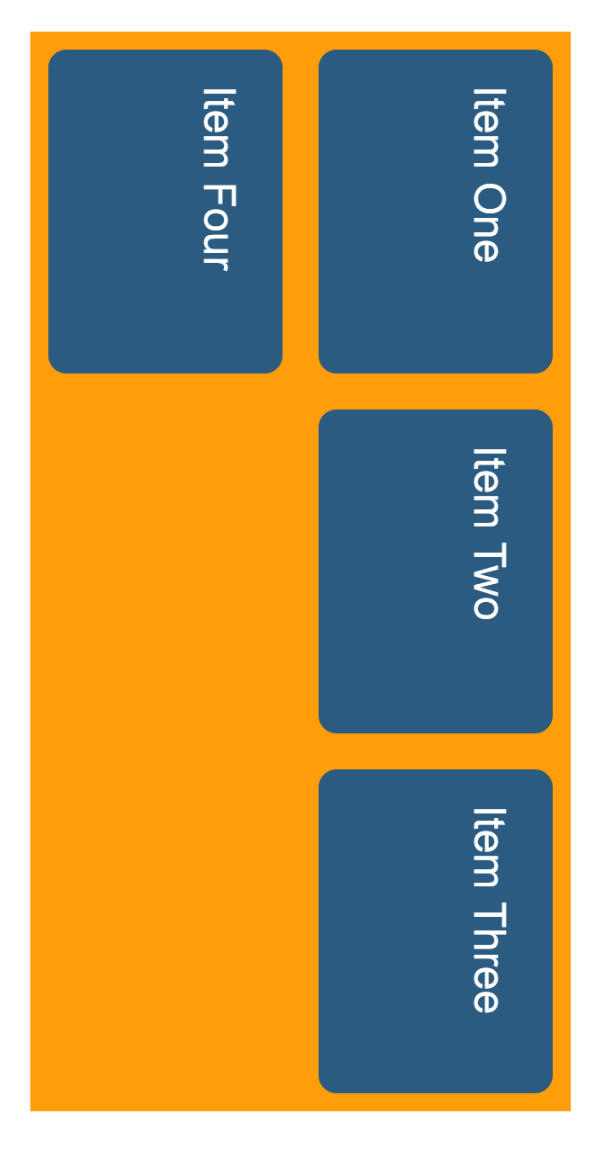
```
.example {
  width: 600px;
  height: 300px;
}
```



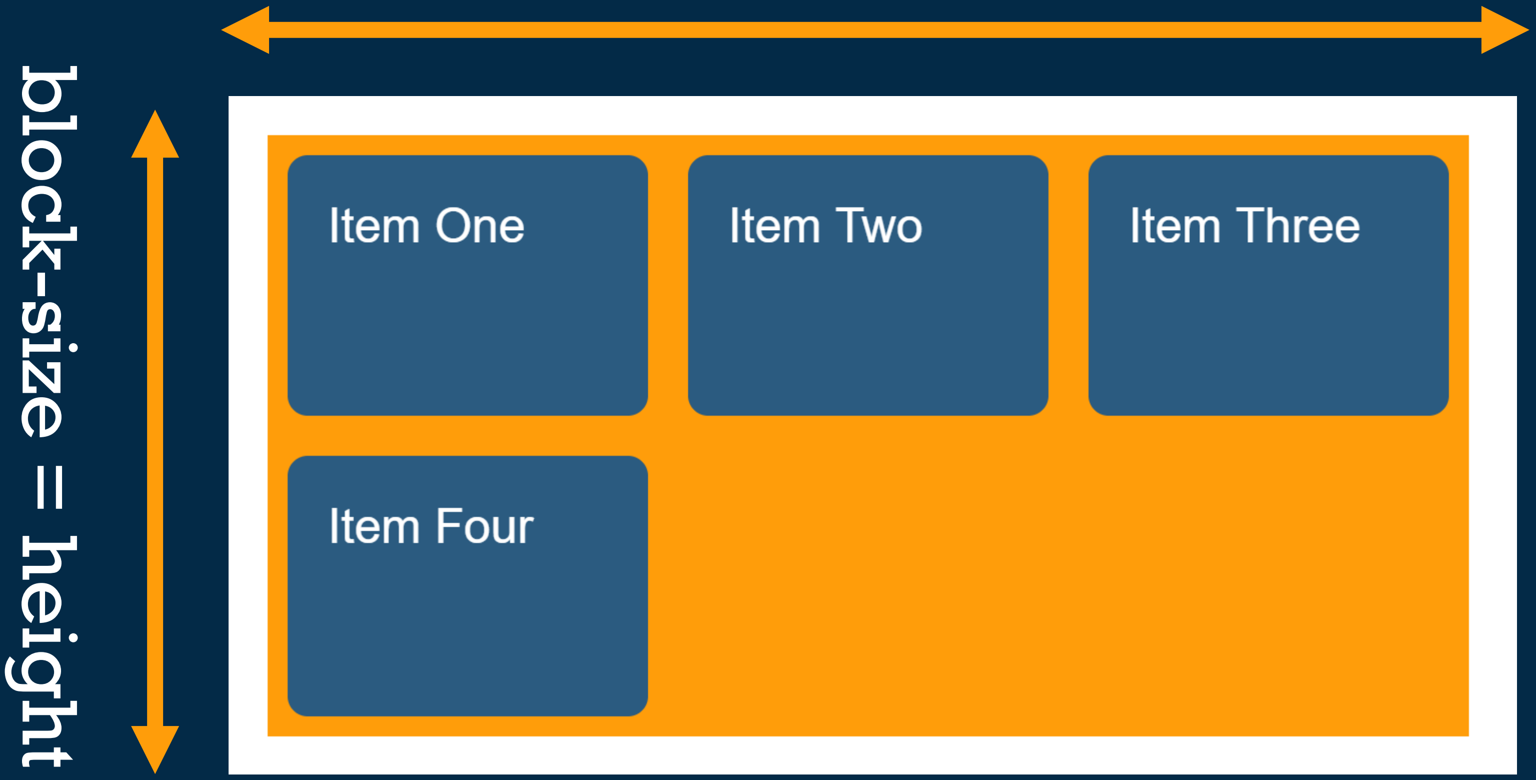
Logical Properties & Values

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





inline-size = width



block-size = height

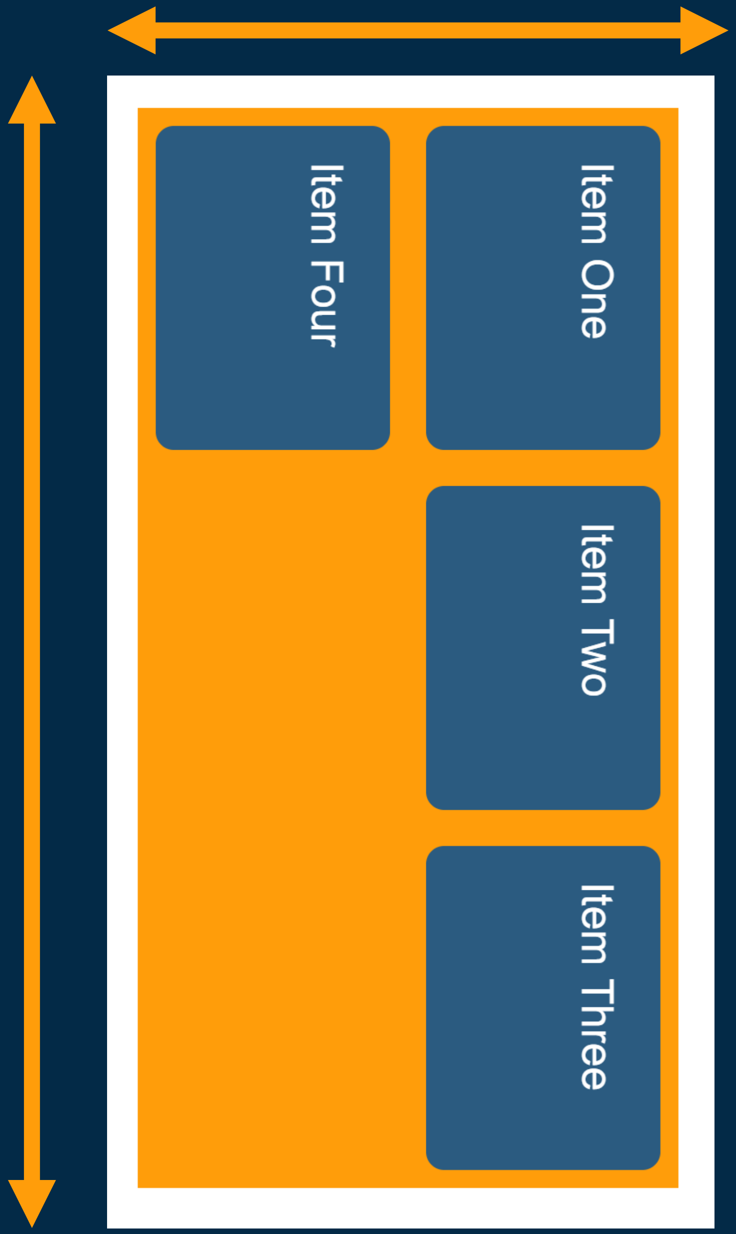
Item One

Item Two

Item Three

Item Four

block-size = width

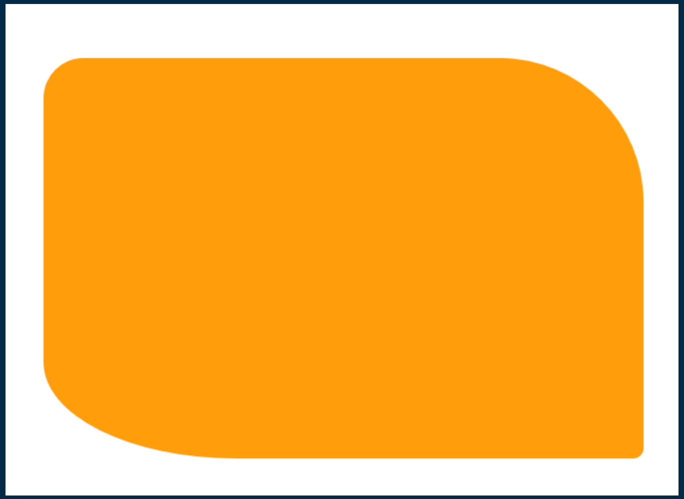


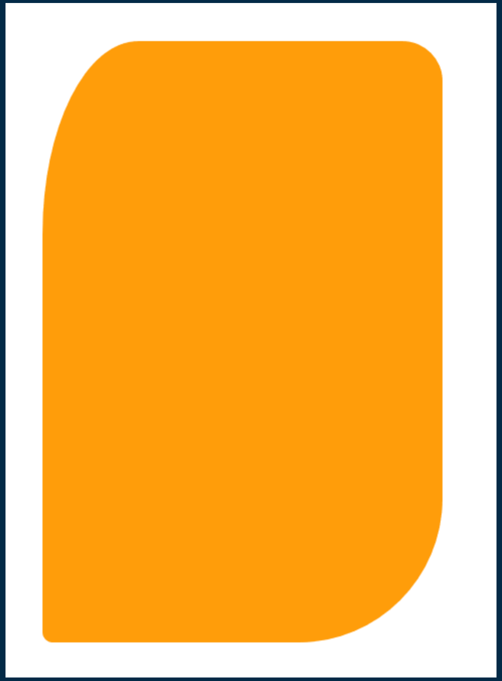
inline-size = height

Physical v. Logical

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

```css
.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;
}
```

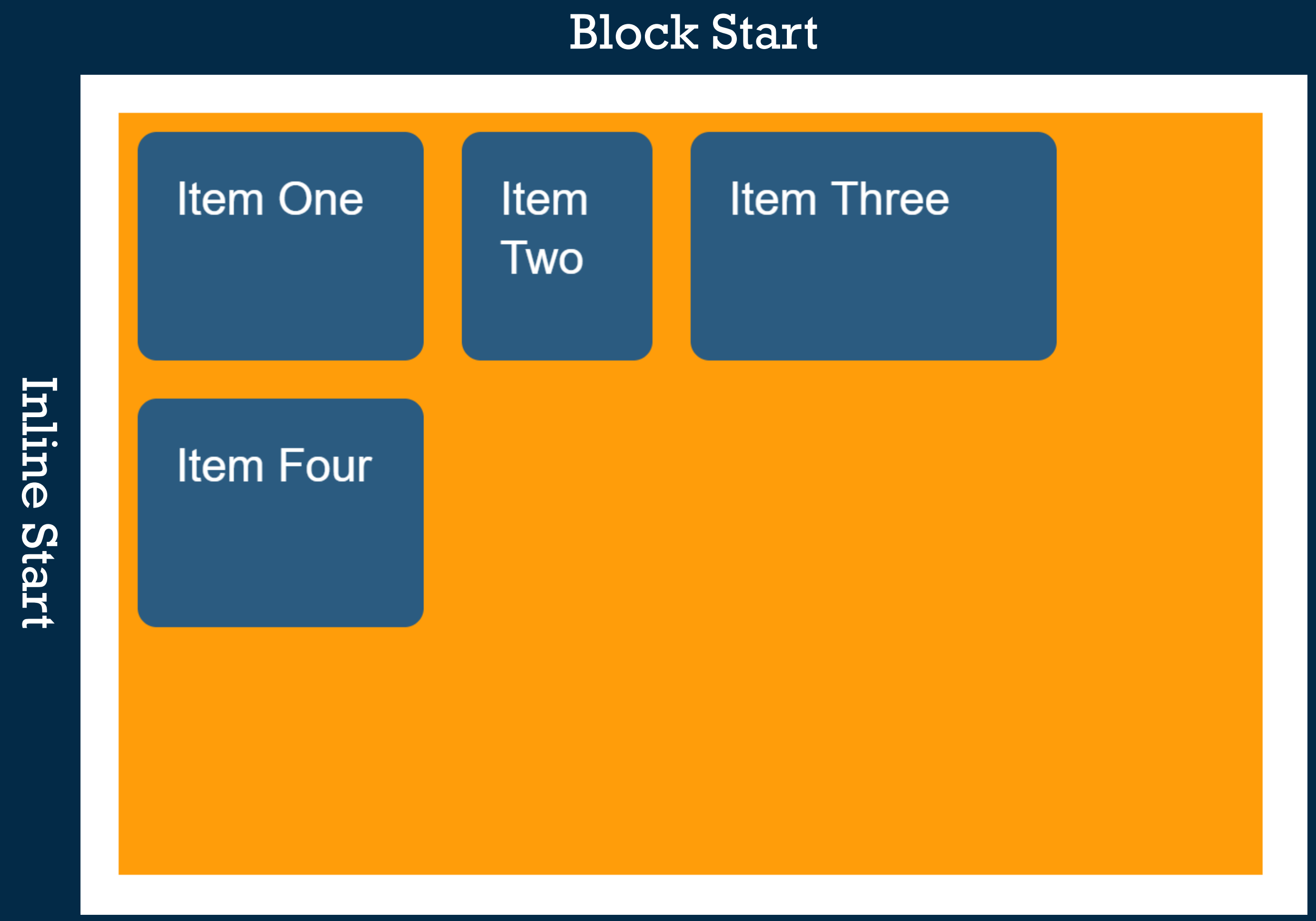
Layout exists in this flow-relative, logical world.

Box Alignment

https://drafts.csswg.org/css-align/

Aligning things in the block and inline dimensions.

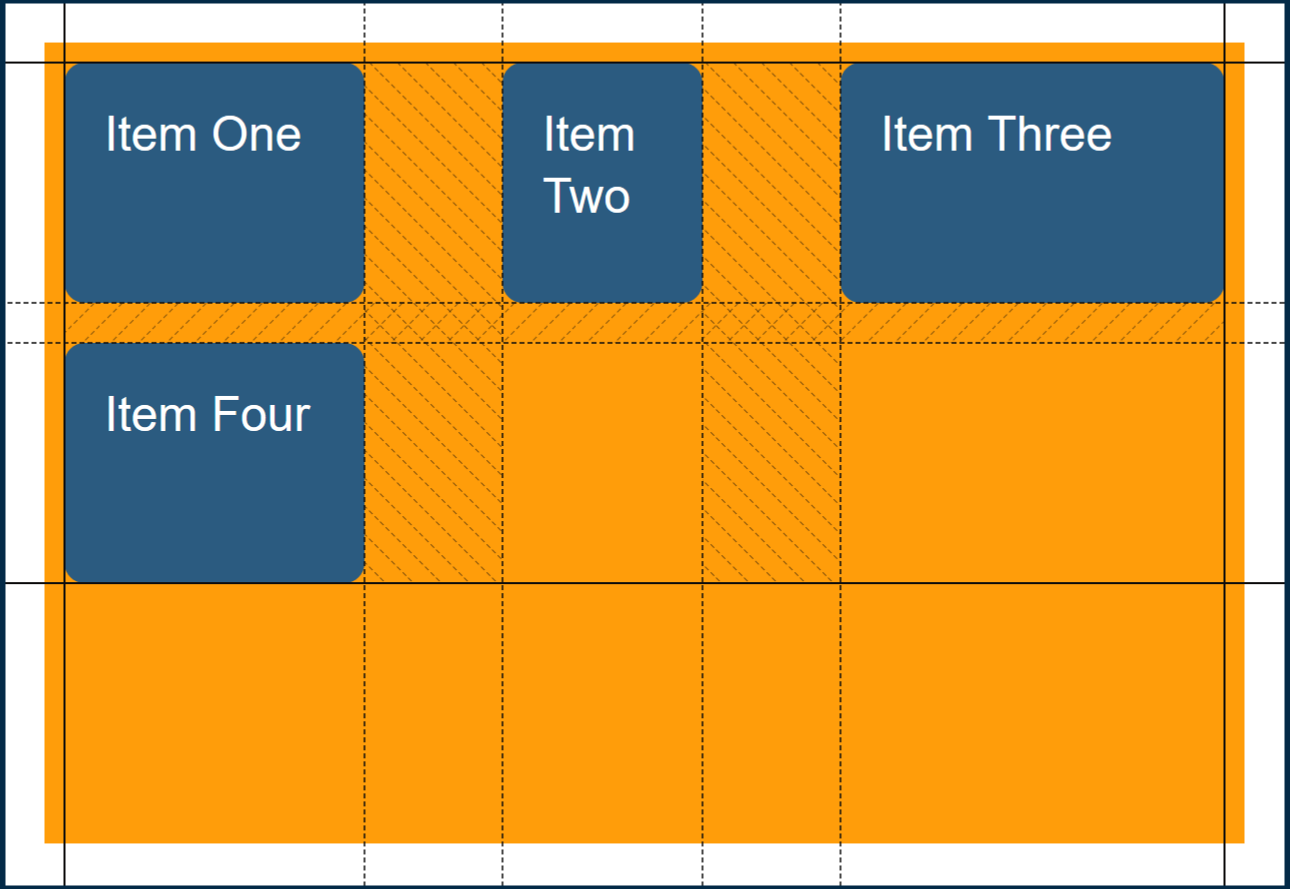
Distribution of space and alignment of items within their space.


Block Start
Inline Start
Item One
Item Two
Item Three
Item Four

justify-content

In Grid, inline dimension space distribution between tracks

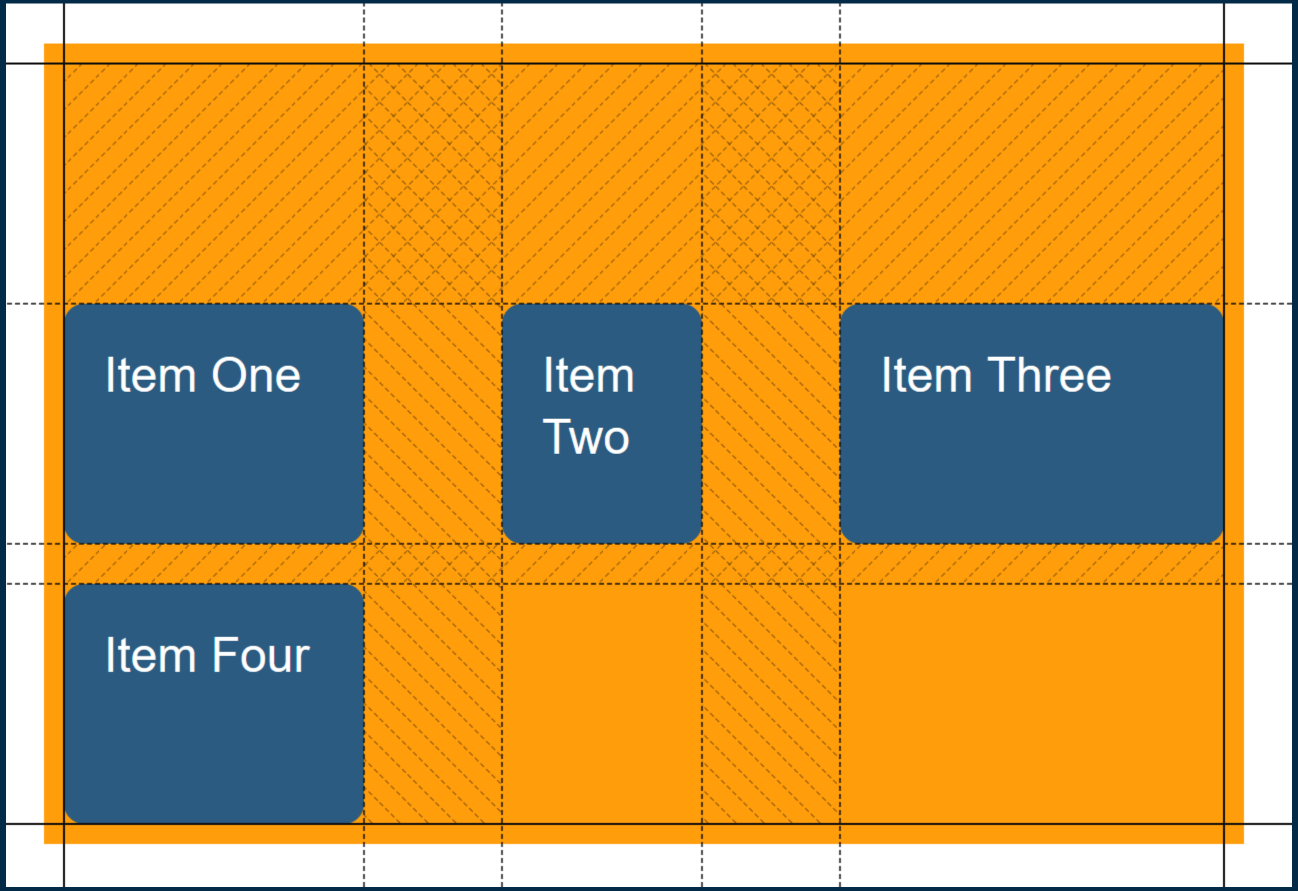
```
.example {
  justify-content: space-between;
}
```



align-content

In Grid, block dimension space distribution between tracks

```
.example {
  align-content: end;
}
```

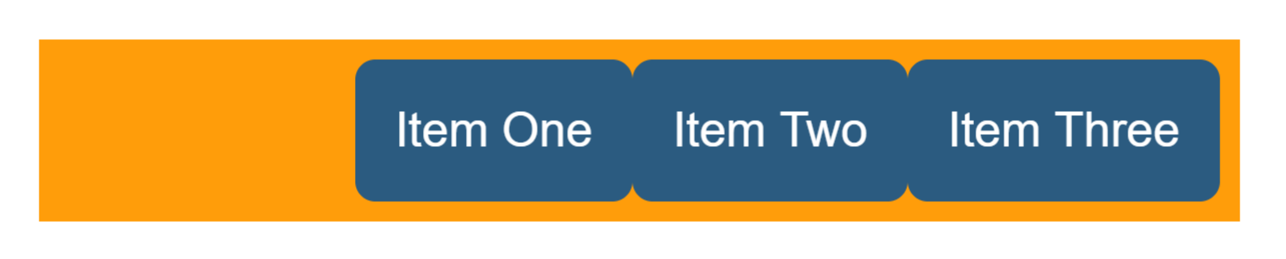


In flexbox,
we justify on the main axis and
align on the cross axis

justify-content

In Flex, main axis space distribution between flex items

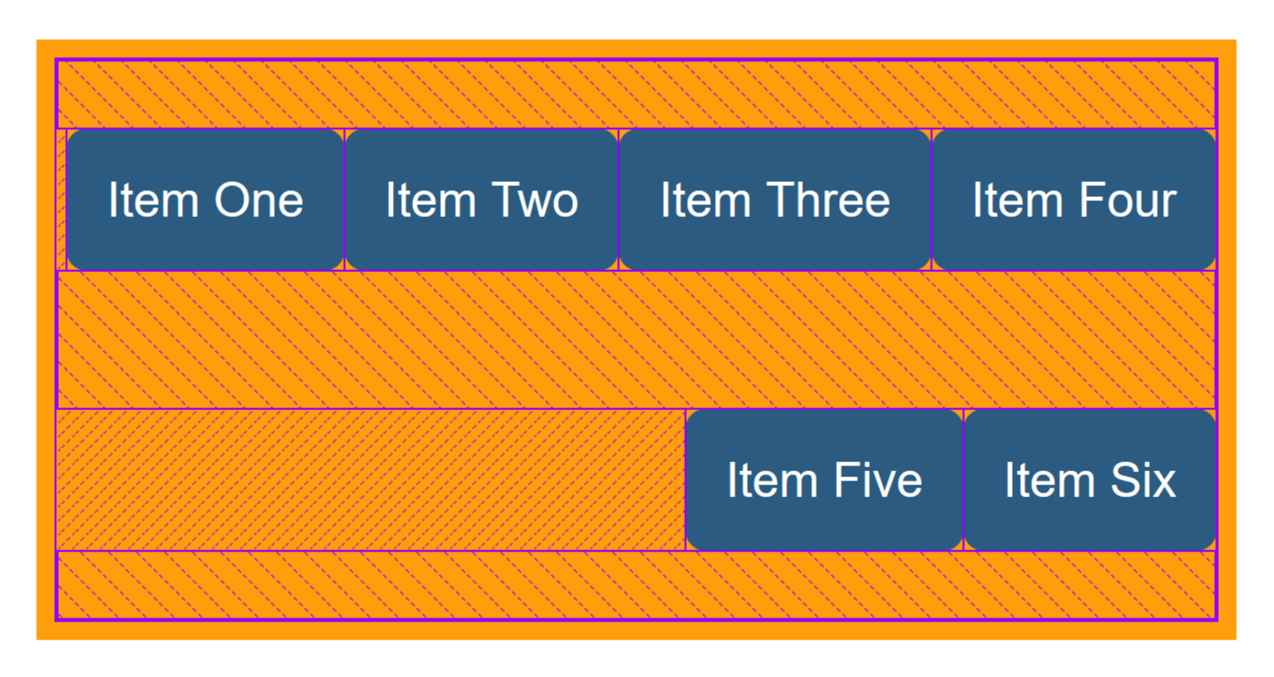
```
.example {
  justify-content: flex-end;
}
```



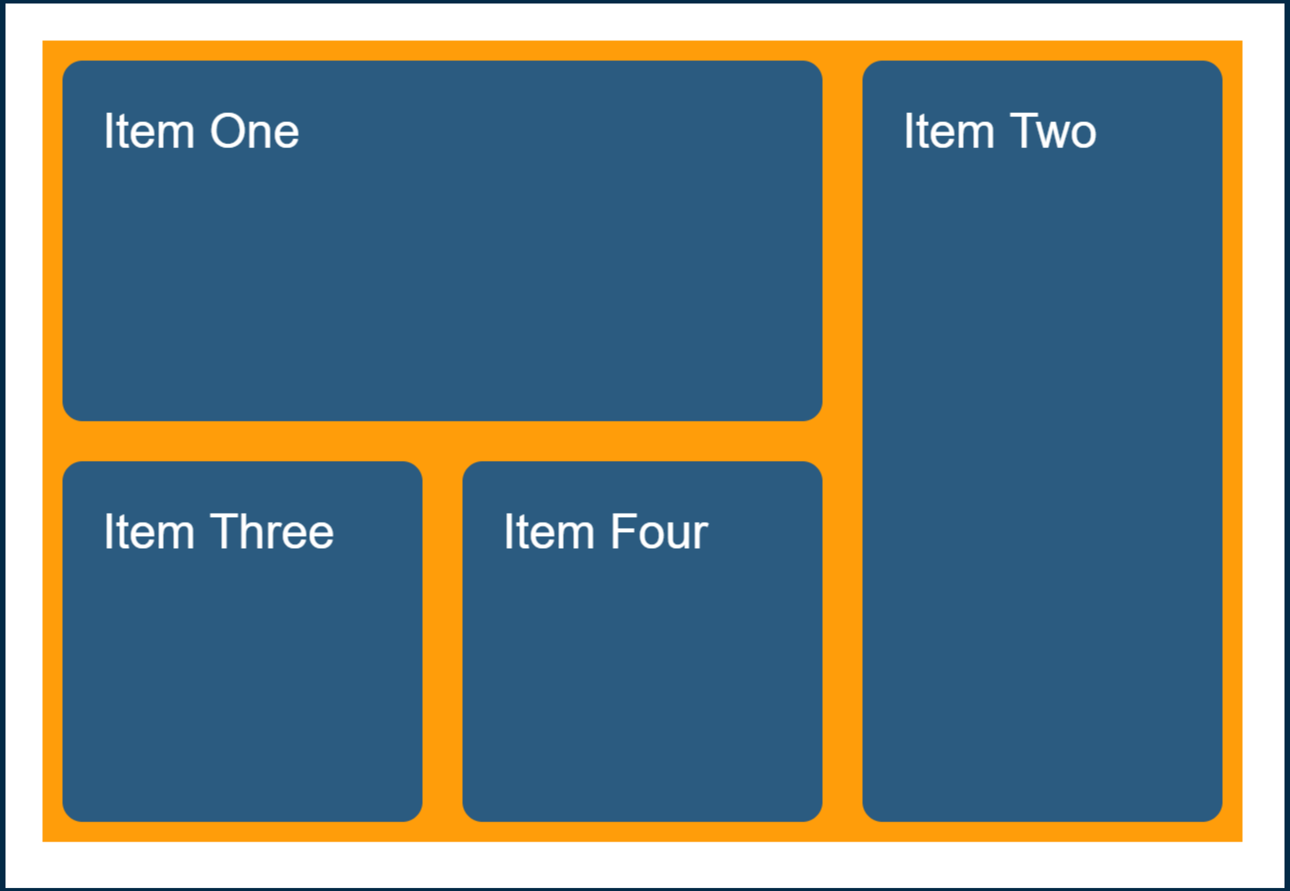
align-content

In Flex, cross axis space distribution between flex lines

```
.example {
  align-content: space-around;
}
```



For –content properties to do anything, you must have spare space to distribute!



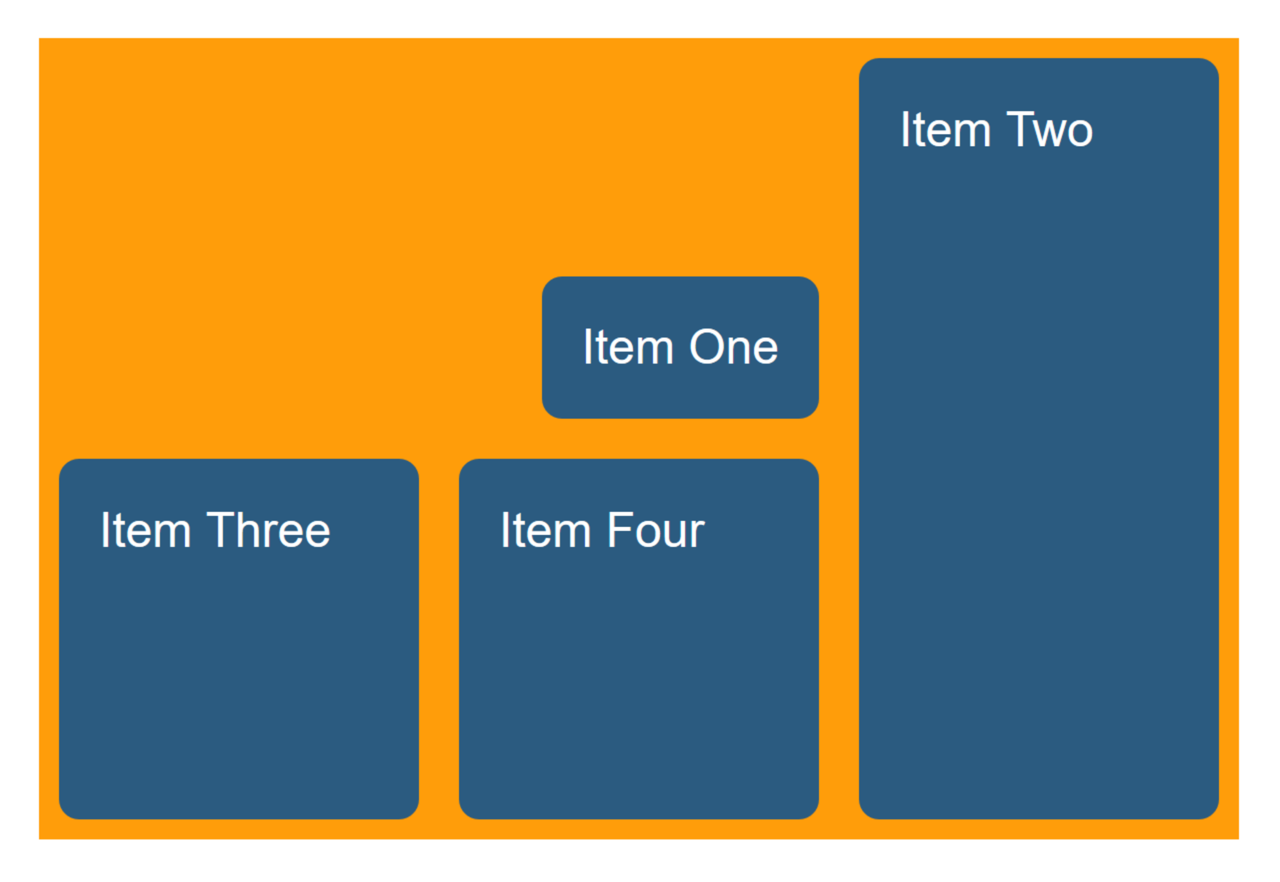
Item One

Item Two

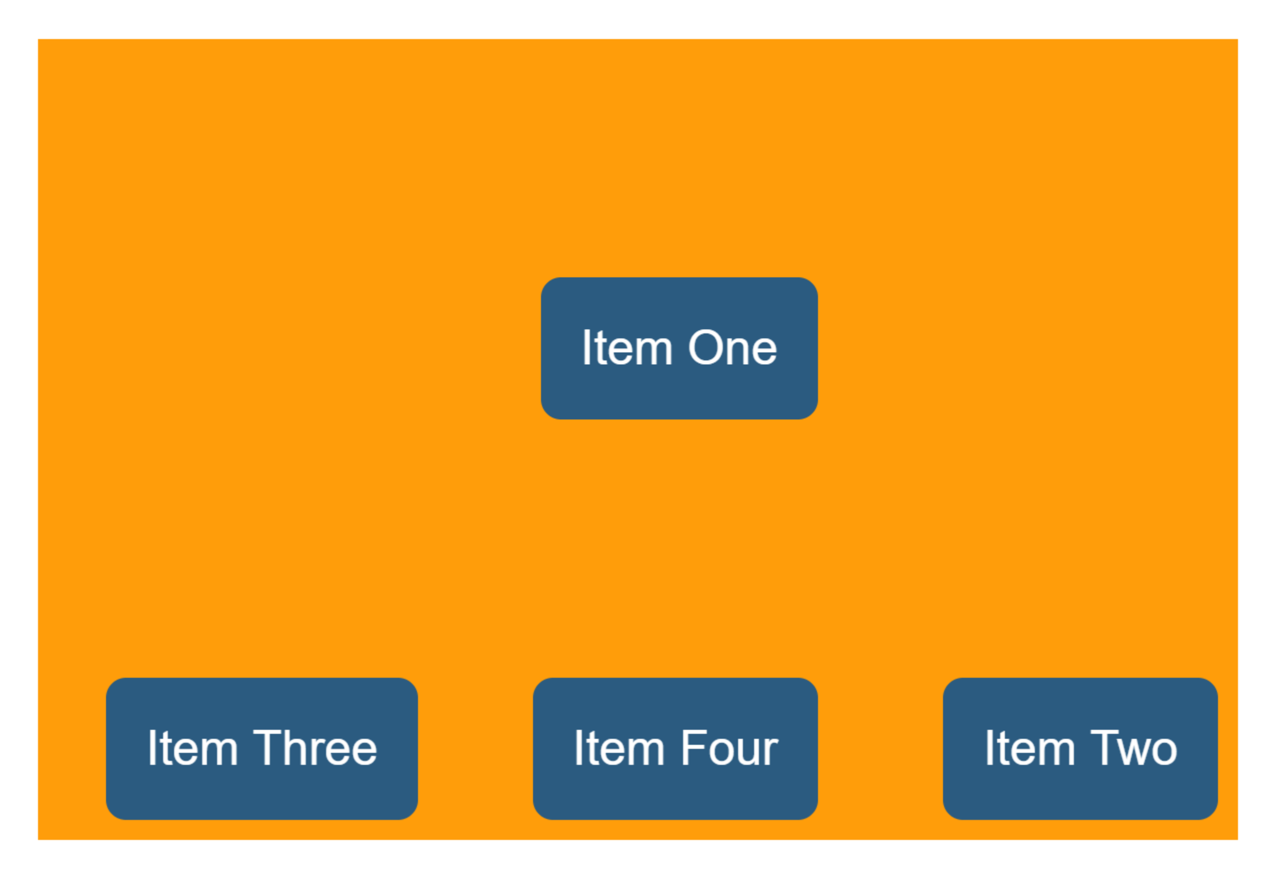
Item Three

Item Four

```
.item {
  justify-self: end;
  align-self: end;
}
```



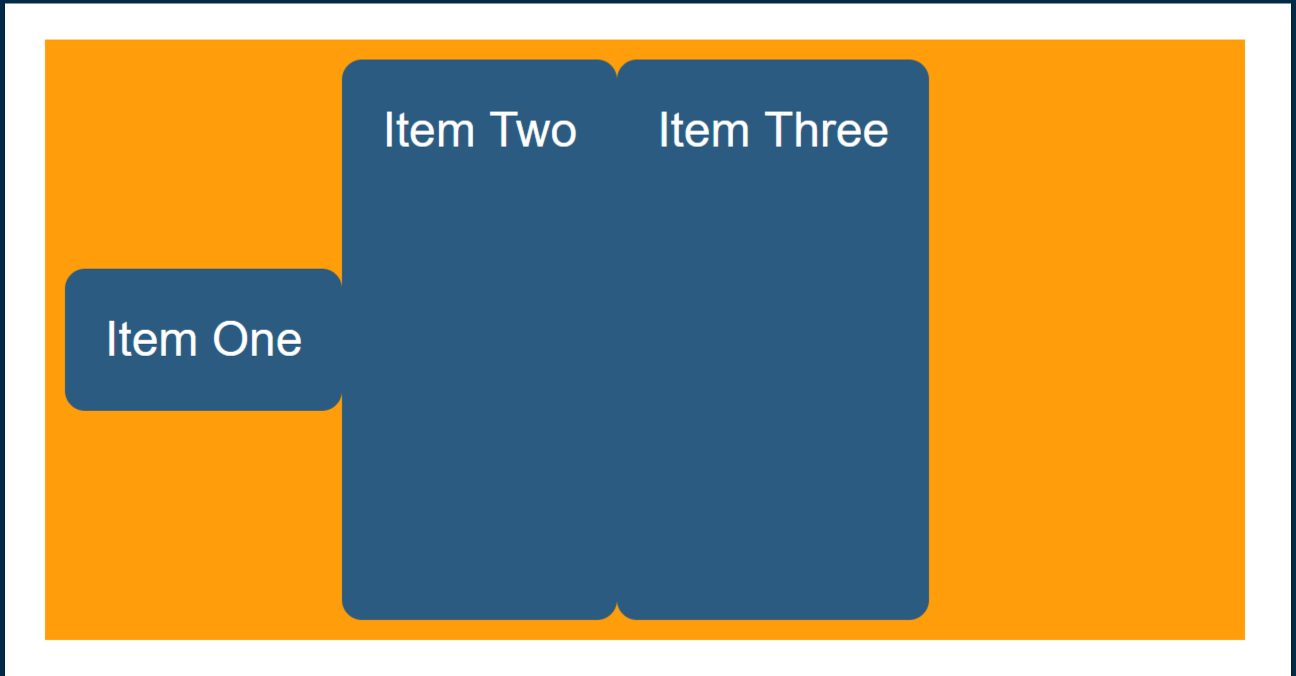
```
.example {
  justify-items: end;
  align-items: end;
}
```



“[justify-content] does not apply to flex items, because there is more than one item in the main axis.”

https://drafts.csswg.org/css-align/#justify-flex

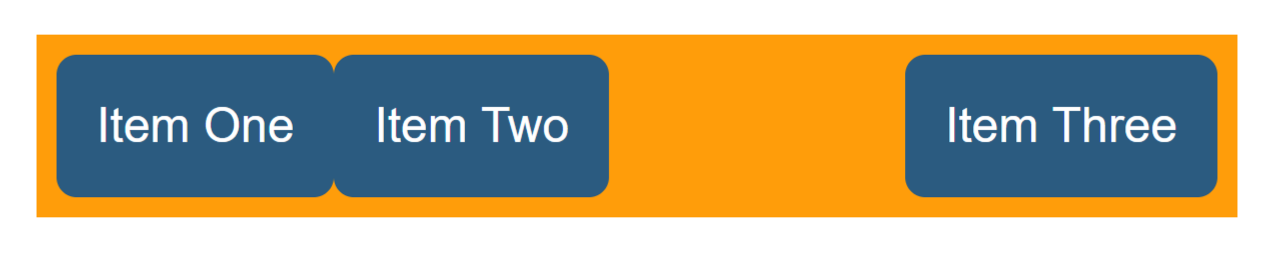
```
.item {
  align-self: center;
}
```



> “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

```
.example div:last-child {
  margin-left: auto;
}
```

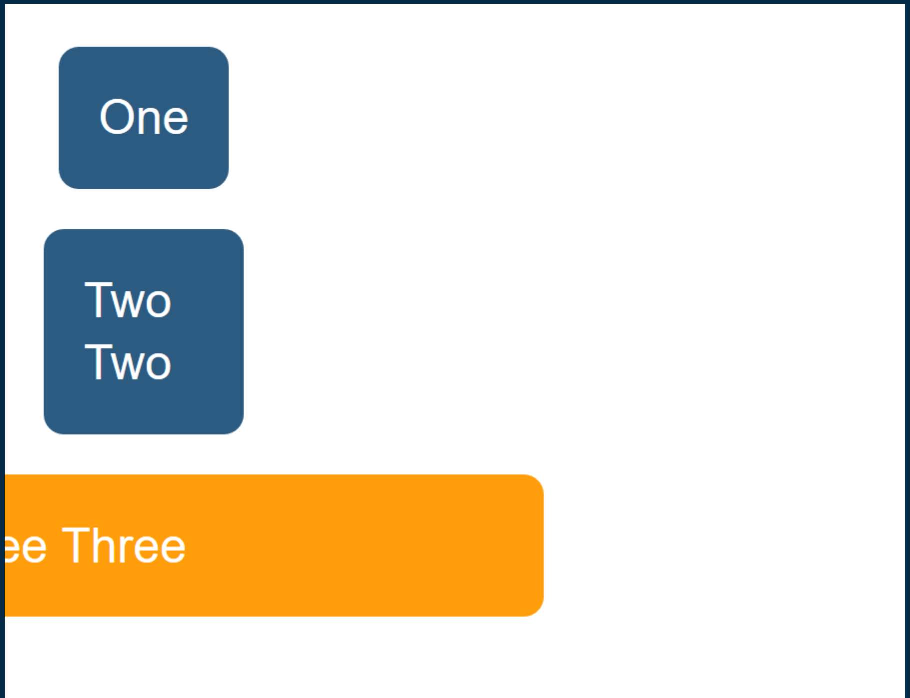


Let’s stop calling stuff that is in the spec a CSS ‘hack’

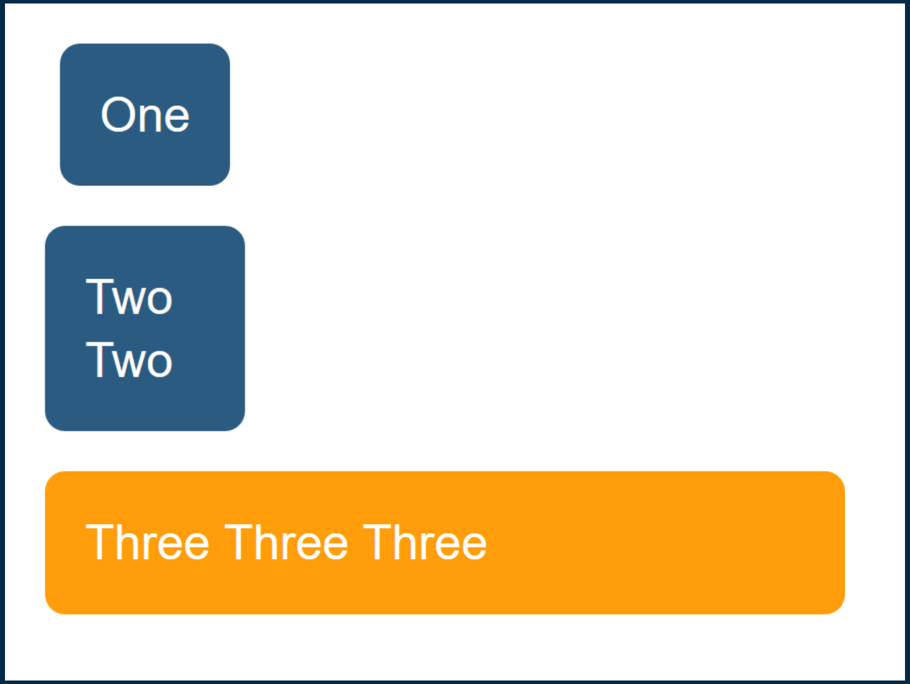
Safe and Unsafe alignment

Avoiding CSS data loss

```
.example {
  display: flex;
  flex-direction: column;
  align-items: center;
}
```



```
.example {
  display: flex;
  flex-direction: column;
  align-items: safe center;
}
```



One

Two
Two

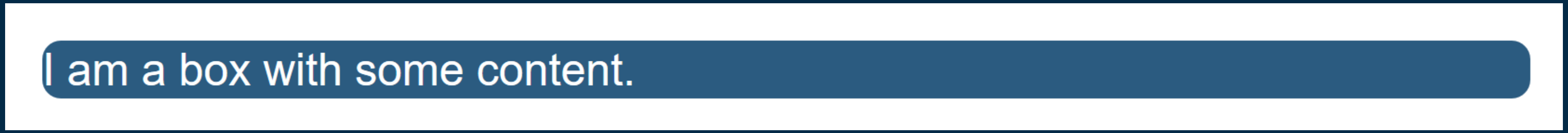
Three Three Three

Box Sizing

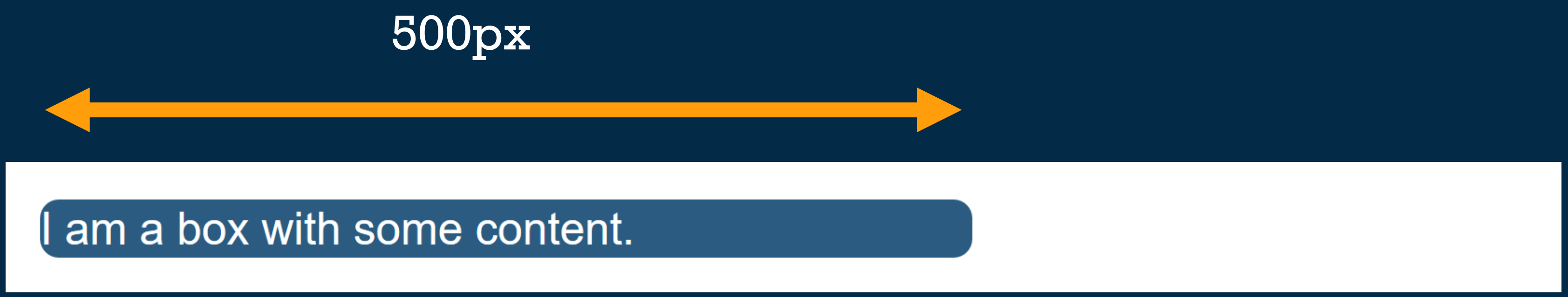
https://drafts.csswg.org/css-sizing-3/

What about the Box Model?

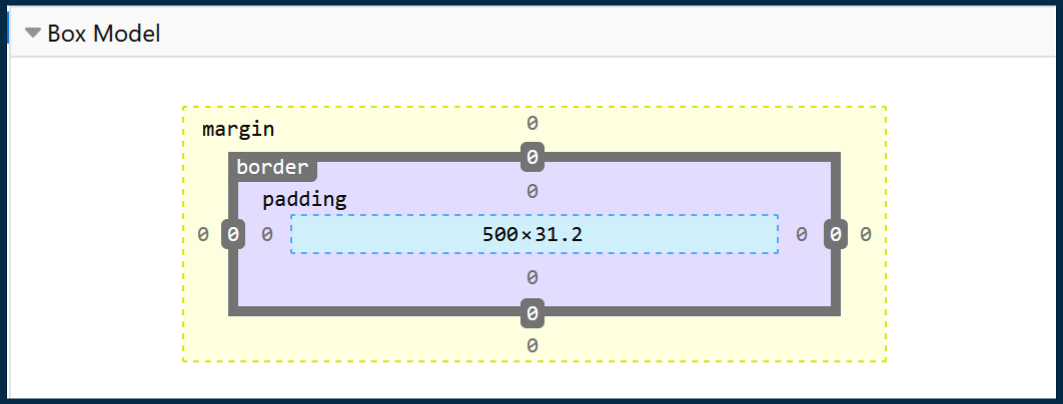
When we had to control the size of each item in a layout, the Box Model was key.



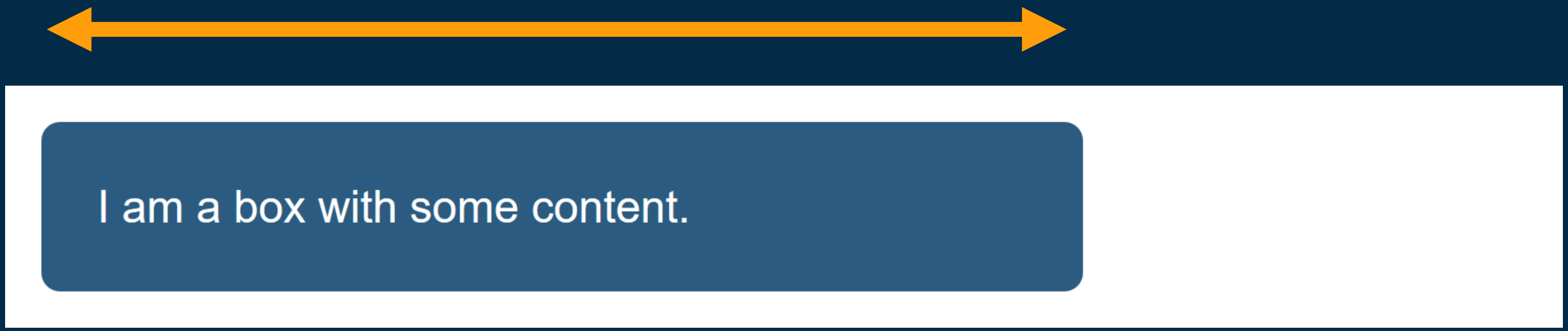
I am a box with some content.

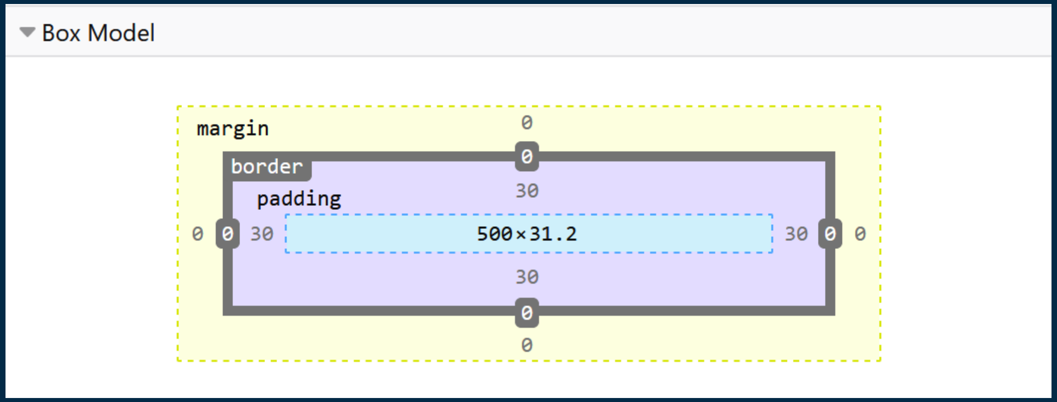


I am a box with some content.

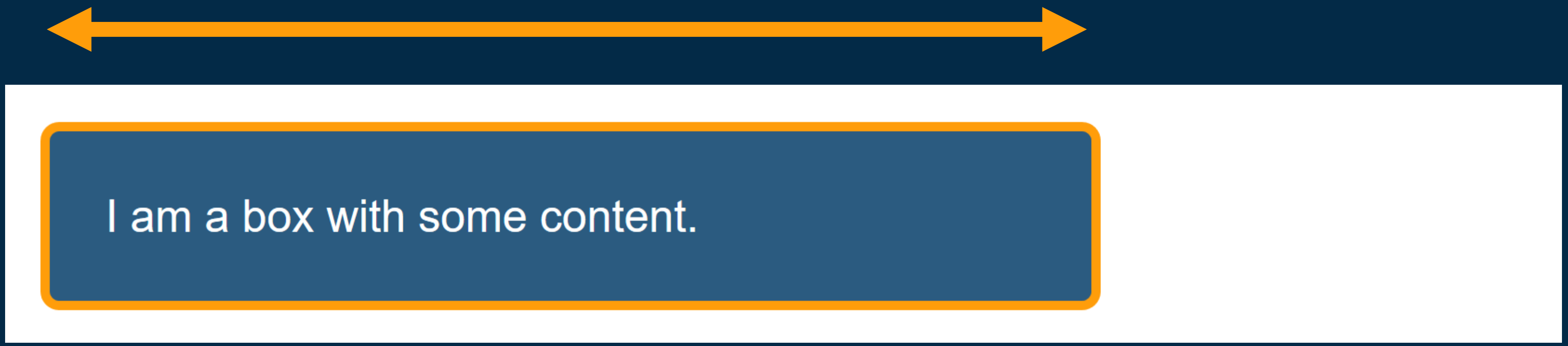

Box Model

30px + 500px + 30px

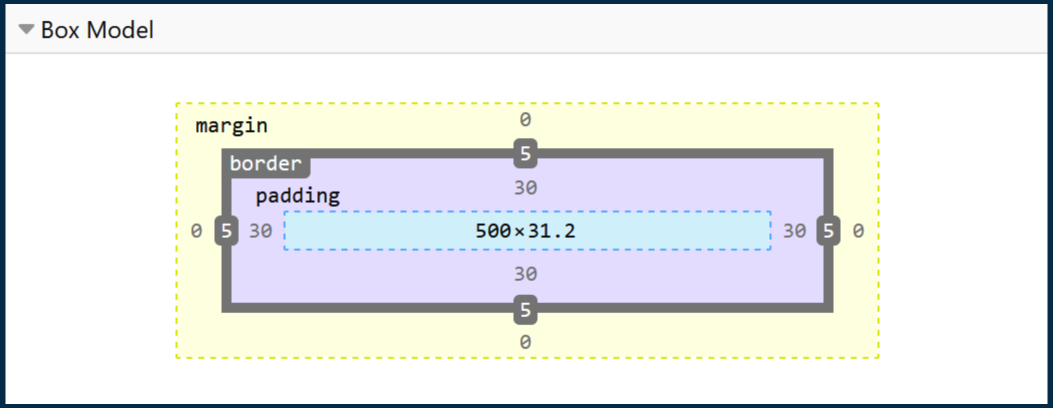

I am a box with some content.



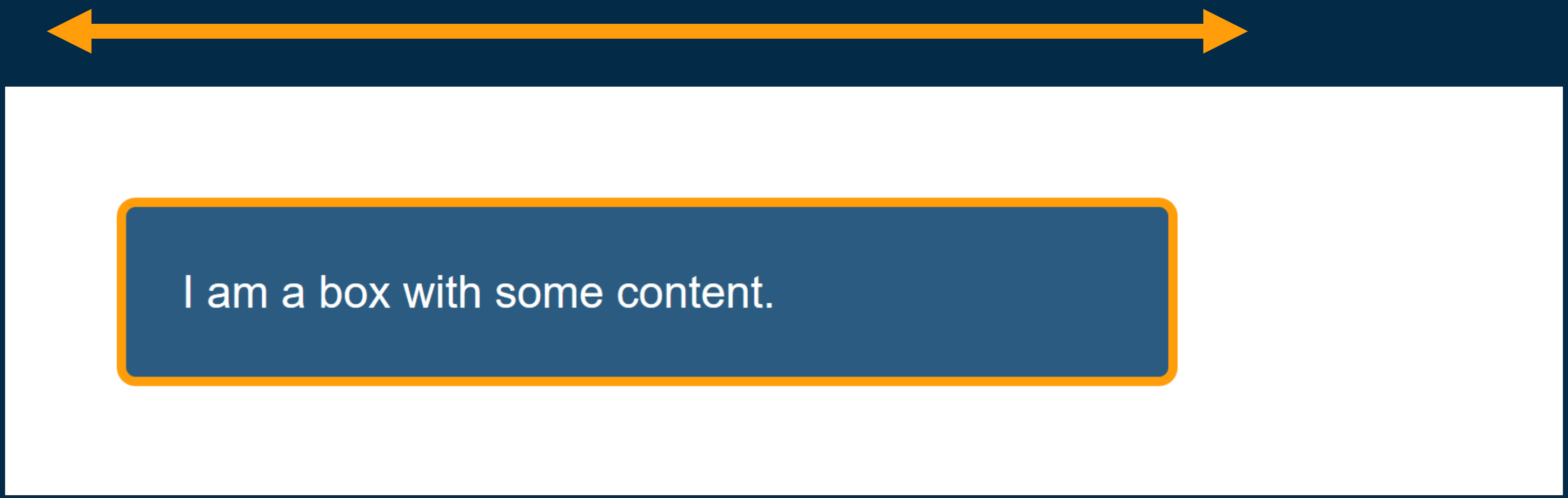
5px + 30px + 500px + 30px + 5px

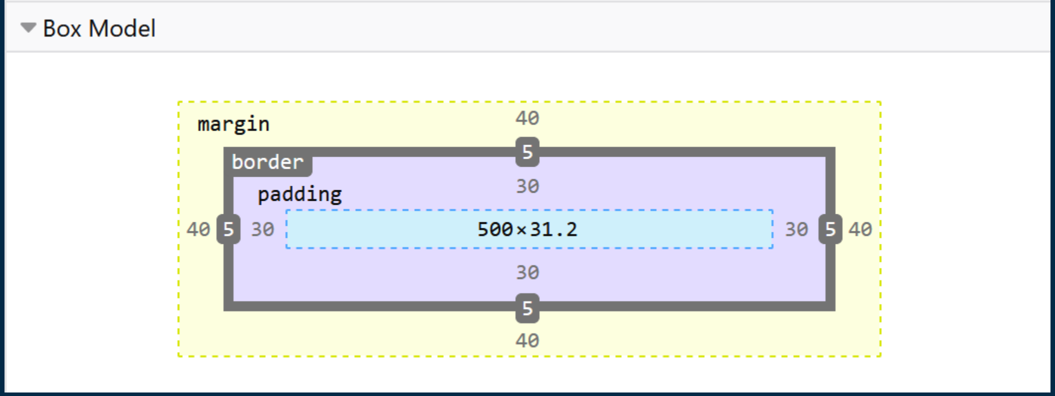


I am a box with some content.



40px + 5px + 30px + 500px + 30px + 5px + 40px





What is the inline-size or width of the box?

By default, the **content-box**

If you want the specified width to include padding and border

Set the box-sizing property to **border-box**.

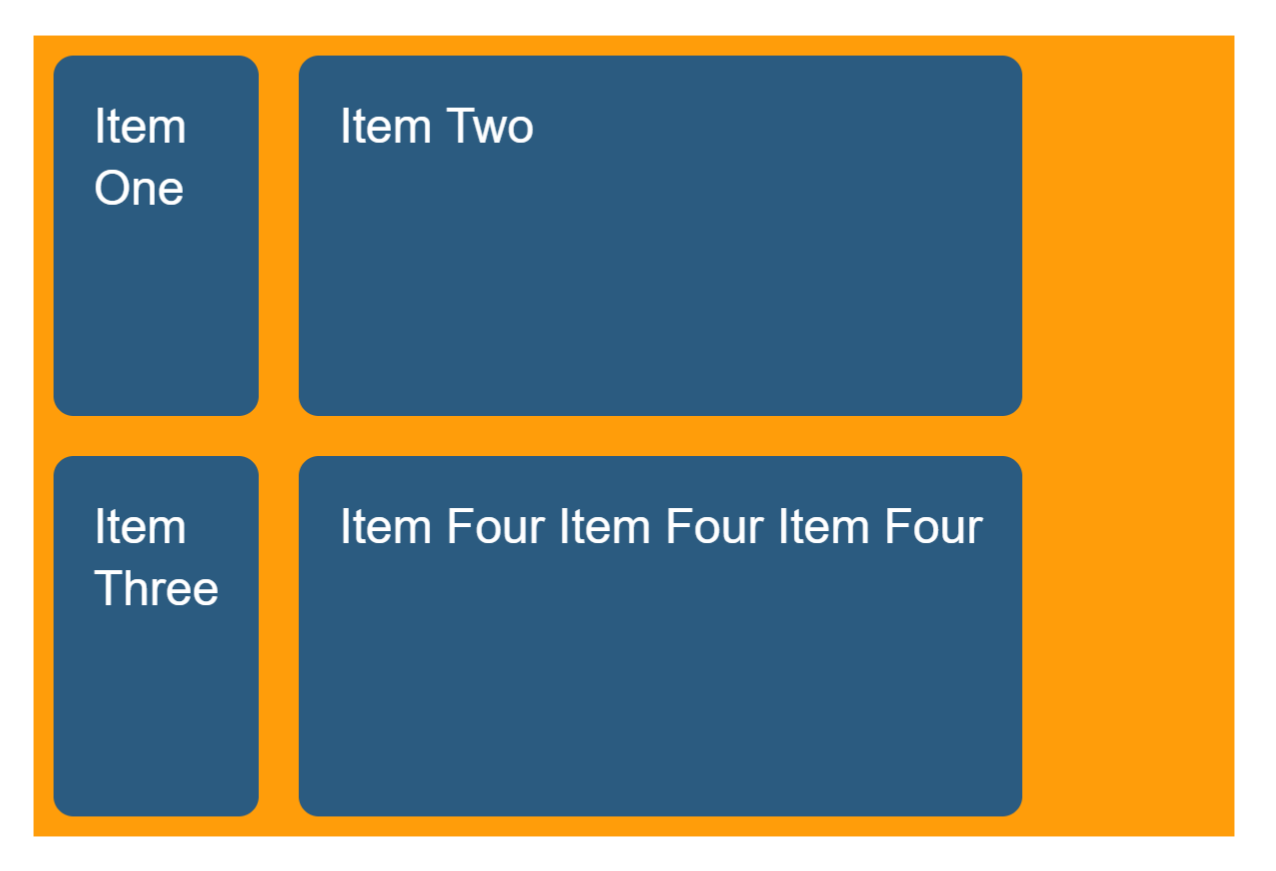
```
.example {
  box-sizing: border-box;
}
```

How big is that box?

In the past everything was a length or a percentage.

What is the **minimum** and **maximum** size of this thing?

```
.example {
  grid-template-columns: min-content max-content;
}
```



Item One

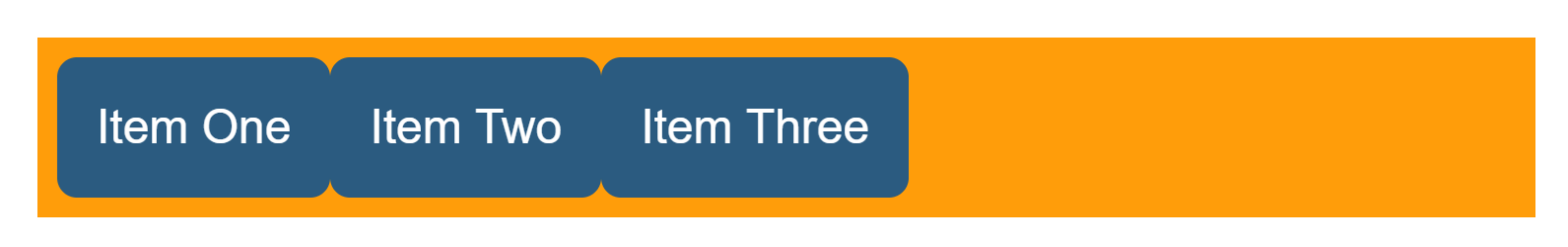
Item Two

Item Three

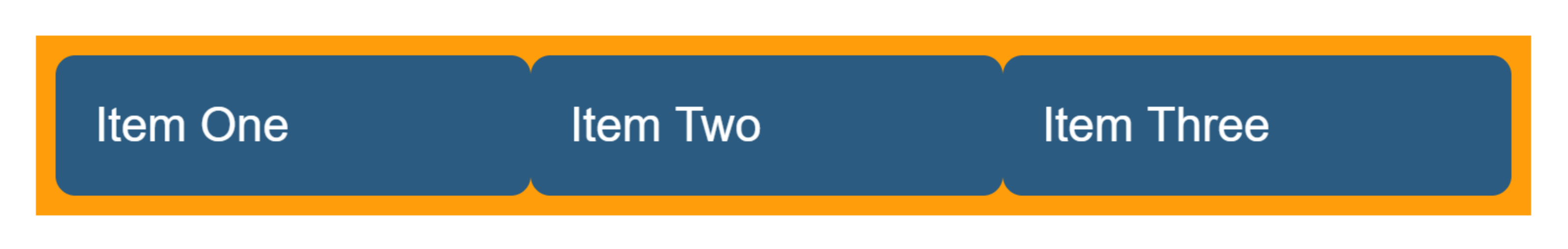
Item Four Item Four Item Four

Any content-based sizing is worked out based on these min and max content sizes.

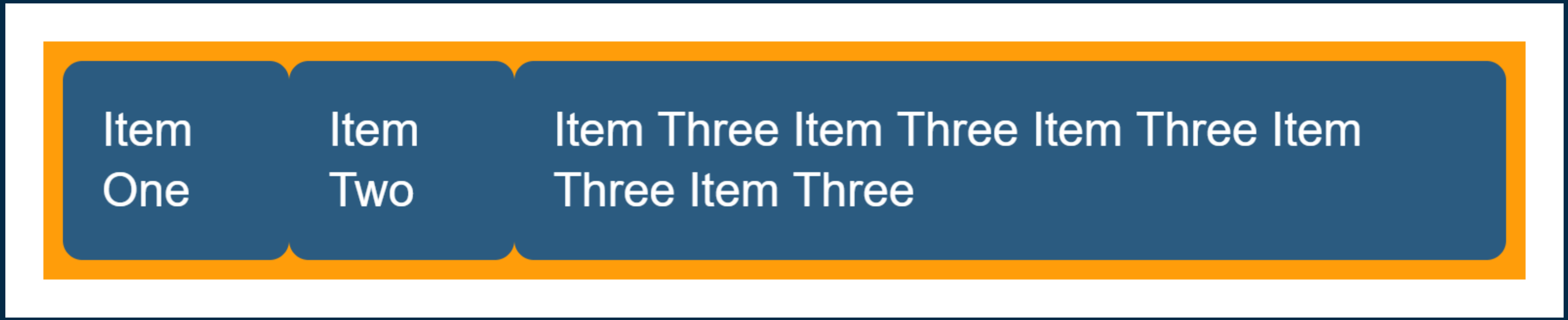
```css
.example {
  display: flex;
}
```



```
.example > * {
  flex: auto;
}
```



```
.example > * {
  flex: auto;
}
```

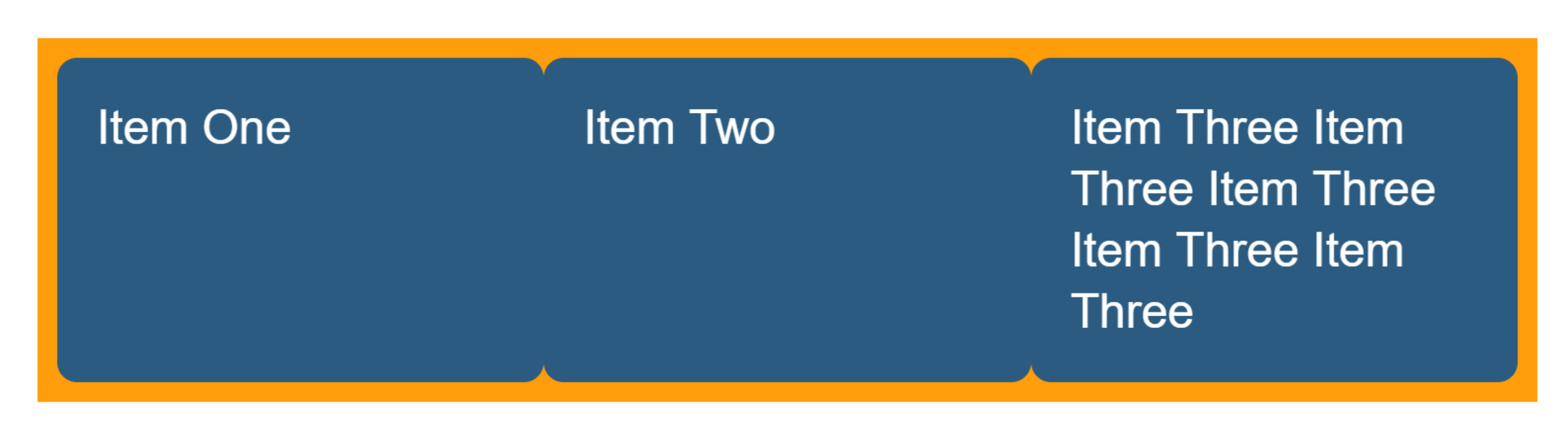


Item One

Item Two

Item Three Item Three Item Three Item Three Item Three

```
.example > * {
  flex: 1;
}
```



Item One

Item Two

Item Three Item Three Item Three Item Three Item Three

Old browsers. They exist.

We have a specification. Some of it isn’t implemented yet.

Lack of support is very different to the buggy support of the past.

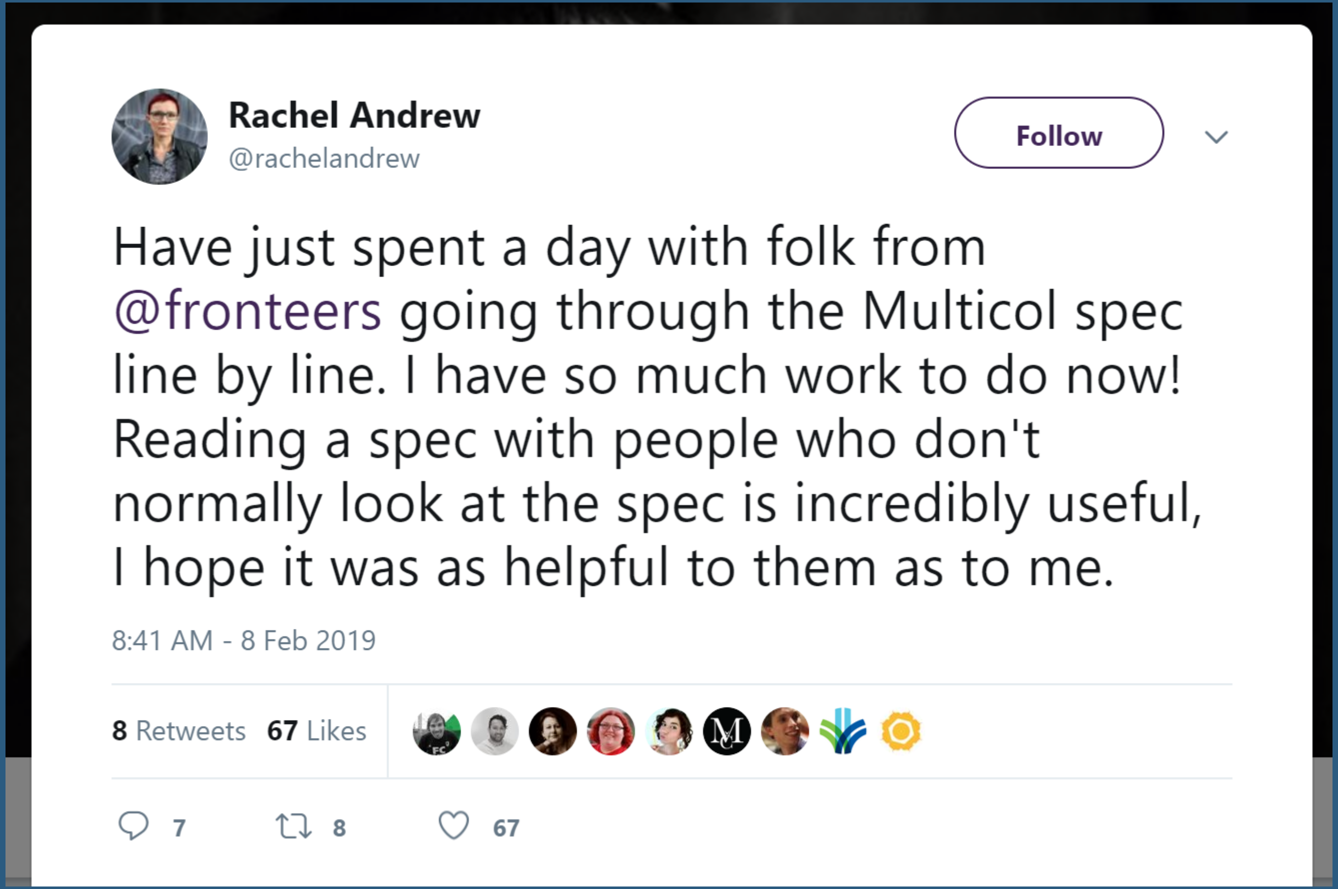
Media & Feature Queries

How big is my viewport? Is this a touchscreen? Does this browser support Grid? Respond based on the answers.

Contributing to the platform

Anyone can contribute to specifications

You don’t need to be an Invited Expert or representative.



Rachel Andrew
@rachelandrew

Have just spent a day with folk from @fronteers going through the Multicol spec line by line. I have so much work to do now! Reading a spec with people who don't normally look at the spec is incredibly useful, I hope it was as helpful to them as to me.

8:41 AM - 8 Feb 2019

8 Retweets 67 Likes

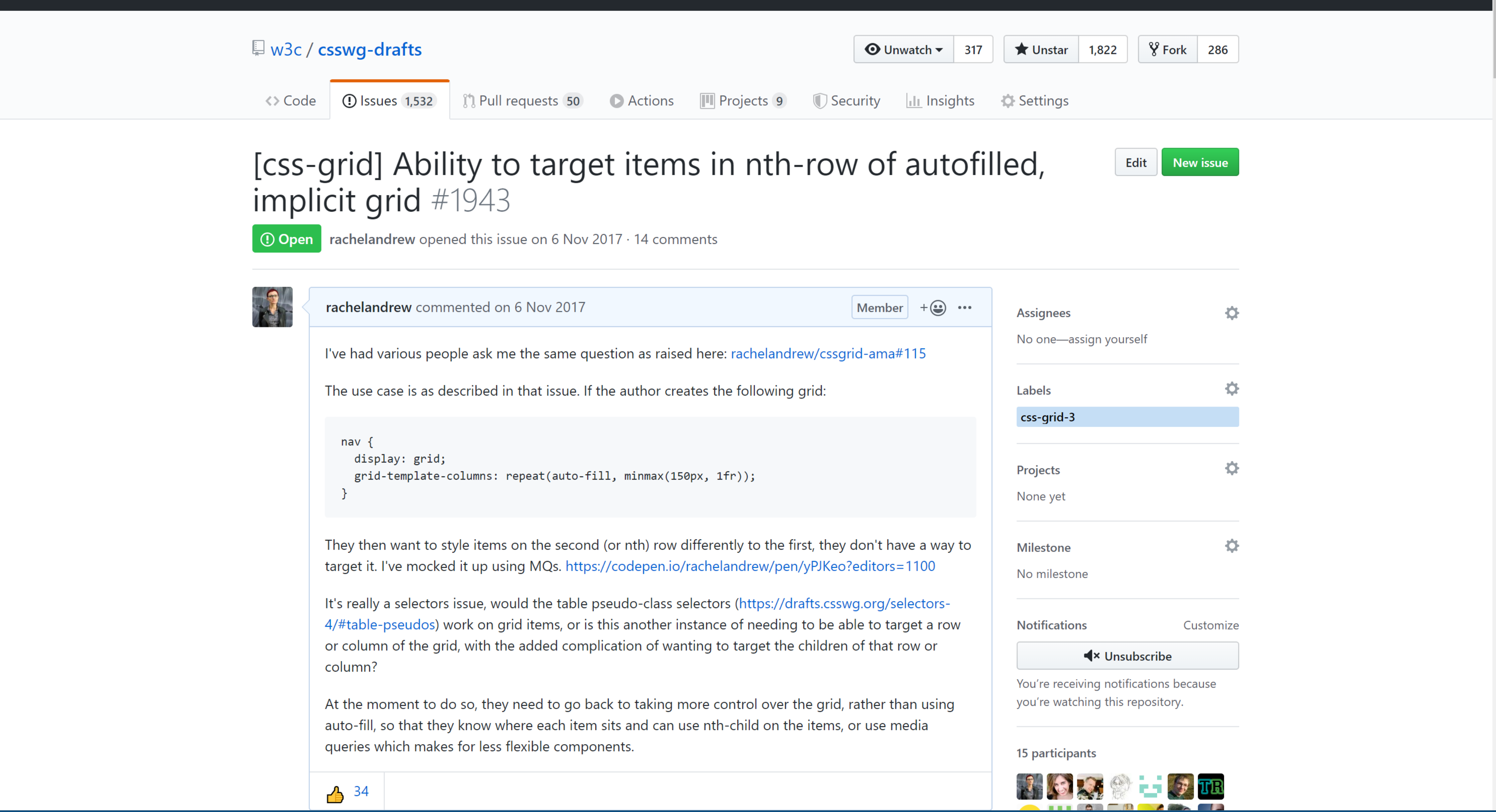
7 8 67

Everyone is allowed to contribute

You don’t need permission, or to be qualified in some way.

Read and comment on specification issues

https://github.com/w3c/csswg-drafts/issues



[css-grid] Ability to target items in nth-row of autofilled, implicit grid #1943

Open rachelandrew opened this issue on 6 Nov 2017 · 14 comments

rachelandrew commented on 6 Nov 2017

I've had various people ask me the same question as raised here: rachelandrew/cssgrid-ama#115

The use case is as described in that issue. If the author creates the following grid:

```
nav {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
}
```

They then want to style items on the second (or nth) row differently to the first, they don't have a way to target it. I've mocked it up using MQs. https://codepen.io/rachelandrew/pen/yPJKeo?editors=1100

It's really a selectors issue, would the table pseudo-class selectors (https://drafts.csswg.org/selectors-4/#table-pseudos) work on grid items, or is this another instance of needing to be able to target a row or column of the grid, with the added complication of wanting to target the children of that row or column?

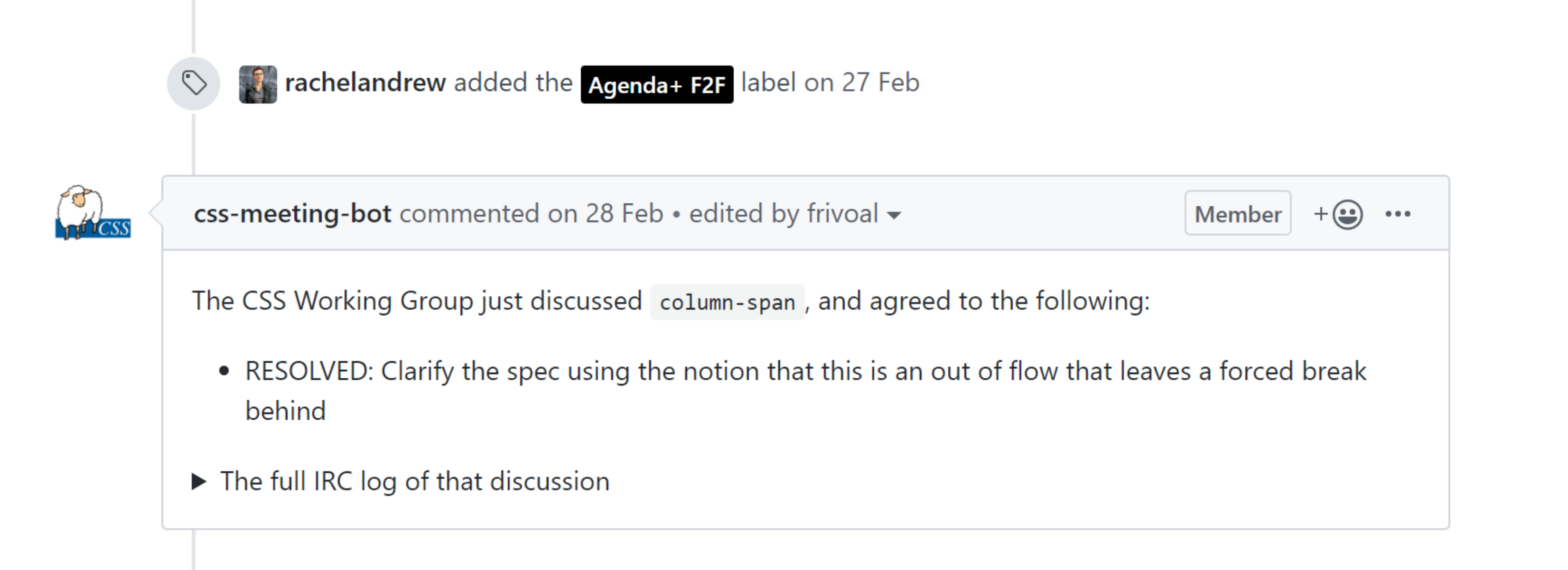
At the moment to do so, they need to go back to taking more control over the grid, rather than using auto-fill, so that they know where each item sits and can use nth-child on the items, or use media queries which makes for less flexible components.

👍 34

Show use cases

As with any software development, real use cases are valuable.

Contribute examples and diagrams to specifications



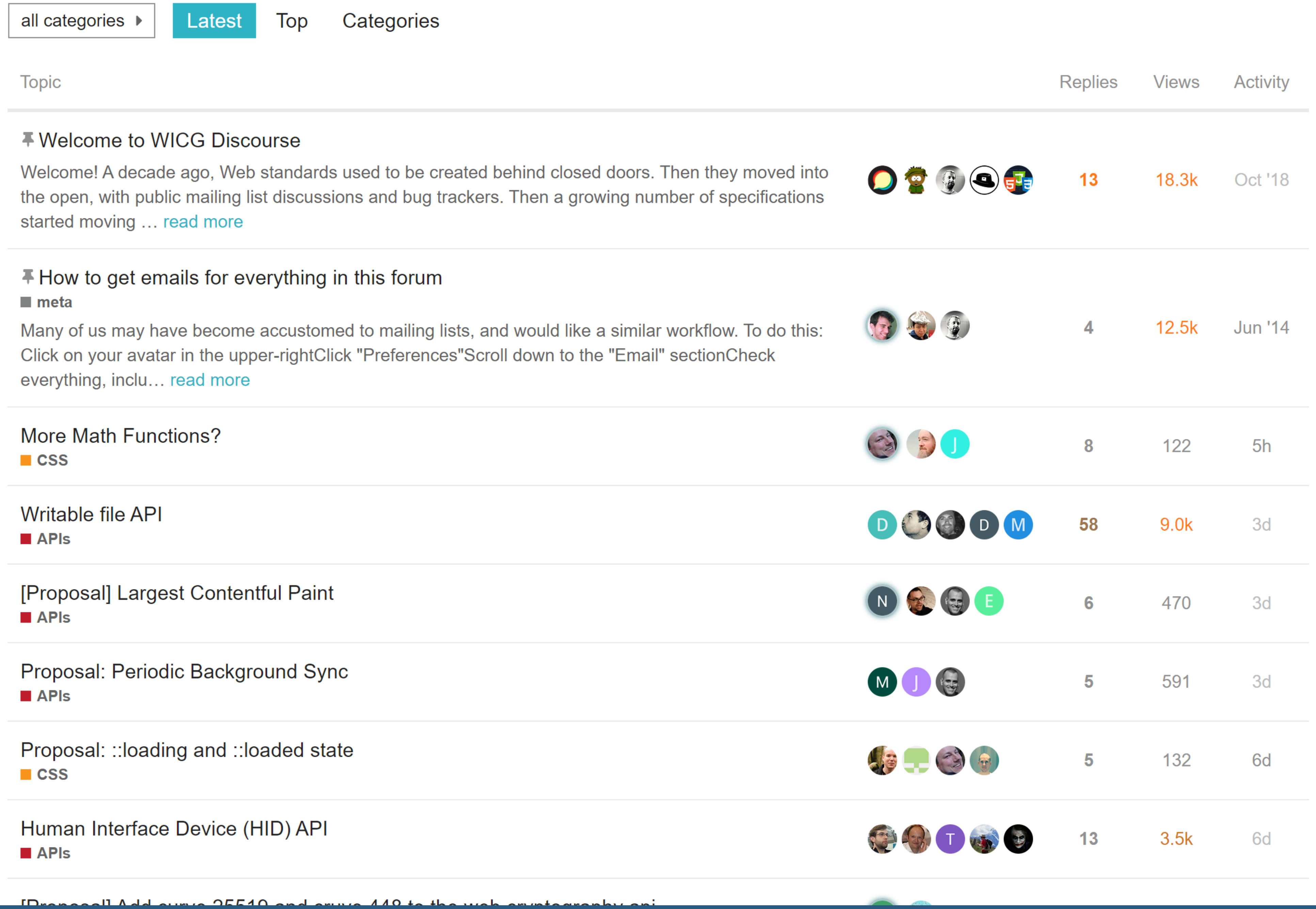
rachelandrew added the **Agenda+ F2F** label on 27 Feb

css-meeting-bot commented on 28 Feb • edited by frivoal

The CSS Working Group just discussed `column-span`, and agreed to the following:

- RESOLVED: Clarify the spec using the notion that this is an out of flow that leaves a forced break behind

▶ The full IRC log of that discussion



all categories ▸ | Latest | Top | Categories

Topic	Replies	Views	Activity
Welcome to WICG Discourse Welcome! A decade ago, Web standards used to be created behind closed doors. Then they moved into the open, with public mailing list discussions and bug trackers. Then a growing number of specifications started moving … read more	13	18.3k	Oct '18
How to get emails for everything in this forum ■ meta Many of us may have become accustomed to mailing lists, and would like a similar workflow. To do this: Click on your avatar in the upper-rightClick "Preferences"Scroll down to the "Email" sectionCheck everything, inclu… read more	4	12.5k	Jun '14
More Math Functions? ■ CSS	8	122	5h
Writable file API ■ APIs	58	9.0k	3d
[Proposal] Largest Contentful Paint ■ APIs	6	470	3d
Proposal: Periodic Background Sync ■ APIs	5	591	3d
Proposal: ::loading and ::loaded state ■ CSS	5	132	6d
Human Interface Device (HID) API ■ APIs	13	3.5k	6d
[Proposal] Add curve 25519 and curve 448 to the web cryptography api			

WICG Discourse

https://discourse.wicg.io/

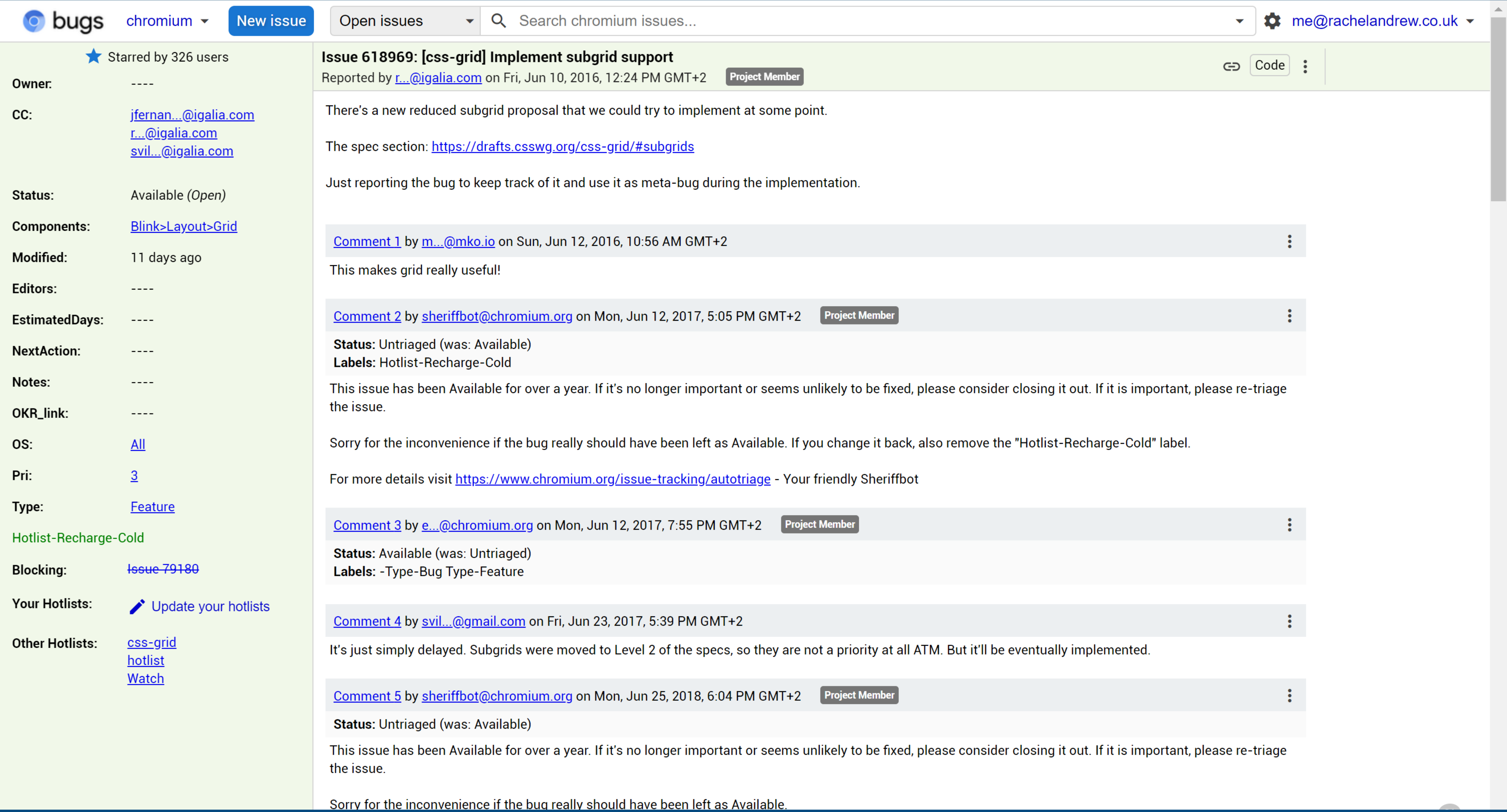
It will probably take longer than you think for changes to be made!

Patience is required for web platform contributions.

Raise browser bugs

Fix bugs, or request that features are implemented.

Learn to create a Reduced Test Case



Issue 618969: [css-grid] Implement subgrid support

Reported by r...@igalia.com on Fri, Jun 10, 2016, 12:24 PM GMT+2 Project Member

Starred by 326 users

Owner:	----
CC:	jfernan...@igalia.com r...@igalia.com svil...@igalia.com
Status:	Available *(Open)*
Components:	Blink>Layout>Grid
Modified:	11 days ago
Editors:	----
EstimatedDays:	----
NextAction:	----
Notes:	----
OKR_link:	----
OS:	All
Pri:	3
Type:	Feature
Hotlist-Recharge-Cold	
Blocking:	~~Issue 79180~~
Your Hotlists:	Update your hotlists
Other Hotlists:	css-grid hotlist Watch

There's a new reduced subgrid proposal that we could try to implement at some point.

The spec section: https://drafts.csswg.org/css-grid/#subgrids

Just reporting the bug to keep track of it and use it as meta-bug during the implementation.

Comment 1 by m...@mko.io on Sun, Jun 12, 2016, 10:56 AM GMT+2

This makes grid really useful!

Comment 2 by sheriffbot@chromium.org on Mon, Jun 12, 2017, 5:05 PM GMT+2 Project Member

Status: Untriaged (was: Available)
Labels: Hotlist-Recharge-Cold

This issue has been Available for over a year. If it's no longer important or seems unlikely to be fixed, please consider closing it out. If it is important, please re-triage the issue.

Sorry for the inconvenience if the bug really should have been left as Available. If you change it back, also remove the "Hotlist-Recharge-Cold" label.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 3 by e...@chromium.org on Mon, Jun 12, 2017, 7:55 PM GMT+2 Project Member

Status: Available (was: Untriaged)
Labels: -Type-Bug Type-Feature

Comment 4 by svil...@gmail.com on Fri, Jun 23, 2017, 5:39 PM GMT+2

It's just simply delayed. Subgrids were moved to Level 2 of the specs, so they are not a priority at all ATM. But it'll be eventually implemented.

Comment 5 by sheriffbot@chromium.org on Mon, Jun 25, 2018, 6:04 PM GMT+2 Project Member

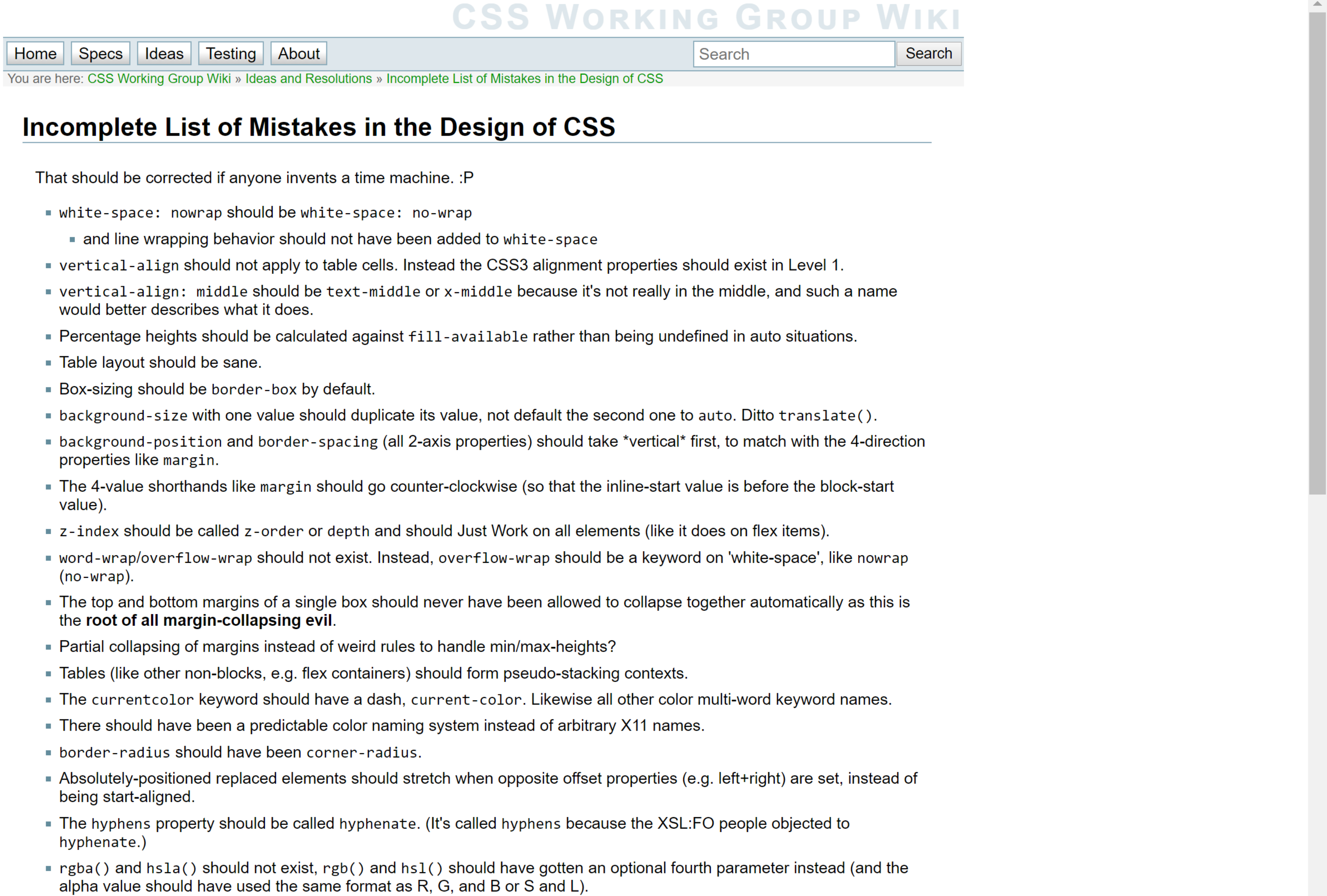
Status: Untriaged (was: Available)

This issue has been Available for over a year. If it's no longer important or seems unlikely to be fixed, please consider closing it out. If it is important, please re-triage the issue.

Sorry for the inconvenience if the bug really should have been left as Available.

Test and offer feedback on features behind flags.

This is the **best time** to make changes to a spec.



You are here: CSS Working Group Wiki » Ideas and Resolutions » Incomplete List of Mistakes in the Design of CSS

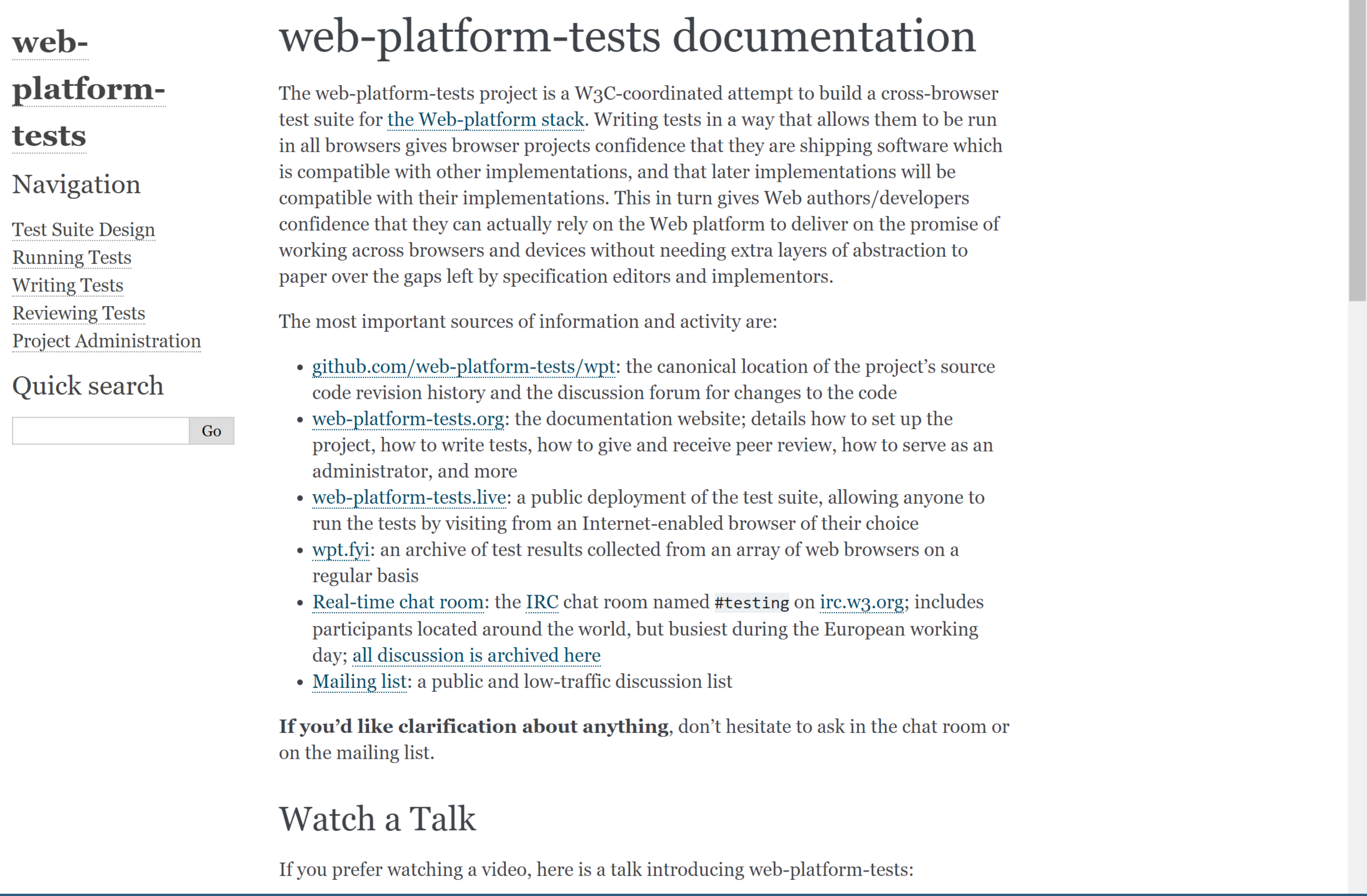
Incomplete List of Mistakes in the Design of CSS

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`.
- Absolutely-positioned replaced elements should stretch when opposite offset properties (e.g. left+right) are set, instead of being start-aligned.
- The `hyphens` property should be called `hyphenate`. (It's called `hyphens` because the XSL:FO people objected to `hyphenate`.)
- `rgba()` and `hsla()` should not exist, `rgb()` and `hsl()` should have gotten an optional fourth parameter instead (and the alpha value should have used the same format as R, G, and B or S and L).

Web Platform Tests

Help us **test** the web platform



web-platform-tests

Navigation

- Test Suite Design
- Running Tests
- Writing Tests
- Reviewing Tests
- Project Administration

Quick search

web-platform-tests documentation

The web-platform-tests project is a W3C-coordinated attempt to build a cross-browser test suite for the Web-platform stack. Writing tests in a way that allows them to be run in all browsers gives browser projects confidence that they are shipping software which is compatible with other implementations, and that later implementations will be compatible with their implementations. This in turn gives Web authors/developers confidence that they can actually rely on the Web platform to deliver on the promise of working across browsers and devices without needing extra layers of abstraction to paper over the gaps left by specification editors and implementors.

The most important sources of information and activity are:

- github.com/web-platform-tests/wpt: the canonical location of the project's source code revision history and the discussion forum for changes to the code
- web-platform-tests.org: the documentation website; details how to set up the project, how to write tests, how to give and receive peer review, how to serve as an administrator, and more
- web-platform-tests.live: a public deployment of the test suite, allowing anyone to run the tests by visiting from an Internet-enabled browser of their choice
- wpt.fyi: an archive of test results collected from an array of web browsers on a regular basis
- Real-time chat room: the IRC chat room named `#testing` on irc.w3.org; includes participants located around the world, but busiest during the European working day; all discussion is archived here
- Mailing list: a public and low-traffic discussion list

If you'd like clarification about anything, don't hesitate to ask in the chat room or on the mailing list.

Watch a Talk

If you prefer watching a video, here is a talk introducing web-platform-tests:

Many people who work on CSS started by writing tests.

Comment on spec issues, raise new issues, contribute examples, raise browser bugs, write tests.

Everything has changed.

Nothing has changed.

We need to **work together** to protect the open web platform.

Thank you!

@rachelandrew