

Modern Layout

CSS Flexbox, CSS Regions
& CSS Shapes

19 March 2014

Hello!

I'm Razvan Caliman



@razvancaliman





Stuff I work on:

CSS Regions

CSS Shapes

CSS Custom Filters

CSS *that has never seen the light of day*

polyfills, prototypes,
demos & tools

Let's talk about layout

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

[What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

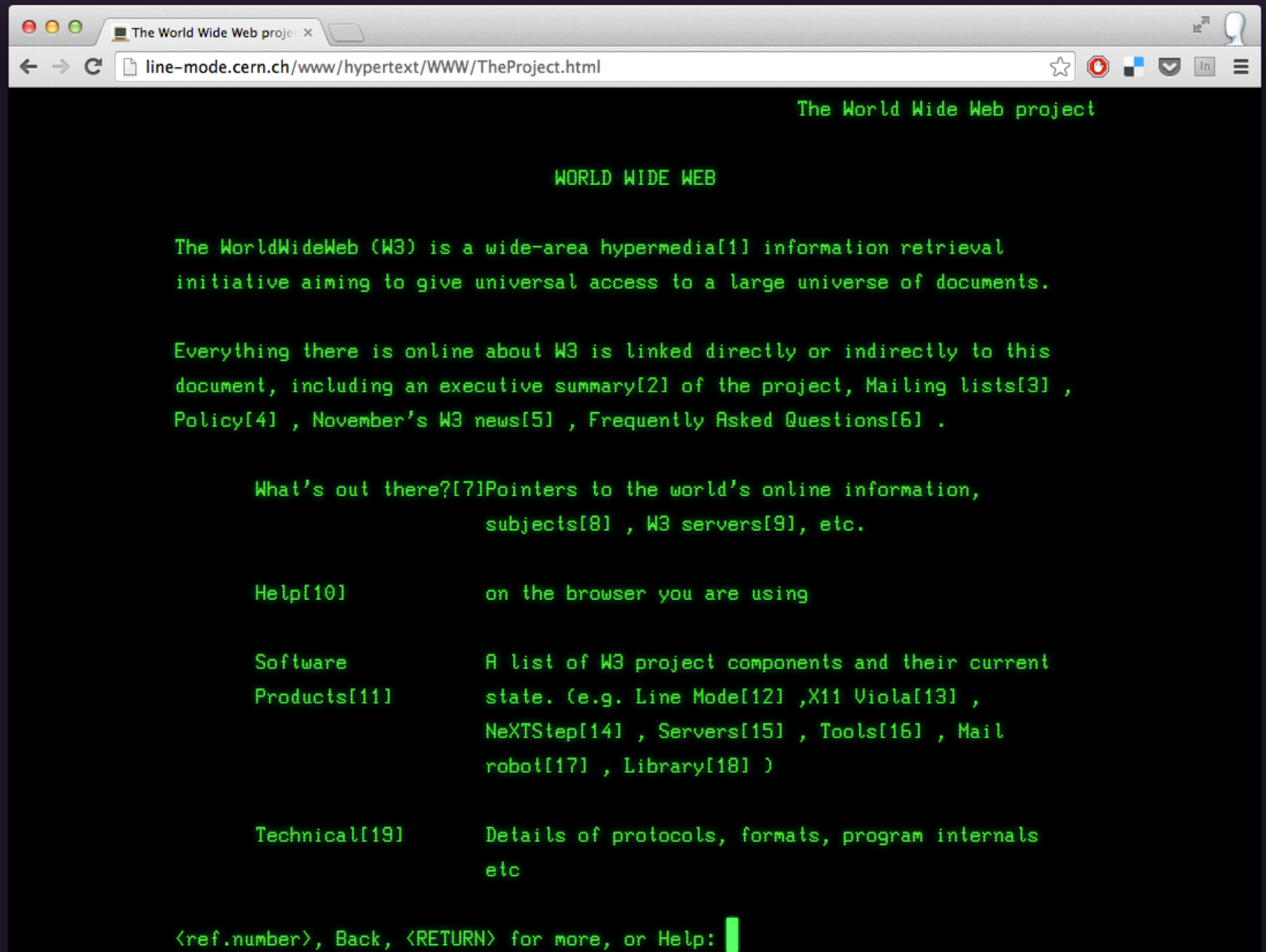
A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#) , etc.



we call them

“web pages”

The Next Web – Internation...

thenextweb.com

TNW

BLOG

MARKET

CONFERENCE

MAGAZINE

ACADEMY

Like

Follow


Follow

Search...

ALL CHANNELS


LATEST

POPULAR




Tools to Help You Proofread Emails

Promoted post




BT Launches UK Pay-to-own Service for Films, TV Box Sets Coming

just now




Google Settles its Antitrust Case with the European Commission

13 minutes ago




Tinie Tempah: Rap Demonstration

35 minutes ago



Valve's Steam Music Service Begins Rolling Out

2 hours ago



WeChat Brings Chinese New Year Traditions Into Mobile Era

About


Goodies

Advertise

Privacy

ToS


Support TNW - [become Pro](#) and we can ditch the banners.



By **EMIL PROTALINSKI** — 5 Feb '14


Moto G launches in India, available from February 6 for \$200 (8 GB) and \$225 (16 GB)

Motorola today launched its Moto G smartphone in India. While the device doesn't yet have a price at the time of writing, it will soon show up and be available... [Keep reading →](#)



5 Feb '14

37signals rebrands as Basecamp, puts all its resources into the project management tool



5 Feb '14

It's been 3 months since Google announced KitKat, and only 1.8% of Android phones have it

ANNOUNCEMENT

Join Google, Microsoft, Apple and 500+ other companies at TNW Europe 2014!

Waiting for thenextweb.com...

we actually mean

**“web
corkboards”**



<table>

semantics WTF

a pain to maintain

“spacer.gif” anyone?

position: absolute;

..come on!

float: left;

“good enough” solution

what about:

- “top”, “bottom” or “center”
- practical grids
- vertical alignment
- portability

CSS Flexbox

position, flex, order

Prefixes

best thing ever *

* if working in a perfect, sterile, purely rational world

Prefixes

worst thing ever

Flexbox evolution

2009 **display: box**

2011 **display: flexbox**

today **display: flex**

“44 years ago we put a man on the Moon,
but we still can’t vertically
center things in CSS”

Anonymous

circa August 2013

display: flex

flex-container



The diagram illustrates the 'display: flex' layout. A large dark purple rectangle represents the 'flex-container'. Inside this container, there are two smaller, medium-purple rectangles representing 'flex-item'. The first 'flex-item' on the left contains a smaller, light-blue rectangle representing a 'regular element'. The second 'flex-item' on the right is empty. The labels 'flex-container', 'flex-item', and 'regular element' are placed near their respective boxes in a light purple or white font.

flex-item

regular
element

flex-item

flex-direction

how to arrange flex-items

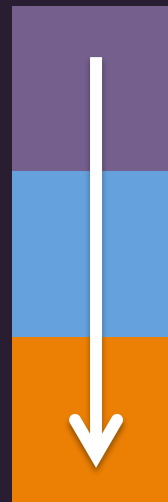


row

default



row-reverse



column

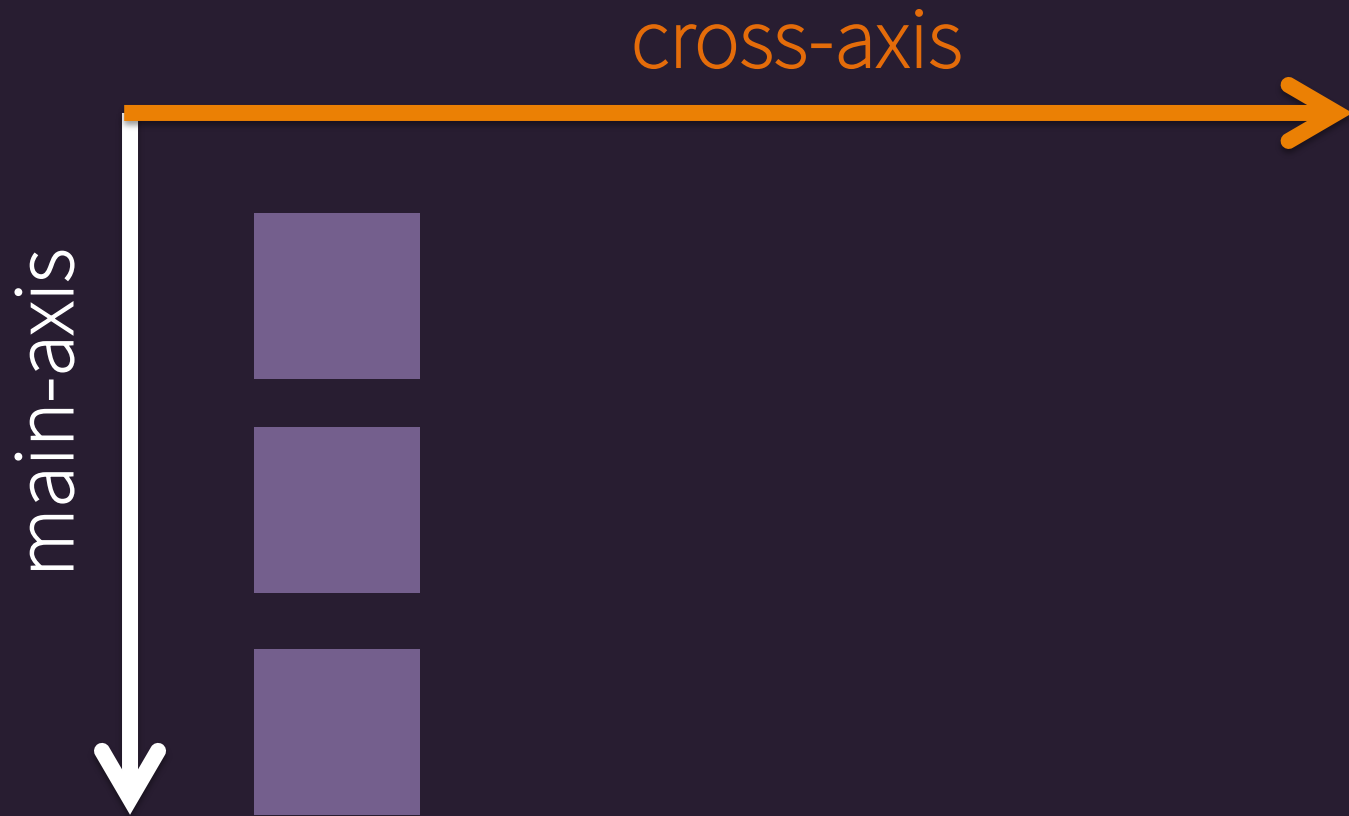


column-reverse

flex-direction: row;

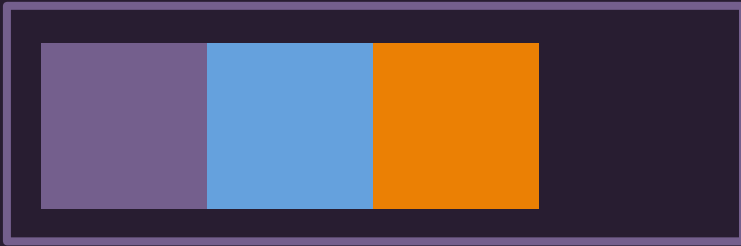


flex-direction: column;

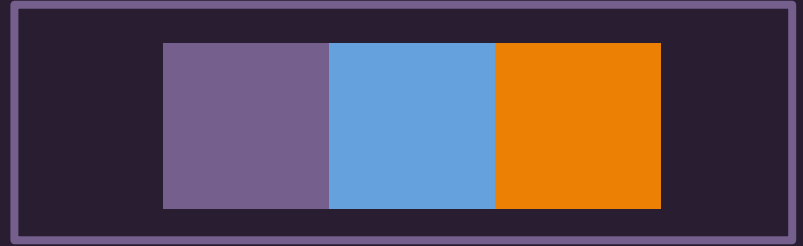


justify-content

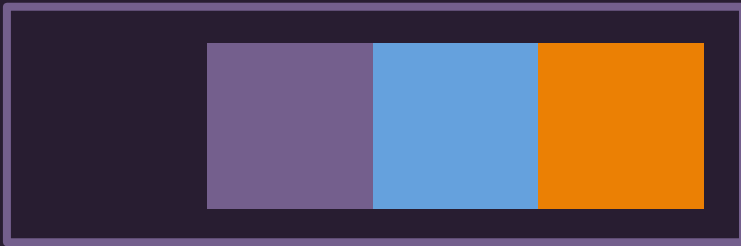
how to align along the main axis



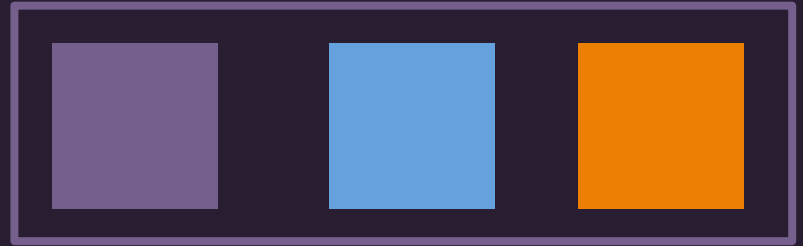
flex-start



center



flex-end



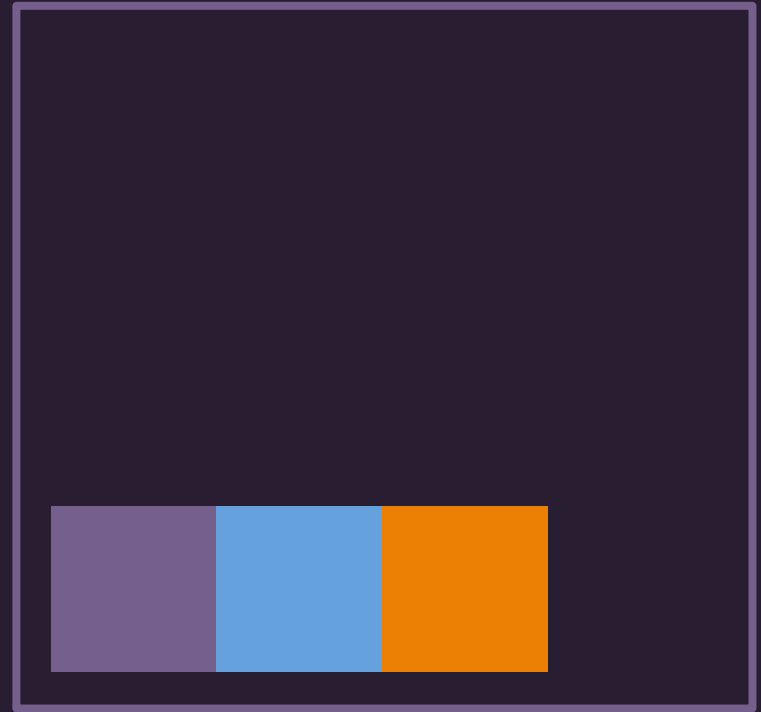
space-between

align-items

how to align flex-items along the cross-axis



flex-start



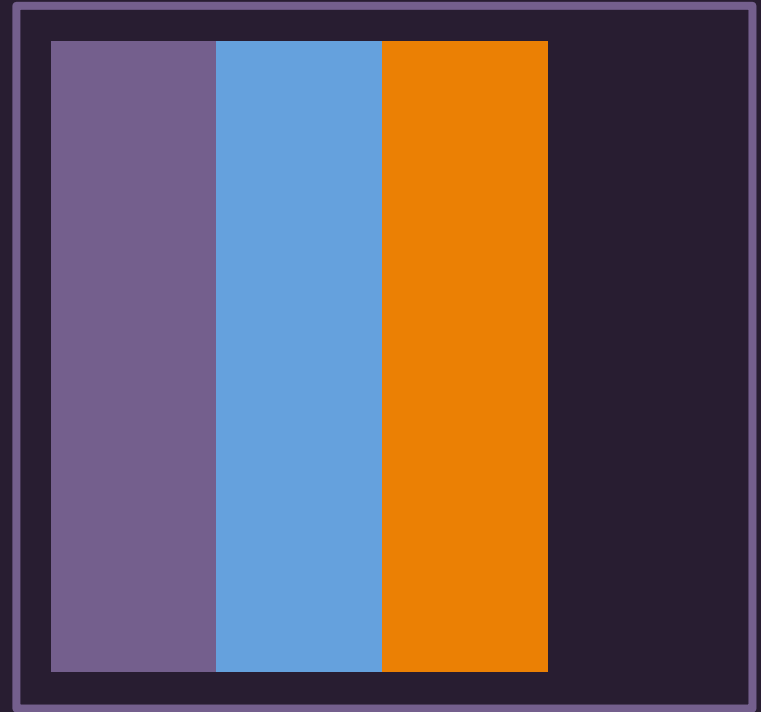
flex-end

align-items

how to align flex-items along the cross-axis



center



stretch

Mind the axes

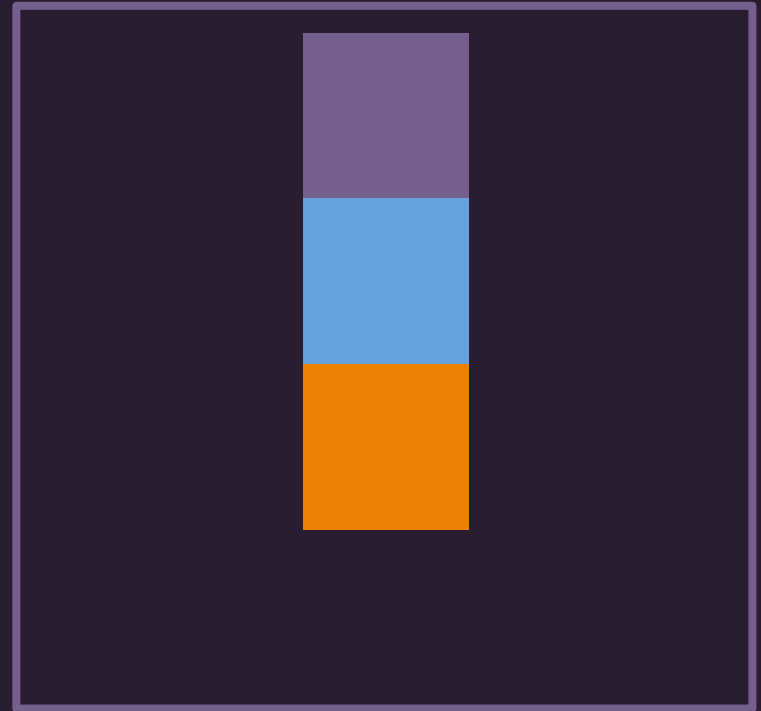
otherwise it gets confusing

```
justify-content: flex-start;  
align-items: center;
```

```
flex-direction: row;
```

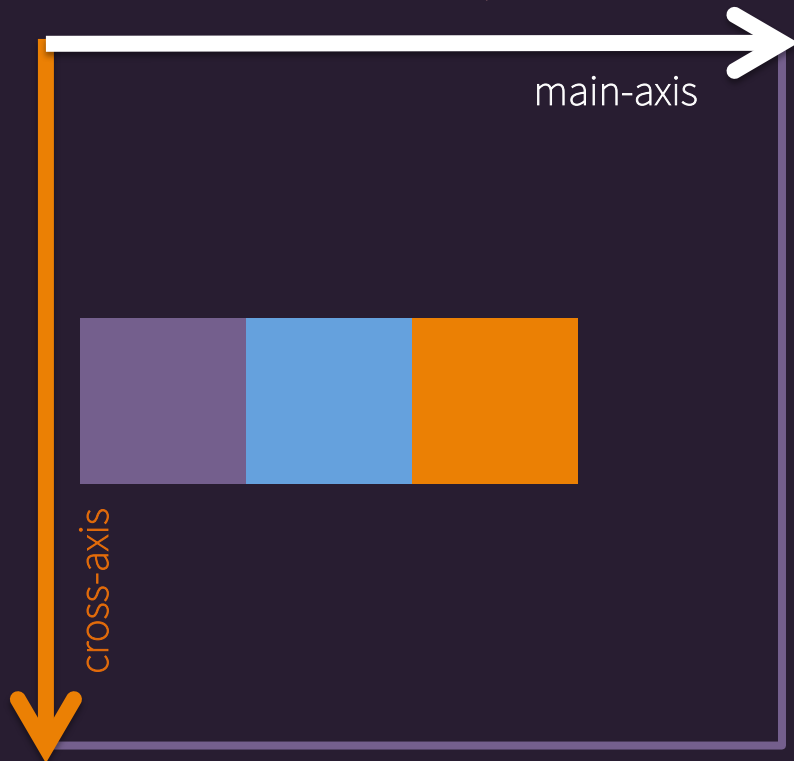


```
flex-direction: column;
```

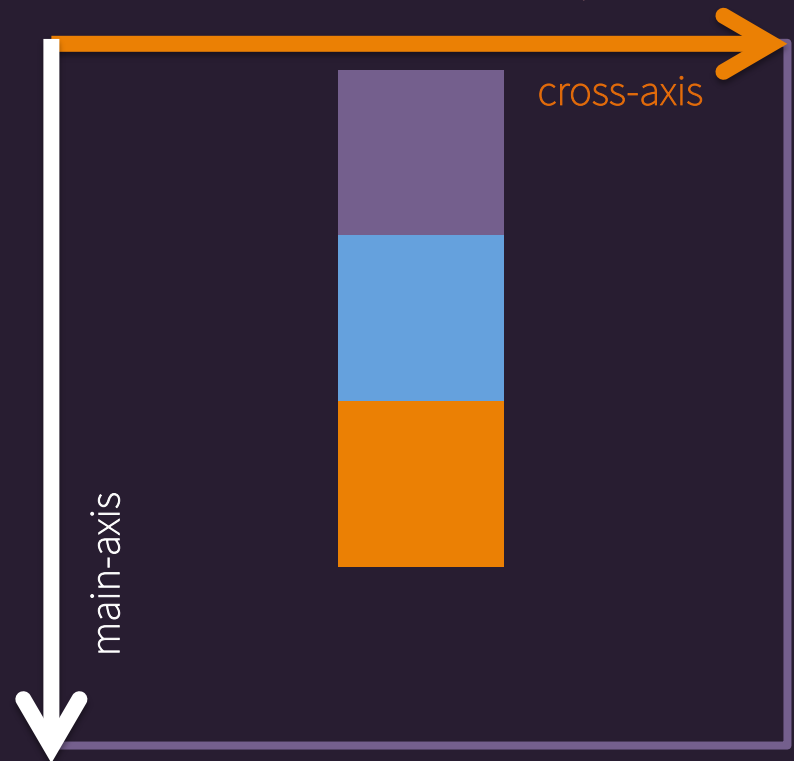


justify-content: flex-start;
align-items: center;

flex-direction: row;



flex-direction: column;



order: <integer>;

shift visual order of flex-items.

DOM order stays the same.



Flex

creating fluid layout

flex: 1 0 100px;

how much can a flex-item grow and shrink along the main-axis, and its initial size.

shorthand notation means:

flex: flex-grow flex-shrink flex-basis;

flex-grow

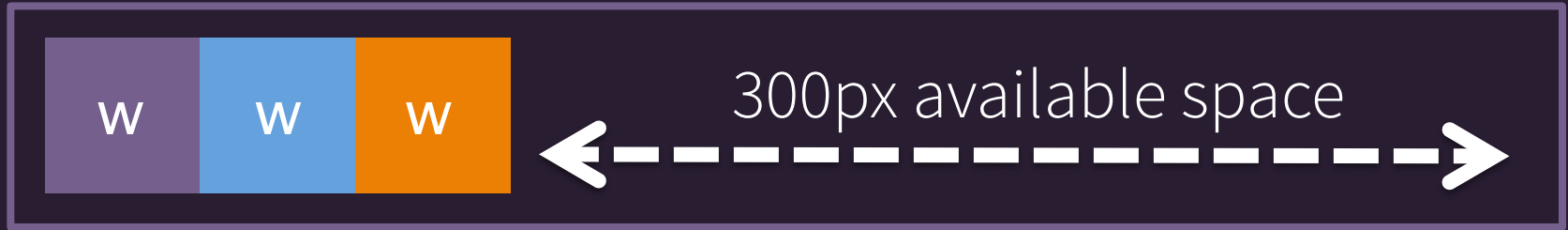
- by what proportion can a flex-item grow

flex-shrink

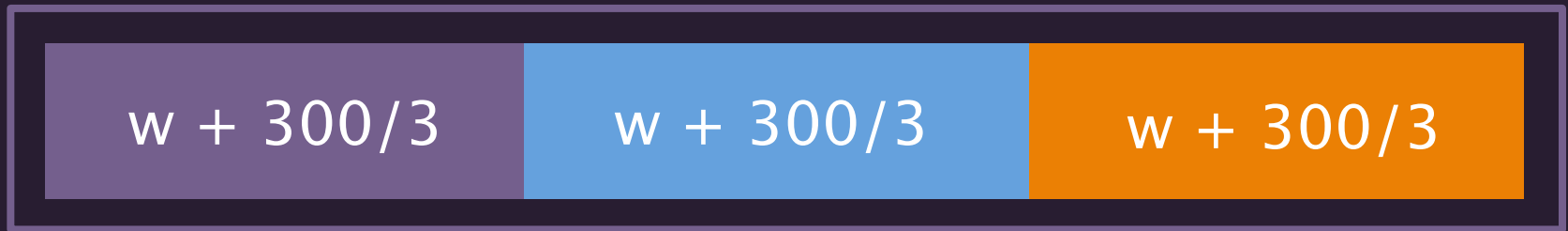
- by what proportion can a flex-item shrink

flex-basis

- initial size of flex-item, before grow or shrink



`flex-grow: 1;`



each flex-item gets an even share
of the available space



first flex-item gets twice as much space as others

flex: 1 0 100px;

- grow by one proportion of the shared available space (**flex-grow: 1**)
- do not shrink (**flex-shrink: 0**)
- get an initial size of a hundred pixels (**flex-basis: 100px**), which means **min-width** here, because it must not shrink

Careful!

```
flex: 1 0 100px;
```

when flex basis is other than zero (**0**),
flex-grow does not distribute
all unused space as you might expect.

The **flex-basis** is subtracted
before space is distributed.

Browser support

latest syntax



2011



2009



cross-browser support

```
.container {  
  display: -webkit-box;  
  display: -ms-flexbox;  
  display: -webkit-flex;  
  display: flex;  
  -webkit-box-direction: normal;  
  -webkit-box-orient: horizontal;  
  -ms-flex-direction: row;  
  -webkit-flex-direction: row;  
  flex-direction: row;  
}
```

2009

Android
Safari 3 - 6

2011

IE 10

latest syntax

Chrome
Chrome Android
Firefox
Safari 7, iOS 7
Opera



Modernizr


```
<html class="flexbox flexbox-legacy no-regions ..." >
```

```
<script src="modernizr.js"></script>
```

```
<style>
```

```
.container { ... }
```

```
.flexbox .container {
```

```
    display: -webkit-flex;
```

```
    display: flex;
```

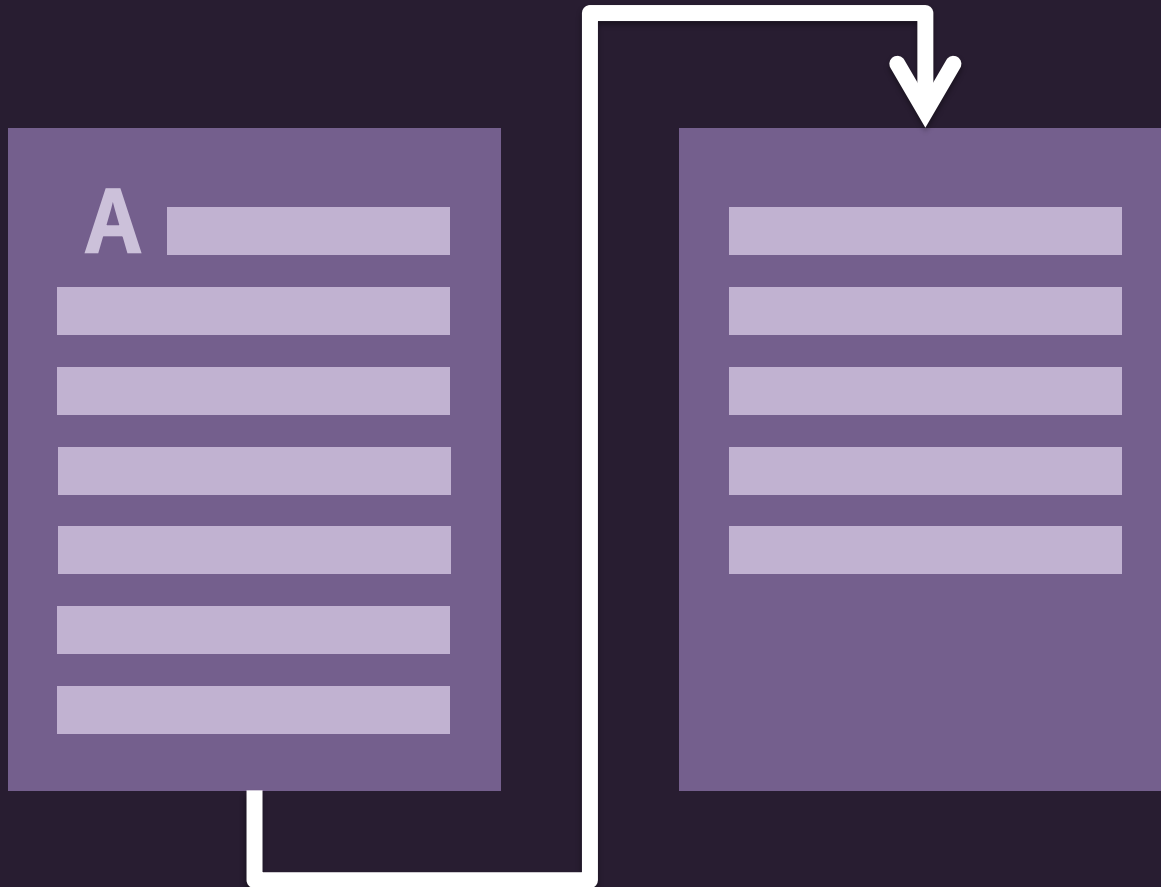
```
}
```

```
</style>
```

CSS Regions

flow content between boxes

CSS Regions



Google removed CSS Regions



Apple keeps CSS Regions



chrome://flags

CSS Regions are disabled by default in Chrome.

How to enable CSS Regions

1. go to `chrome://flags`
2. find “*Experimental Web Platform features*” and click **enable**
3. restart Chrome

-webkit-

CSS Regions are prefixed

2-step process

flow-into: myFlow;

collects content into a named flow

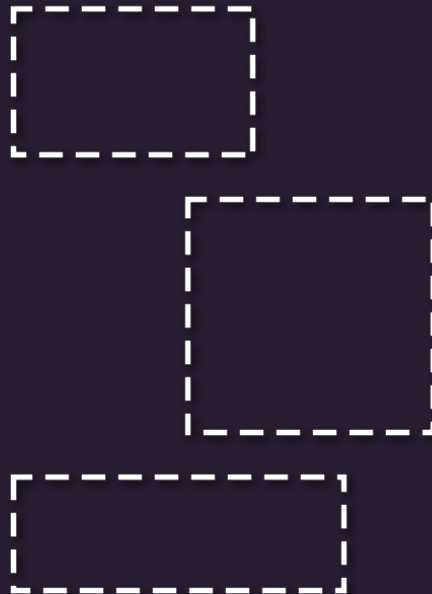
flow-from: myFlow;

renders content from a named flow

flow



regions



**flow
+
regions**



regions

are only visual containers,
they don't reorder the DOM.

good

- event handlers on content still work
- original styles for content still apply

meh

- tricky to add event handlers to a region
- limited styling of content in a specific region



A LIST APART

THE A LIST APART BLOG PRESENTS:

CSS Regions Considered Harmful

by [HÅKON WIUM LIE](#) · January 22, 2014 · 46 Comments

Share This:

[Twitter](#)

[Facebook](#)

[Google+](#)

Håkon Wium Lie is the father of CSS, the CTO of Opera, and a pioneer advocate for web standards. His [last article](#) in this magazine led directly to real fonts on the web. When Håkon speaks, whether we always agree or not, we listen. In today's post, Håkon shares his opinion on CSS Regions.

Way back, when designers first started emigrating from their islands of desktop publishing onto the web, they asked a seemingly simple question: how could they take with them their favorite colors, fonts and layouts? At first, we had no good answer. HTML offered semantic tags to describe the structure of documents, not their presentation. Browsers couldn't be told what fonts to use, or where to fetch them. Authors, however, soon found a sneaky shortcut: the `img` element. By making images of their text, authors could achieve their colorful helveticized designs. In the process, all semantics (that is, information about the meaning of elements, as opposed to the presentation) was removed and even the searchable text was gone. CSS was [proposed](#) in

CSS Regions Matter

CSS Design Featured Opinion Web January 27, 2014



SHARING

	Twitter	24
	Facebook	0
	Google +	0
	LinkedIn	0

By [Sara Soueidan](#)

I wrote this article before [the news](#) came out that Google decided to pull Regions out of Blink, which, in my opinion, is a big loss for the web community. So, even though the content of the article may go in vain, I think it's still worth sharing why I think CSS Regions mattered, and wish they weren't ditched like that. I hope you guys like it anyway.

Disclaimer: In no way is this article meant as an attack on Mr. Håkon Wium Lie's [article](#) about CSS Regions. Mr. Håkon's article served as an incentive for me to dig deeper into CSS Regions, and during the process, I learned a lot about them, and about other future

WEB & MOBILE DEVELOPMENT NEWS

Best of JavaScript, HTML & CSS – Week of January 27, 2014

Web Standards Library Update – Week of January 20, 2014

Best of JavaScript, HTML & CSS – Week of January 20, 2014

Web Standards Library Update – Week of January 13, 2014

Best of JavaScript, HTML & CSS – Week of January 13, 2014

CATEGORIES

- Best of
- CSS
- Design
- Featured
- Firefox OS
- Game Development
- HTML
- JavaScript
- Mobile
- News
- Node JS

Myth

Regions require empty placeholder `<div>s`

- any block-level item can become a region
- `::before`, `::after`
- upcoming block-generating specs
like CSS Grids, `overflow: fragment`

Myth

Regions are not responsive

- regions work with every layout model
- regions can auto-size to their content
- not a replacement for CSS Multi-column

Myth

Regions are a bad layout model

- regions are not a layout model, such as Flexbox or Multi-col
- regions are a fragmentation model
- building blocks

CSS Regions Object Model

JavaScript API for flows and regions

```
document.getNamedFlows()['myFlow']
```

```
myFlow.getRegions()
```

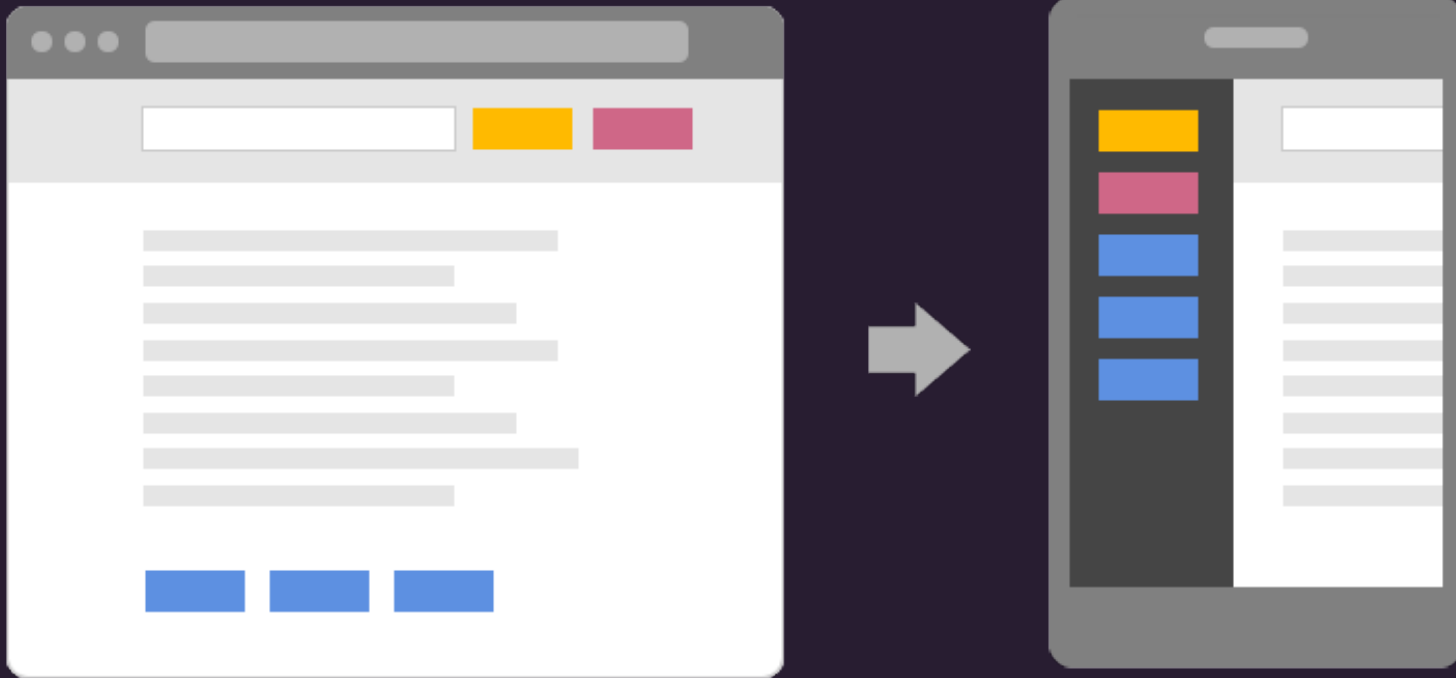
```
myFlow.getContent()
```

```
myFlow.overset
```

Responsive Menu with CSS Regions

<http://cdpn.io/tdHEg>

Adaptive UI with CSS Regions



Browser support

enabled by default



behind chrome://flags



older implementation



Polyfill

github.com/FremyCompany/css-regions-polyfill



9+



post-Presto

CSS Shapes

wrap content inside and
around custom shapes

The Shapes of CSS

All of the below use only a single HTML element. Any kind of CSS goes, as long as it's supported in at least one browser.

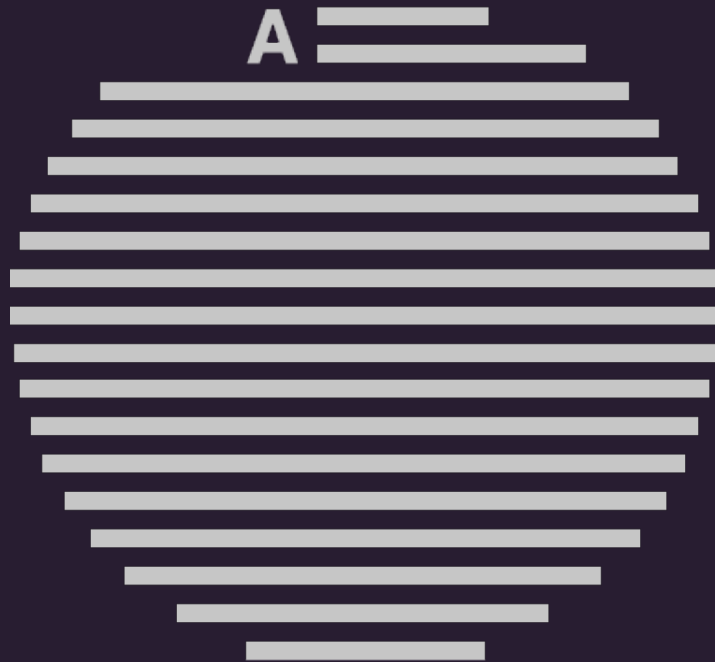
Triangle Up



```
#triangle-up {  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 100px solid red;  
}
```


shape-inside

wrap content inside of a custom shape



shape-outside

wrap content around a custom shape



How to enable CSS Shapes

1. go to `chrome://flags`
2. find “*Experimental Web Platform features*” and click **enable**
3. restart Chrome

Browser support

behind `chrome://flags`



Resources for CSS Shapes

[Creating Non-Rectangular Layouts
with CSS Shapes](#)

[CSS Shapes on html.adobe.com](#)

[Alice in Wonderland demo
with CSS Shapes](#)

CSS Flexbox

position, align, flex and order elements

CSS Regions

control flow of content

CSS Shapes

wrap content inside/around shapes

Experiment!



@razvancaliman