BASTA!

Timo Korinth | MAXIMAGO

Flexbox

CSS Layouting der Zukunft

Special Day "**Modern Business Applications**"

Thema	Sprecher	Datum, Uhrzeit	Raum
TypeScript für .NET-Entwickler	Christian Wenz	DI, 20. September 2016, 10.00 bis 11.15	Zagreb B
Angular 2: Komponentenbasierte HTML5-Anwendungen	Christian Weyer	DI, 20. September 2016, 11.30 bis 12.45	Zagreb B
Flexbox – CSS Layouting der Zukunft	Timo Korinth	DI, 20. September 2016, 14.15 bis 15.30	Zagreb B
Leichtgewichtige Architekturen mit ASP.NET Web API und SignalR	Christian Weyer	DI, 20. September 2016, 16.15 bis 17.30	Zagreb B
Mobile Apps und Zugriff auf Unternehmensdaten – ohne Cloud-Speicher und VPN	Christian Liebel	DI, 20. September 2016, 18.30 bis 19.45	Zagreb B

TIMO KORINTH

MAXIMAGO

EXPERIENCE MATTERS

- Director Technology
  @ MAXIMAGO
- UI-Developer
  WPF, HTML5, AngularJS, Silverlight
- Speaker, Author

Timo Stönner

Benutzerorientierte Software-Entwicklung

Fallstudie zur Entwicklung einer Datenvisualisierungssoftware

THEMEN

- ❖ Warum?
- ❖ Flexbox
- ❖ Beispiele
- ❖ Best practices

Warum?

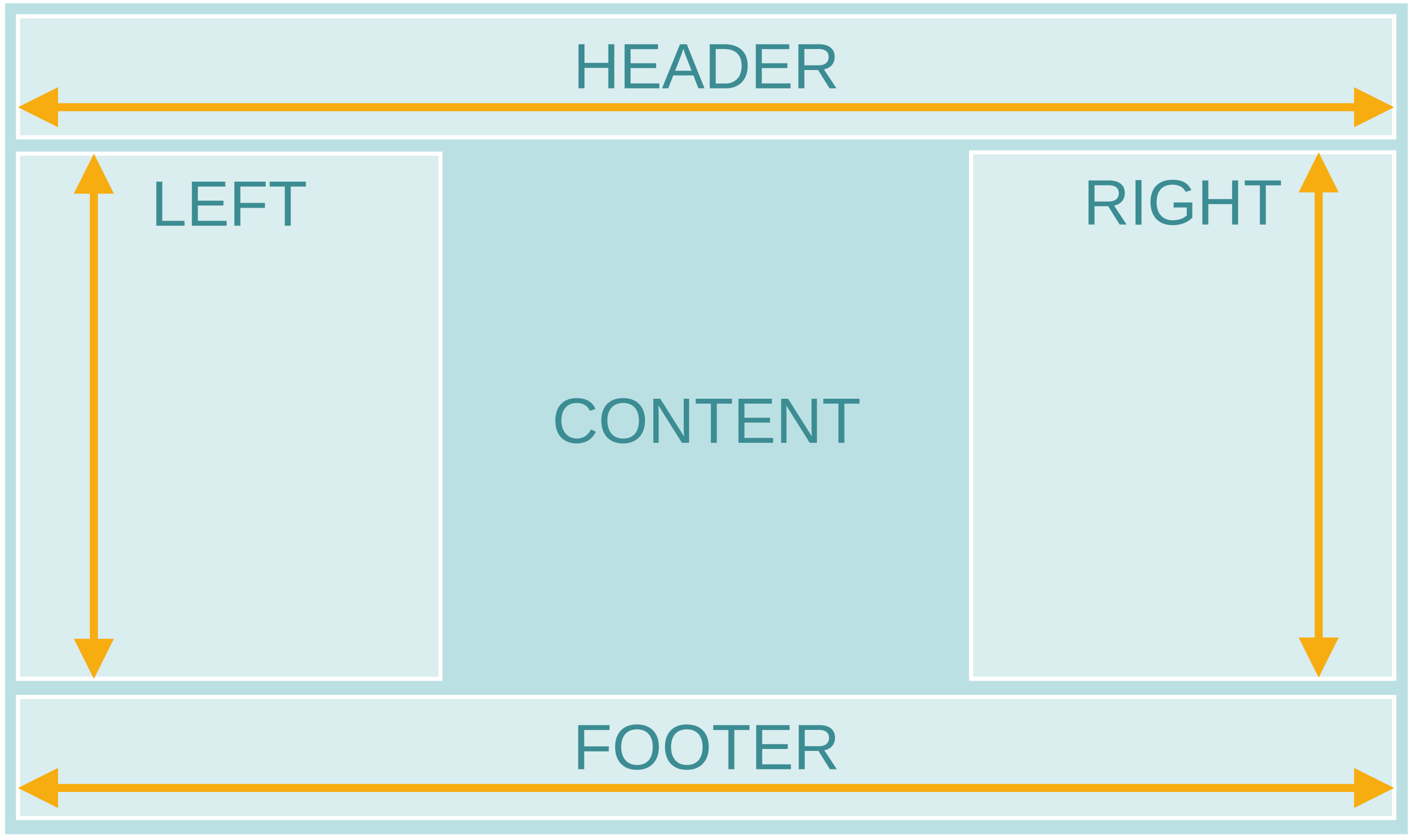
CENTER

intriguing.com

Der heilige Gral

(Applikations-Layout)

- Volle Breite
- Gleich hoch
- Code Reihenfolge
- So wenig Code wie möglich



Demo

Applikations-Layouts

- Meist genutzt:
  - Block & Float (z.B. auch von Bootstrap)
  - Absolute Positionierung
  - Feste Größen
  - JavaScript / jQuery
  - CSS Hacks / Tricks (eigener Studiengang)

Flexbox

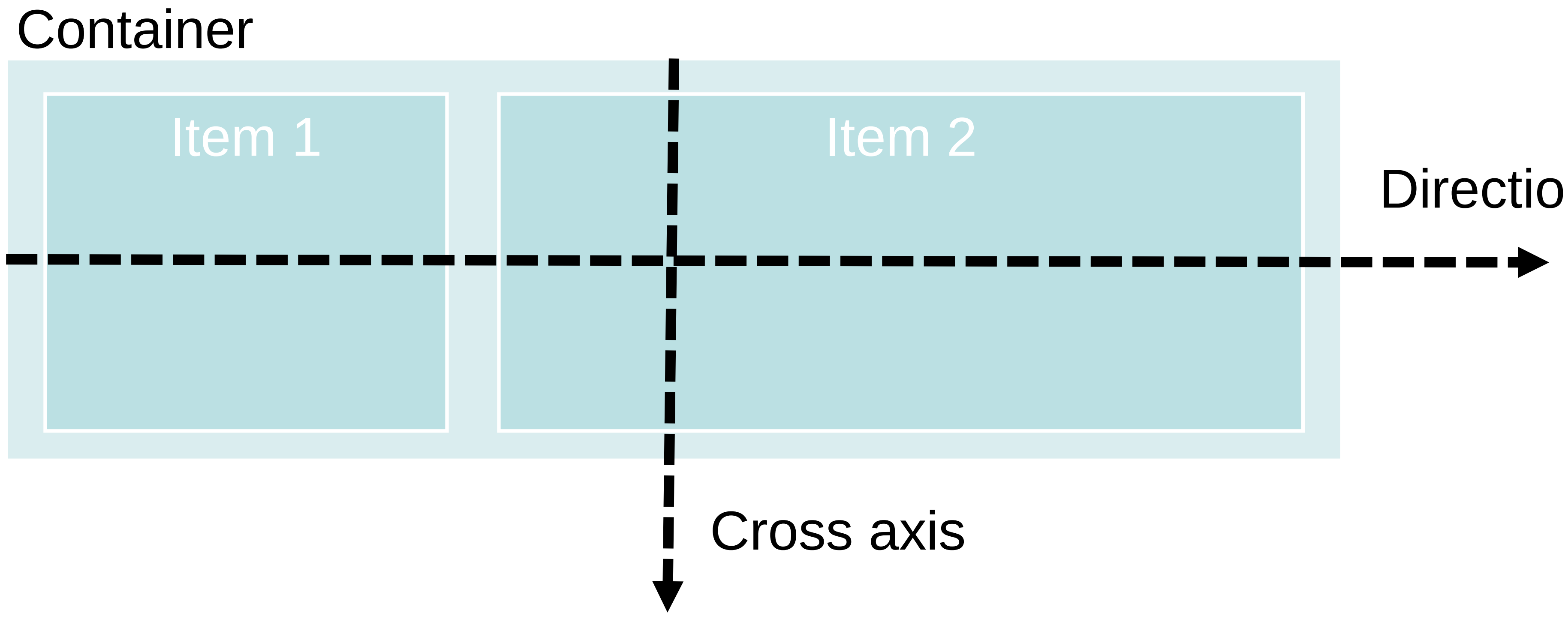
Flexbox

The Flexbox Layout (Flexible Box) module (currently a W3C Last Call Working Draft) aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex").

Flexbox

The Flexbox Layout (Flexible Box) module (currently a W3C Last Call Working Draft) aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex").

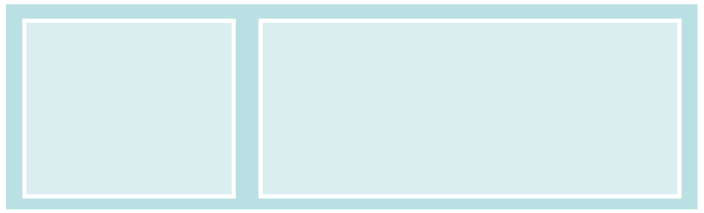
Flexbox



Flexbox

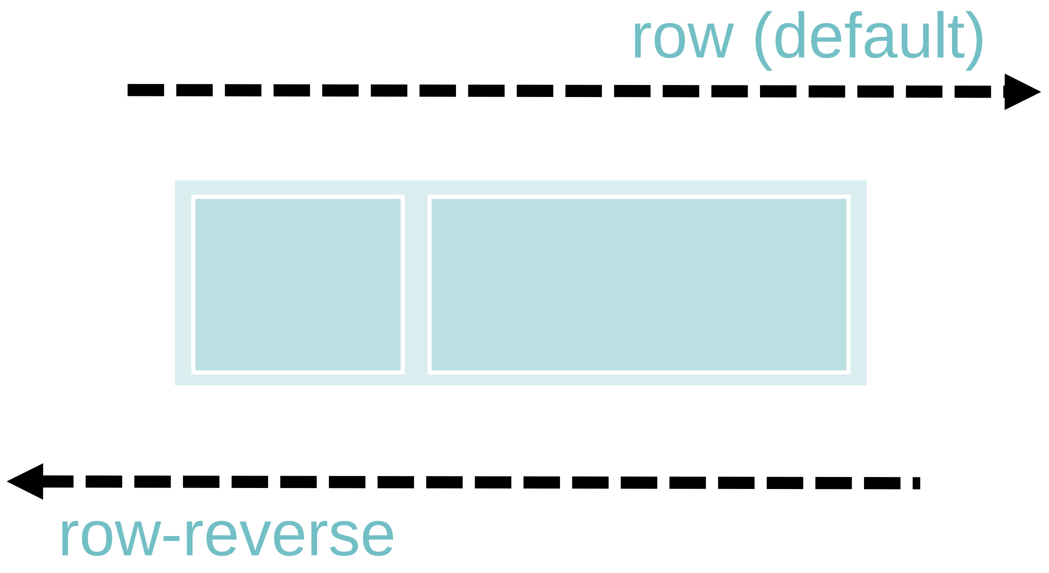
```
display: flex;
```

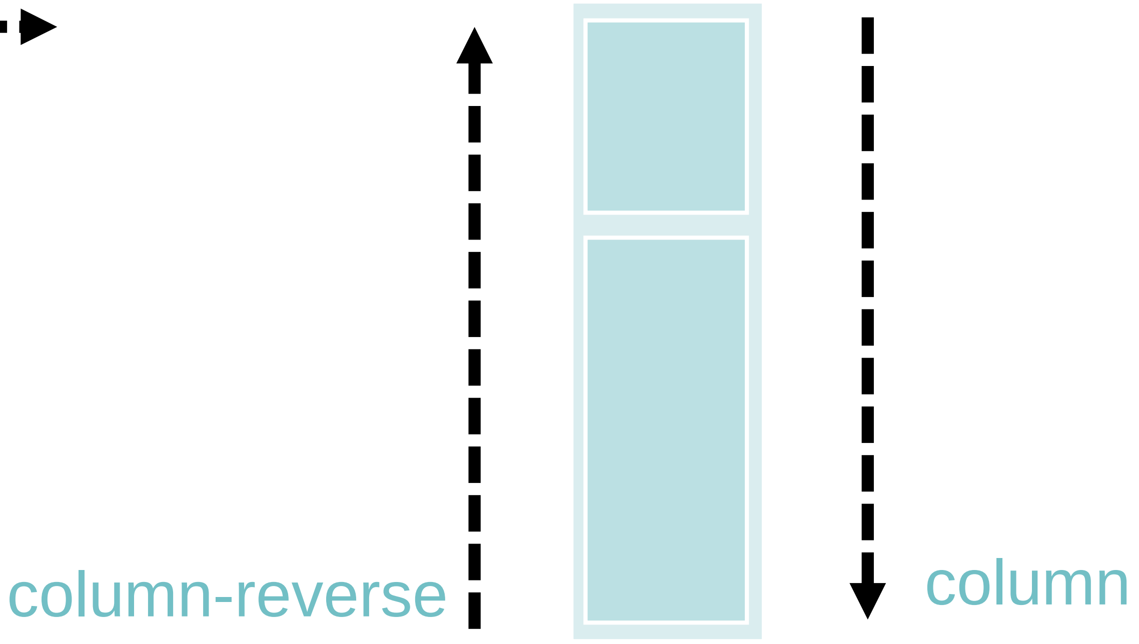
Container



Flexbox

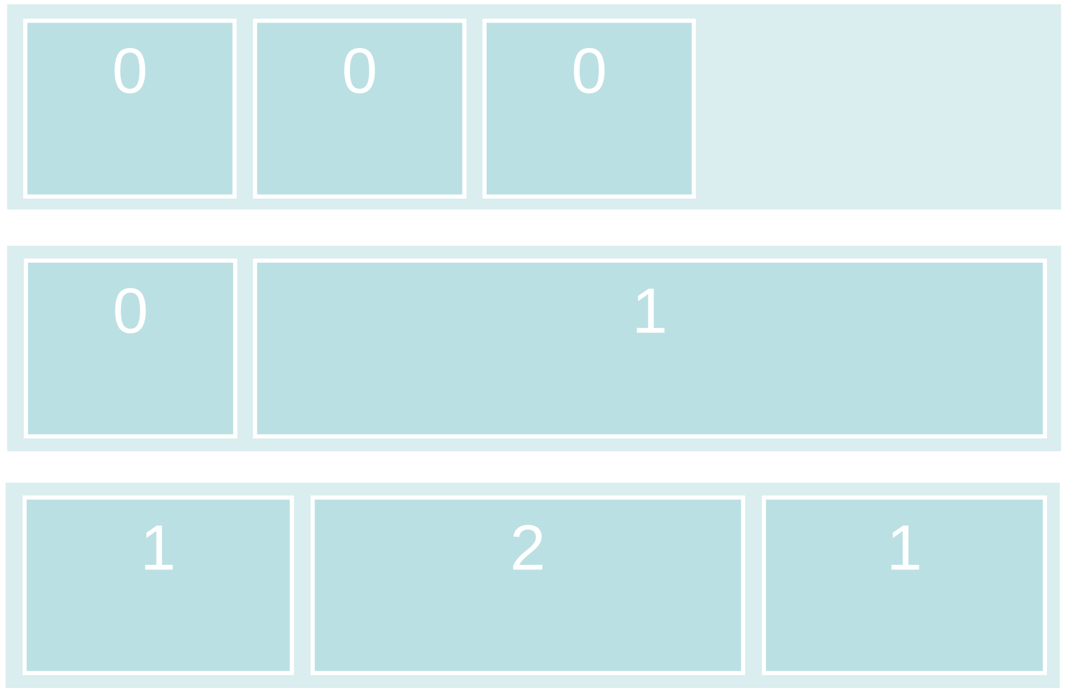
flex-direction: row | row-reverse | column | column-reverse;





Flexbox

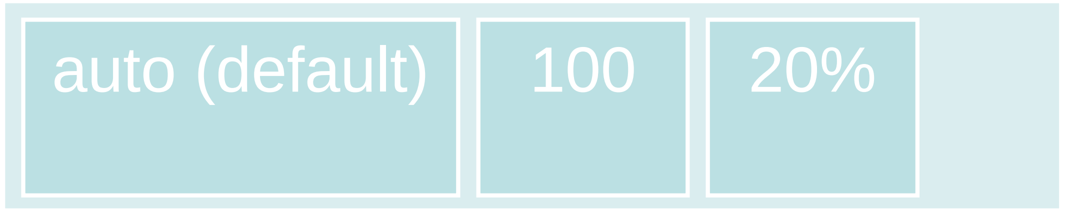
flex-grow: <number>;



0 (default)

Flexbox

flex-basis: <length> | auto;



Flexbox

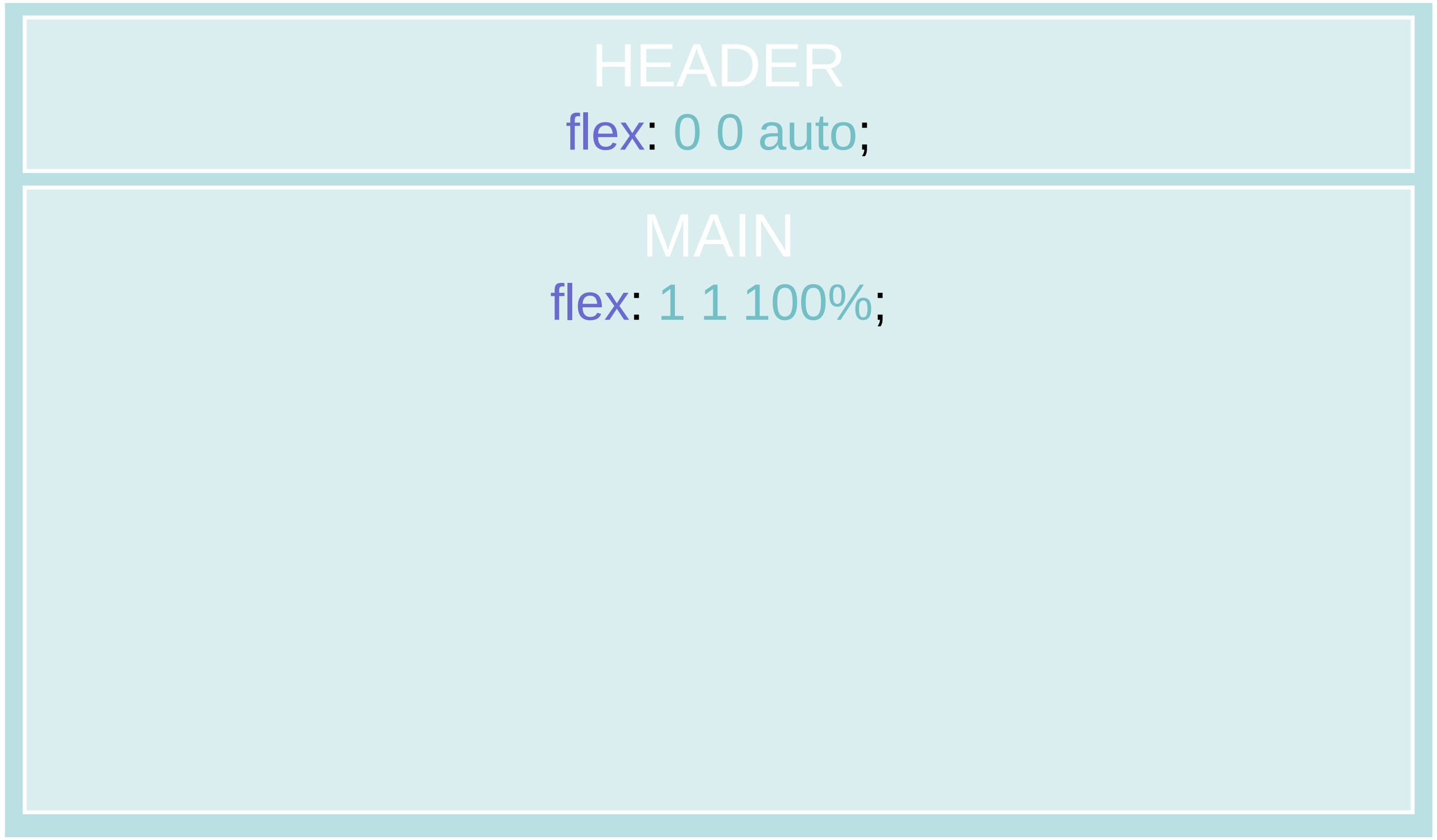
```
flex-shrink: <number>;        1 (default)
```

Kurzschreibweise:

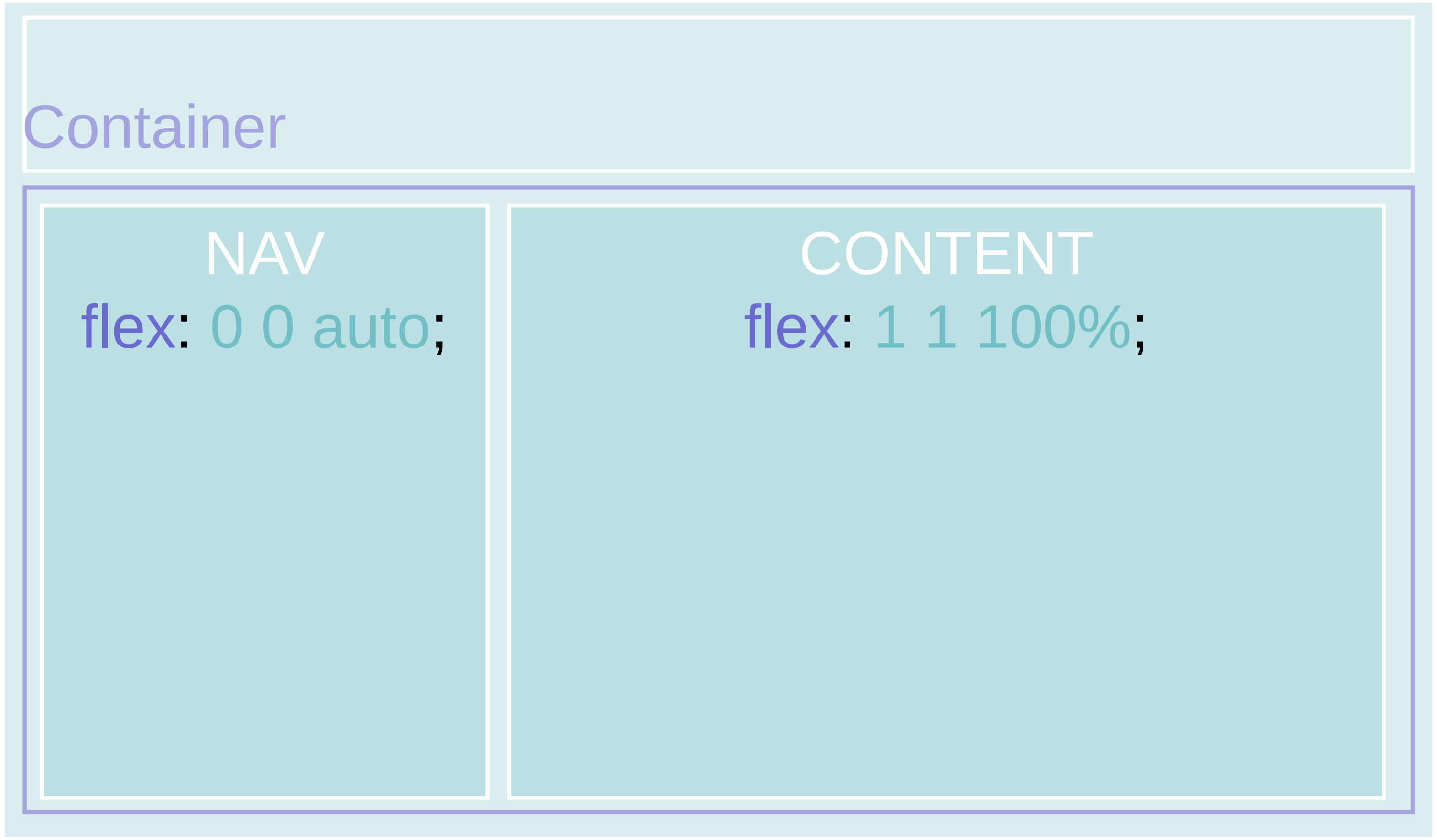
```
flex: <grow> <shrink> <basis>;
```

Flexbox

Container `flex-direction: column;`

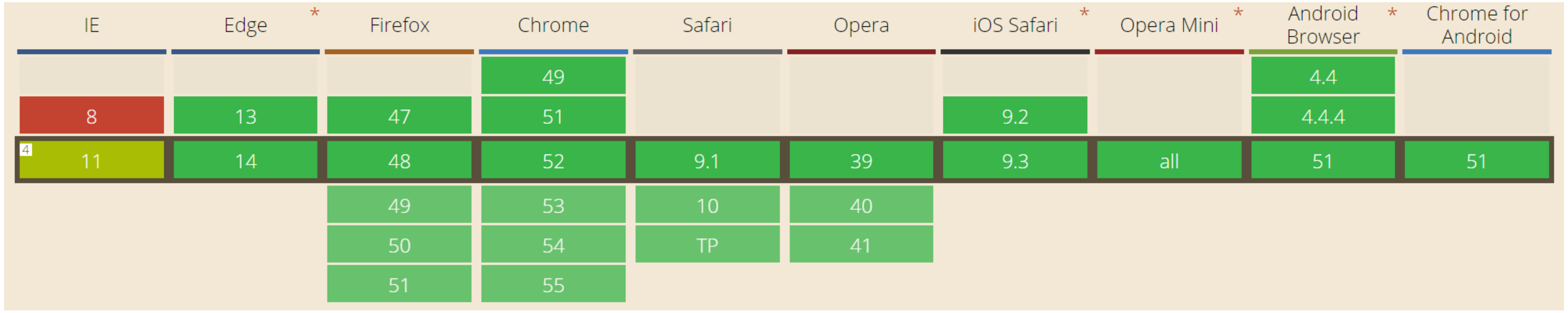


Flexbox



Flexbox

- Browser Support



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49					4.4	
8	13	47	51			9.2		4.4.4	
[4] 11	14	48	52	9.1	39	9.3	all	51	51
		49	53	10	40				
		50	54	TP	41				
		51	55						

http://caniuse.com/#feat=flexbox

Browser Prefixe

```
display:  flex;
display: -webkit-box;
display: -webkit-flex;
display: -ms-flexbox;
```

Ist eigentlich …

Mixins / Autoprefixer

```
.flexDisplay {
    display: flex;
    display: -webkit-box;
    display: -webkit-flex;
    display: -ms-flexbox;
}
```

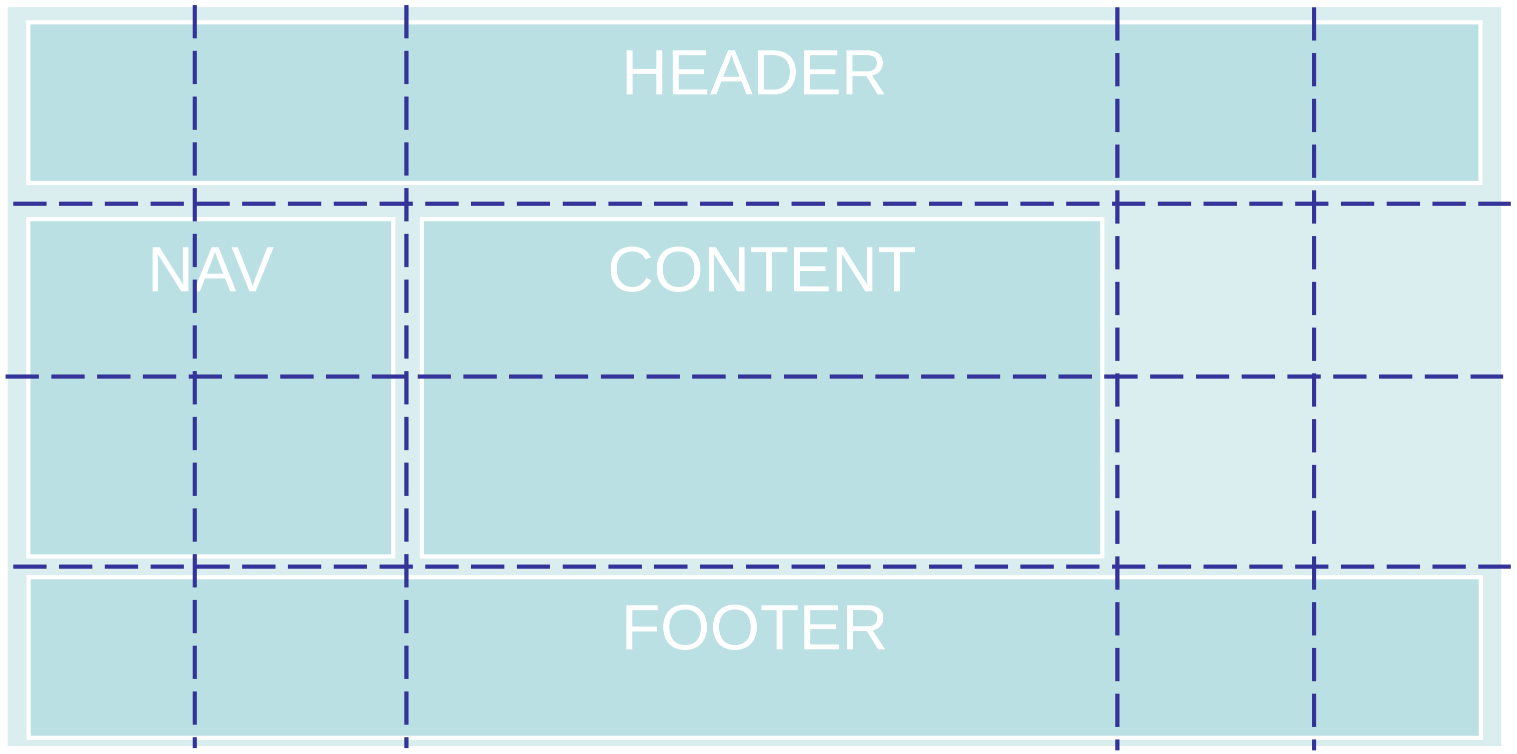
https://github.com/postcss/autoprefixer

Demo

Flexbox Alternative

CSS Grid Layout

Grid Layout



Grid Layout

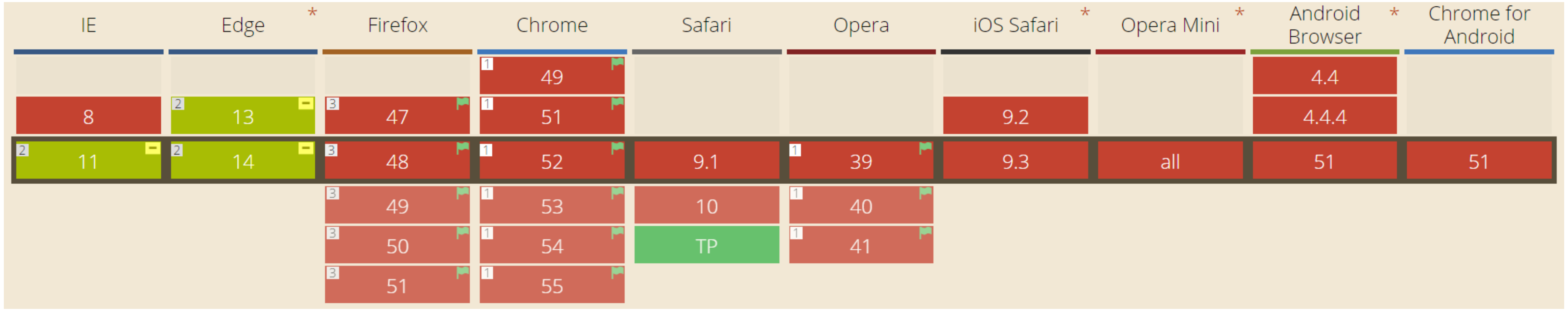
Imagine defining the layout of your entire page, and then completely rearranging it to accommodate a different screen width all with only a couple lines of CSS. Grid is one of the most powerful CSS modules ever introduced.

Grid Layout

An important thing to understand about Grid is that it's not ready to be used in production yet.
It's currently a W3C Working Draft and isn't supported correctly in any browsers yet by default. Internet Explorer 10 and 11 support it, but it's an old implementation with an outdated syntax.

Grid Layout

- Browser Support



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49[1]					4.4	
8	13[2]	47[3]	51[1]			9.2		4.4.4	
11[2]	14[2]	48[3]	52[1]	9.1	39[1]	9.3	all	51	51
		49[3]	53[1]	10	40[1]				
		50[3]	54[1]	TP	41[1]				
		51[3]	55[1]						

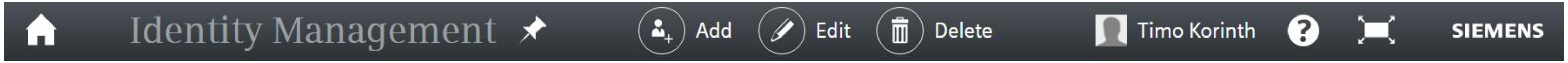
http://caniuse.com/#feat=css-grid

Best practices

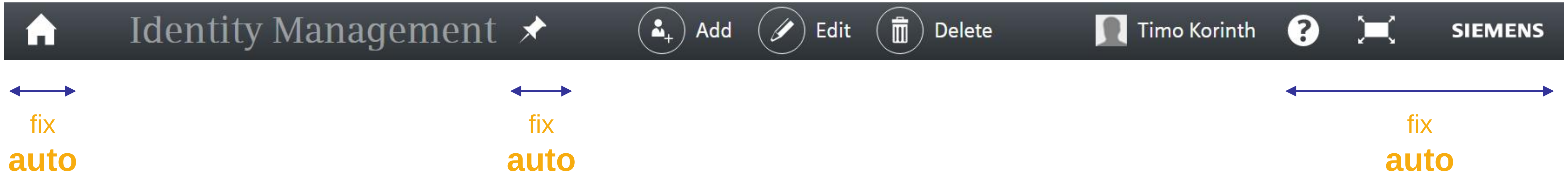
- Less | Sass
- Mixins für Browser Prefixe oder Autoprefixer
- Überall nutzen (auch tiefe Schachtelung möglich)
- Wichtigste Einstellung:

$$\left.\begin{array}{l}\texttt{flex: 0 0 auto;}\\ \texttt{flex: 1 1 100\%;}\end{array}\right\}\ 99\%$$

Advanced

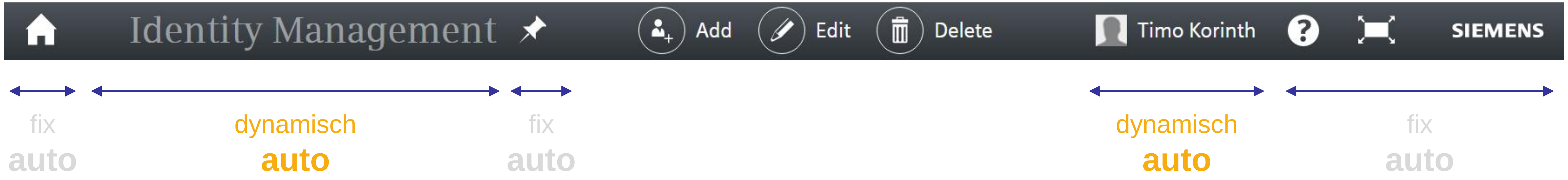


Advanced



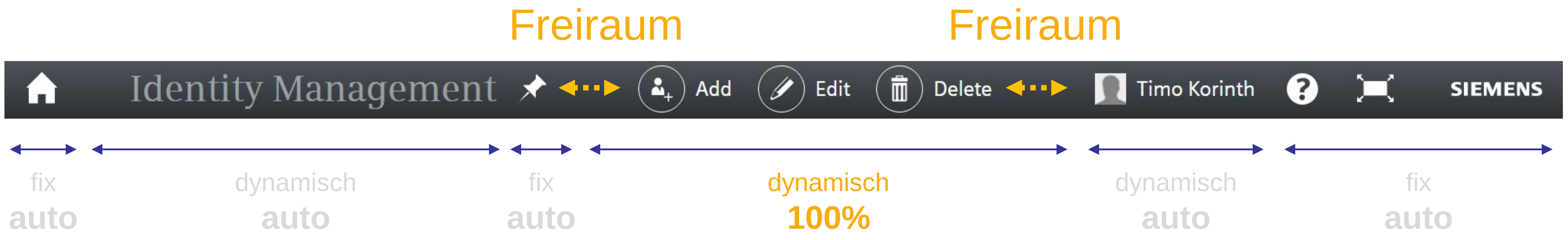
auto → flex: 0 0 auto;

Advanced



auto → flex: 0 0 auto;

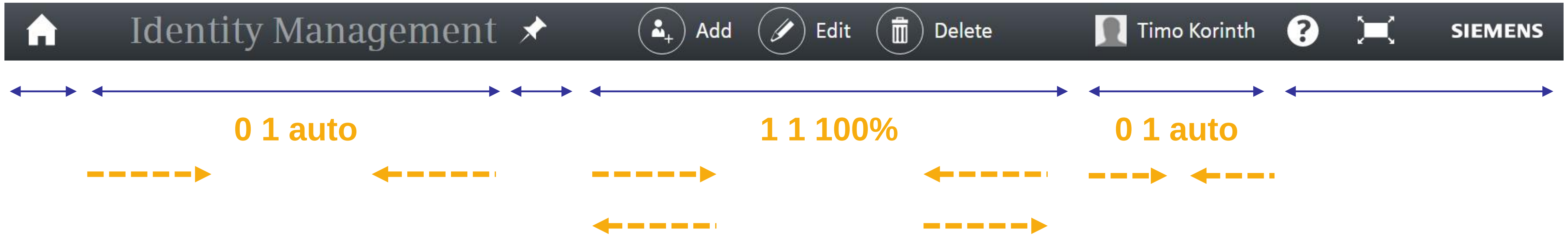
Advanced



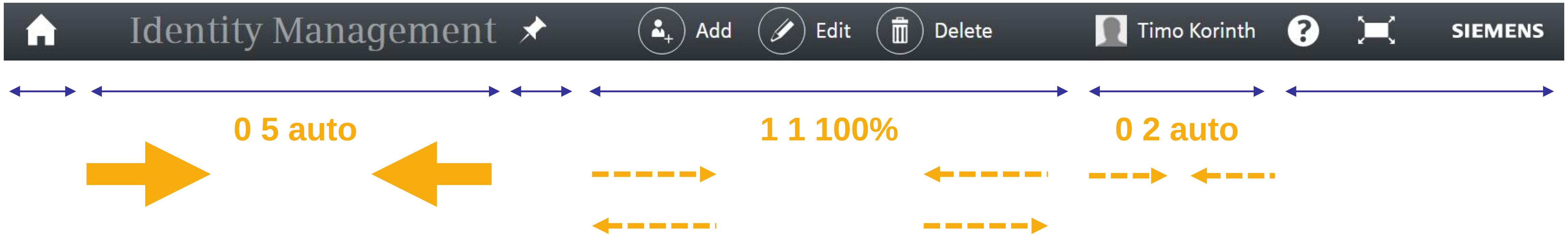
auto → flex: 0 0 auto;

100% → flex: 1 1 100%;

Responsive


Identity Management
Add
Edit
Delete
Timo Korinth
SIEMENS
0 1 auto
1 1 100%
0 1 auto

Responsive


Identity Management
Add
Edit
Delete
Timo Korinth
SIEMENS
0 5 auto
1 1 100%
0 2 auto

Performance

“Old flexbox (display: box) is 2.3x slower than new flexbox (display: flex).”

Google Dev, Flexbox layout isn't slow

“Flexbox is 4x faster than using floats.”

Google Dev, Avoid large, complex layouts and layout thrashing

Zukunft

Bulma

A modern CSS framework based on Flexbox

http://bulma.io/

Flexbox Grid

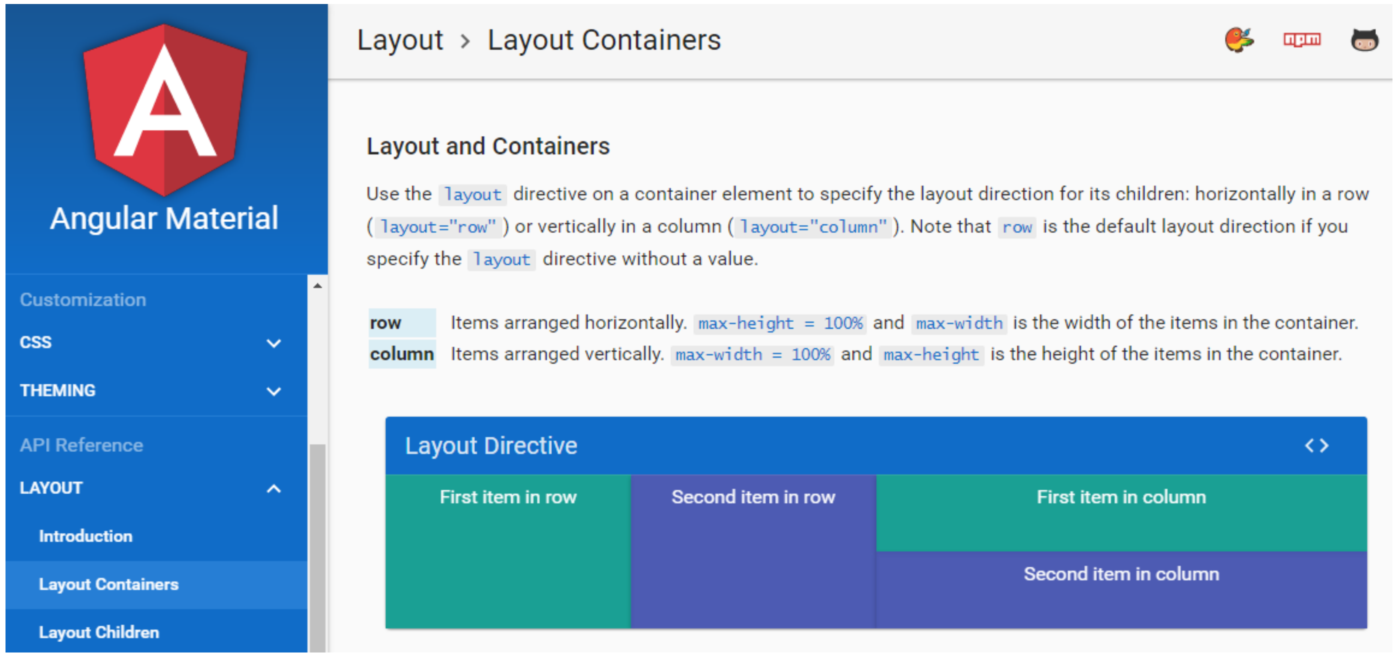
A grid system based on the `flex` display property.

http://flexboxgrid.com/

GRIDLEX

Just a Flexbox Grid System

http://gridlex.devlint.fr/



Layout and Containers

Use the `layout` directive on a container element to specify the layout direction for its children: horizontally in a row (`layout="row"`) or vertically in a column (`layout="column"`). Note that `row` is the default layout direction if you specify the `layout` directive without a value.

row	Items arranged horizontally. `max-height = 100%` and `max-width` is the width of the items in the container.
column	Items arranged vertically. `max-width = 100%` and `max-height` is the height of the items in the container.

https://material.angularjs.org/latest/

THX!

TIMO KORINTH ☺

tko@maximago.de
www.maximago.de