



MÜNCHEN .NET MEETUP, 12. FEBRUARY 2020

JOY HERON

Web Component Design

Maintaining and Reusing your Frontend

INNOQ



Joy Heron

Senior Consultant @ INNOQ

joy.heron@innoq.com

 @iamjoyheron

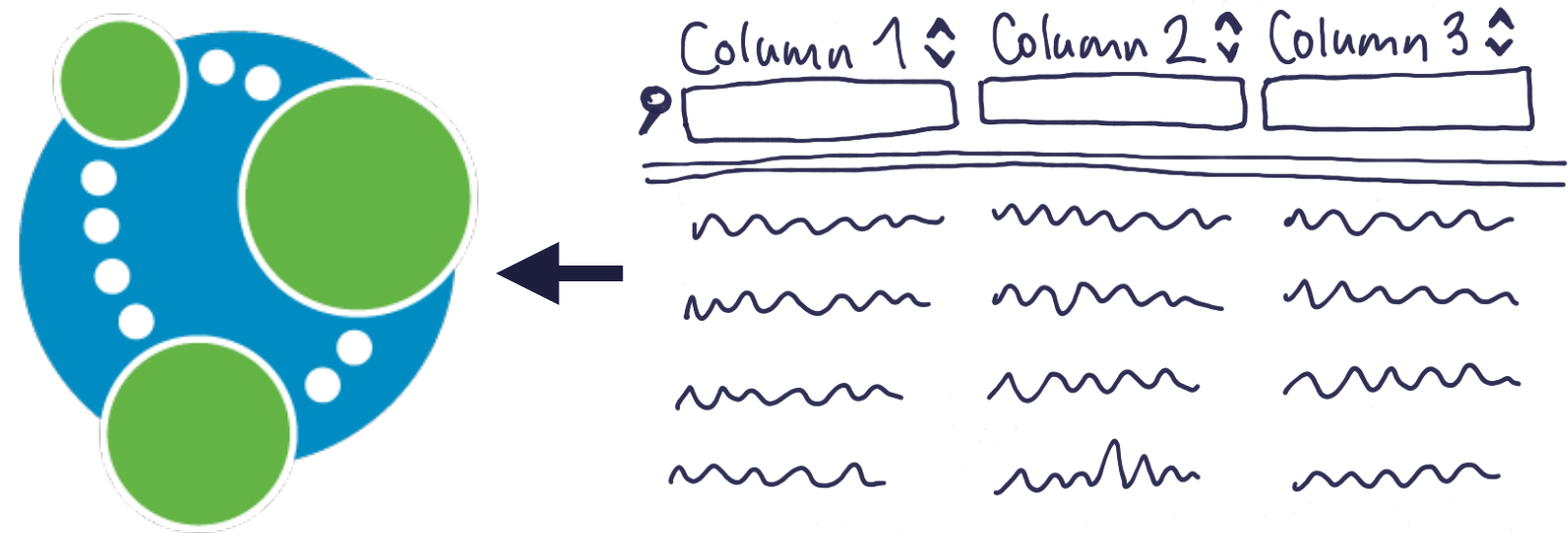
My Journey

Spring 2016

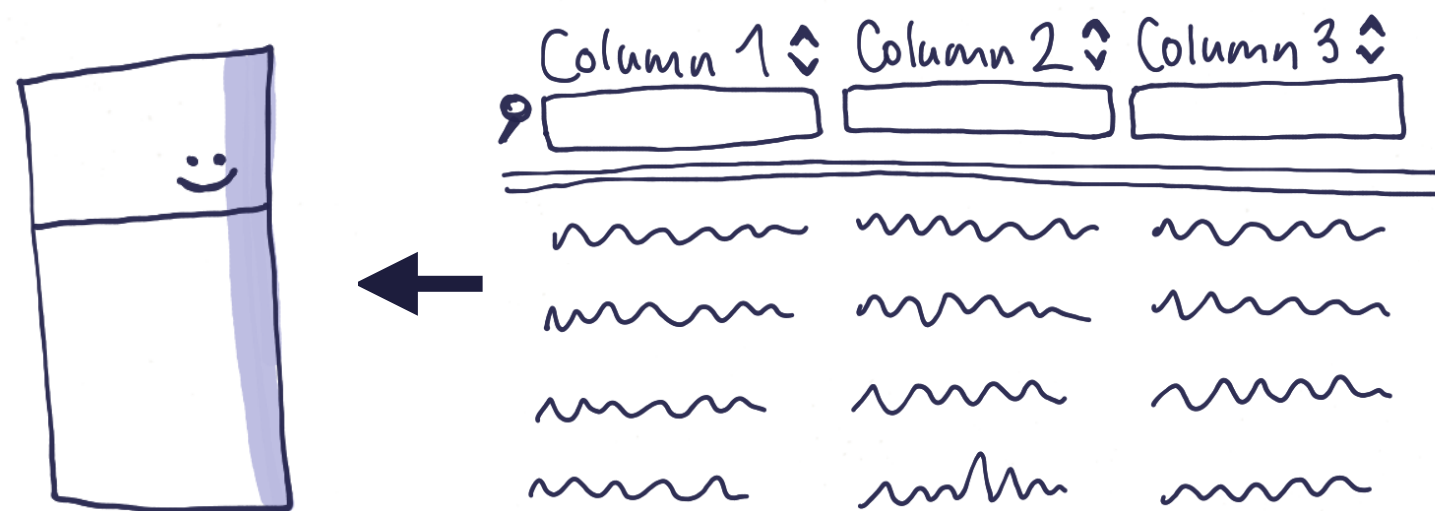


Column 1 ↕	Column 2 ↕	Column 3 ↕
🔑		
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~

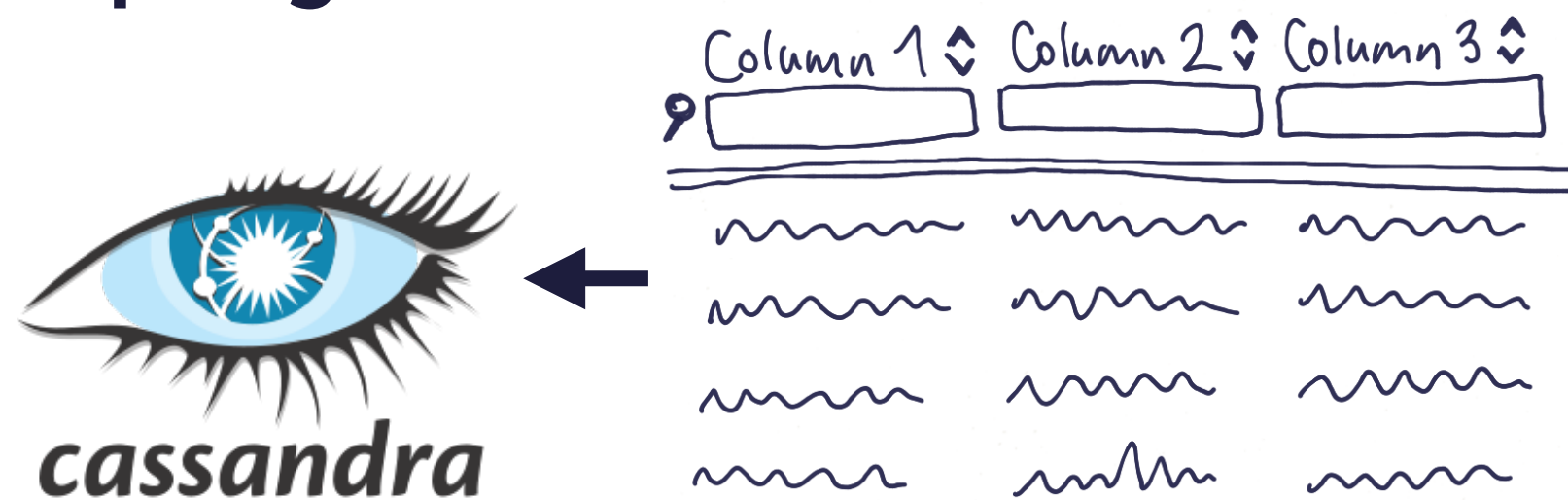
Spring 2016



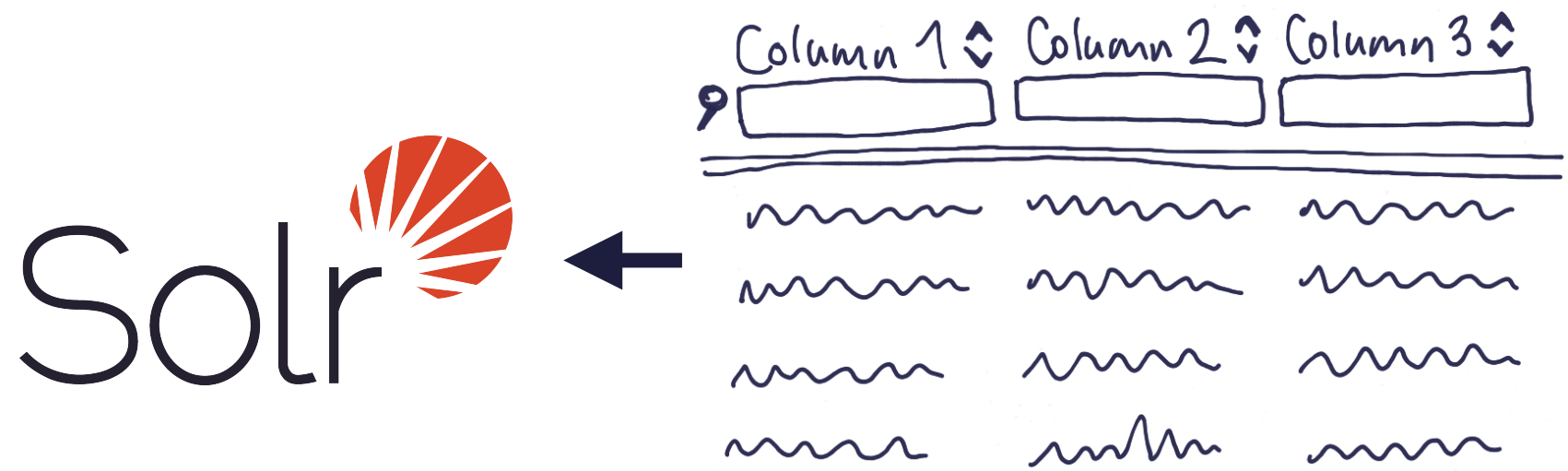
Summer 2016



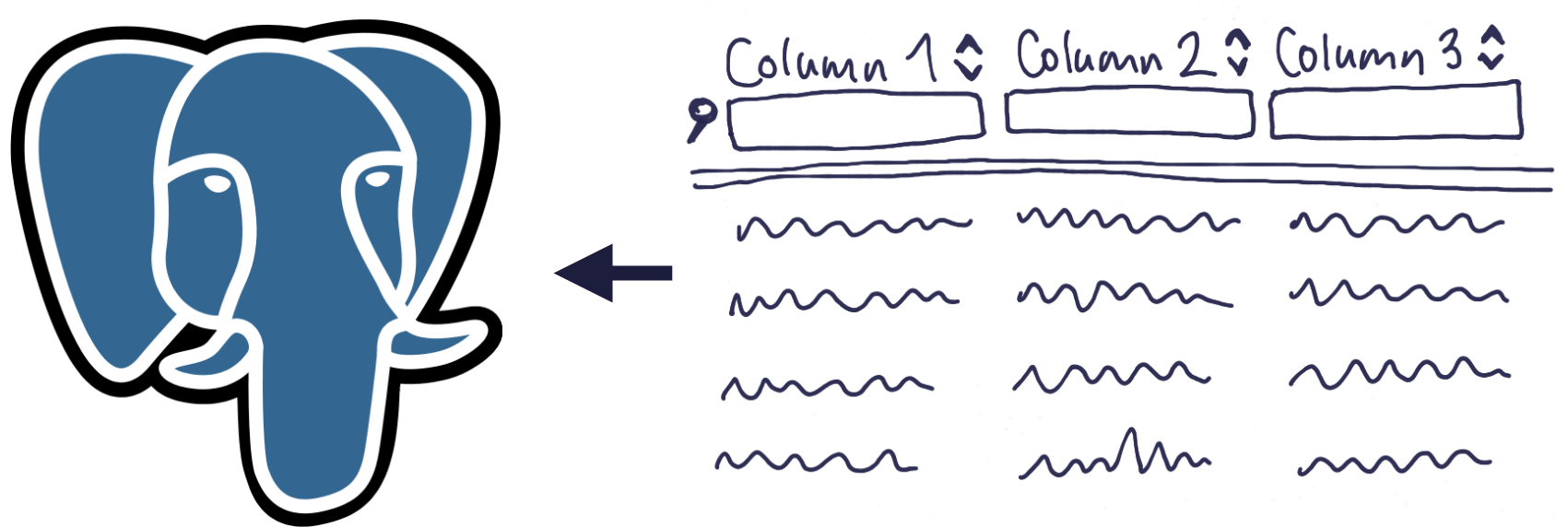
Spring 2017



Winter 2017



Fall 2018



... and in the future??

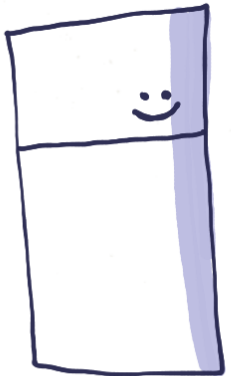


Column 1 ↕	Column 2 ↕	Column 3 ↕
9		
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~



Column 1 ↕	Column 2 ↕	Column 3 ↕
9		
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~

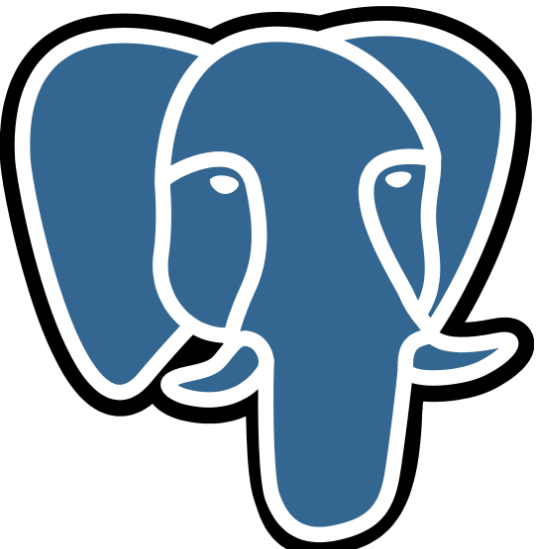
Why am I always
implementing the
same thing?



Column 1 ↕	Column 2 ↕	Column 3 ↕
9		
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~



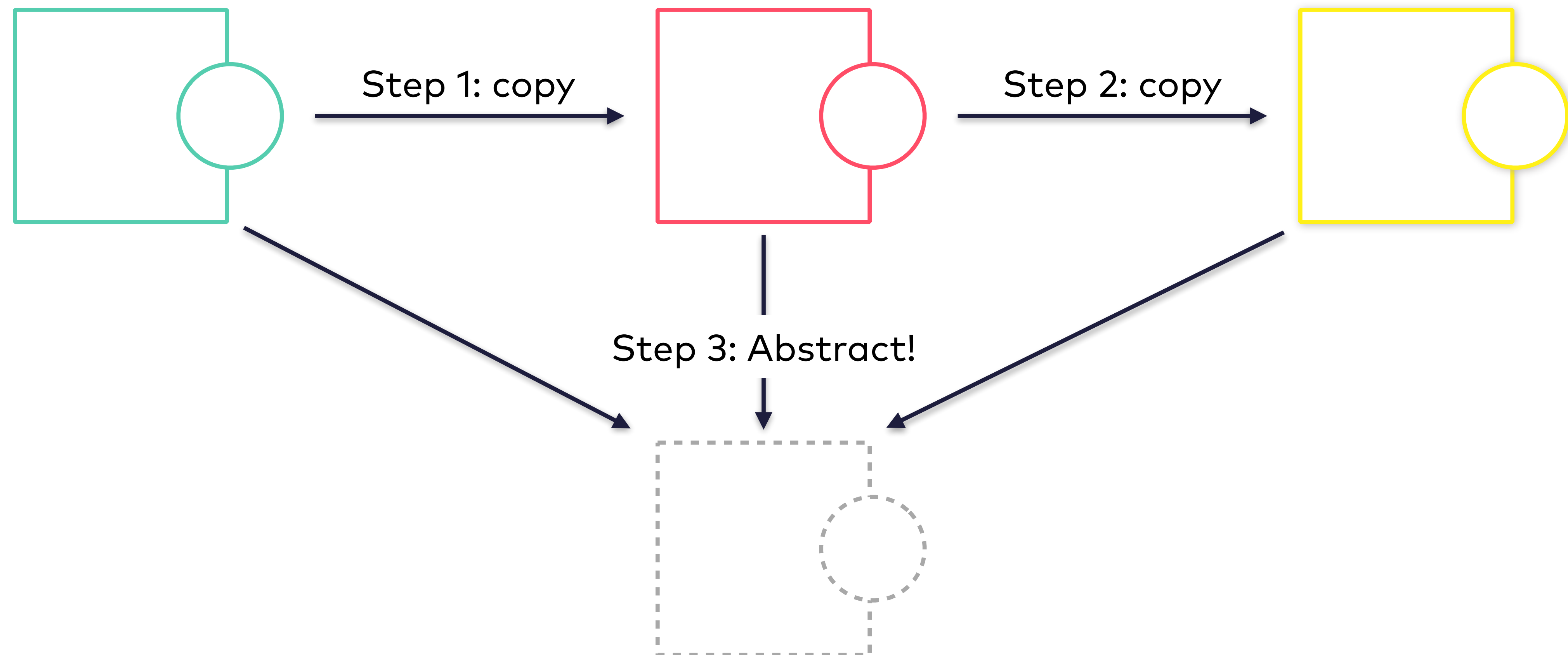
Column 1 ↕	Column 2 ↕	Column 3 ↕
9		
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~



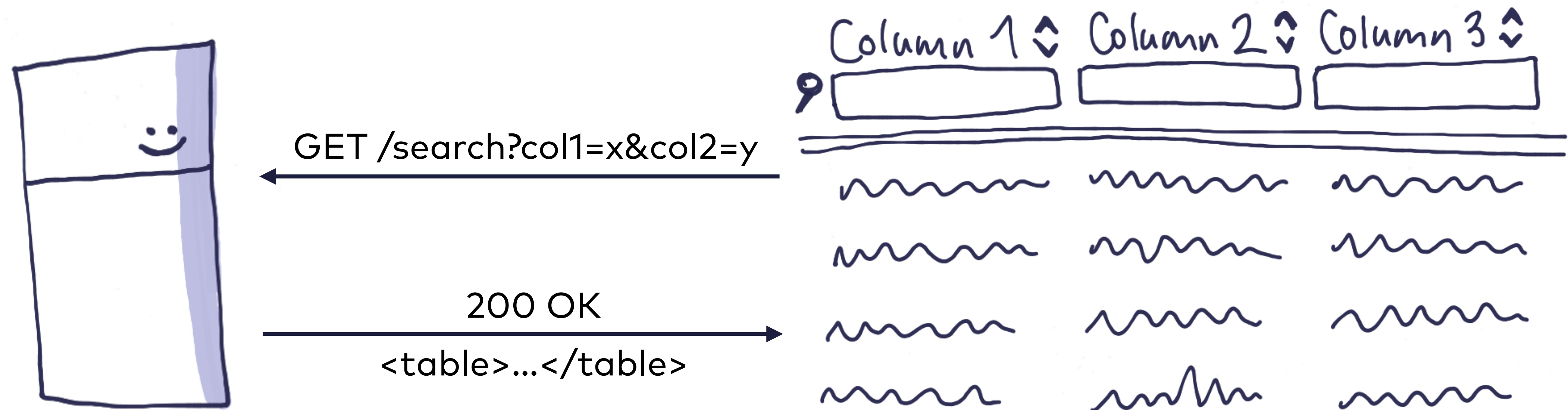
Column 1 ↕	Column 2 ↕	Column 3 ↕
9		
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~
~~~~~	~~~~~	~~~~~

**Would I do this in the
backend?**

Abstraction - Rule of Three



Time to find an abstraction!



**And now let's build
it with web
components!**

**That's what I did.
It's called: Tabelle**

<https://github.com/innoq/tabelle>

What is a web component?

possibly

JavaScript

make it better

probably

CSS

make it pretty

HTML

make it work
FOR ALL USERS!

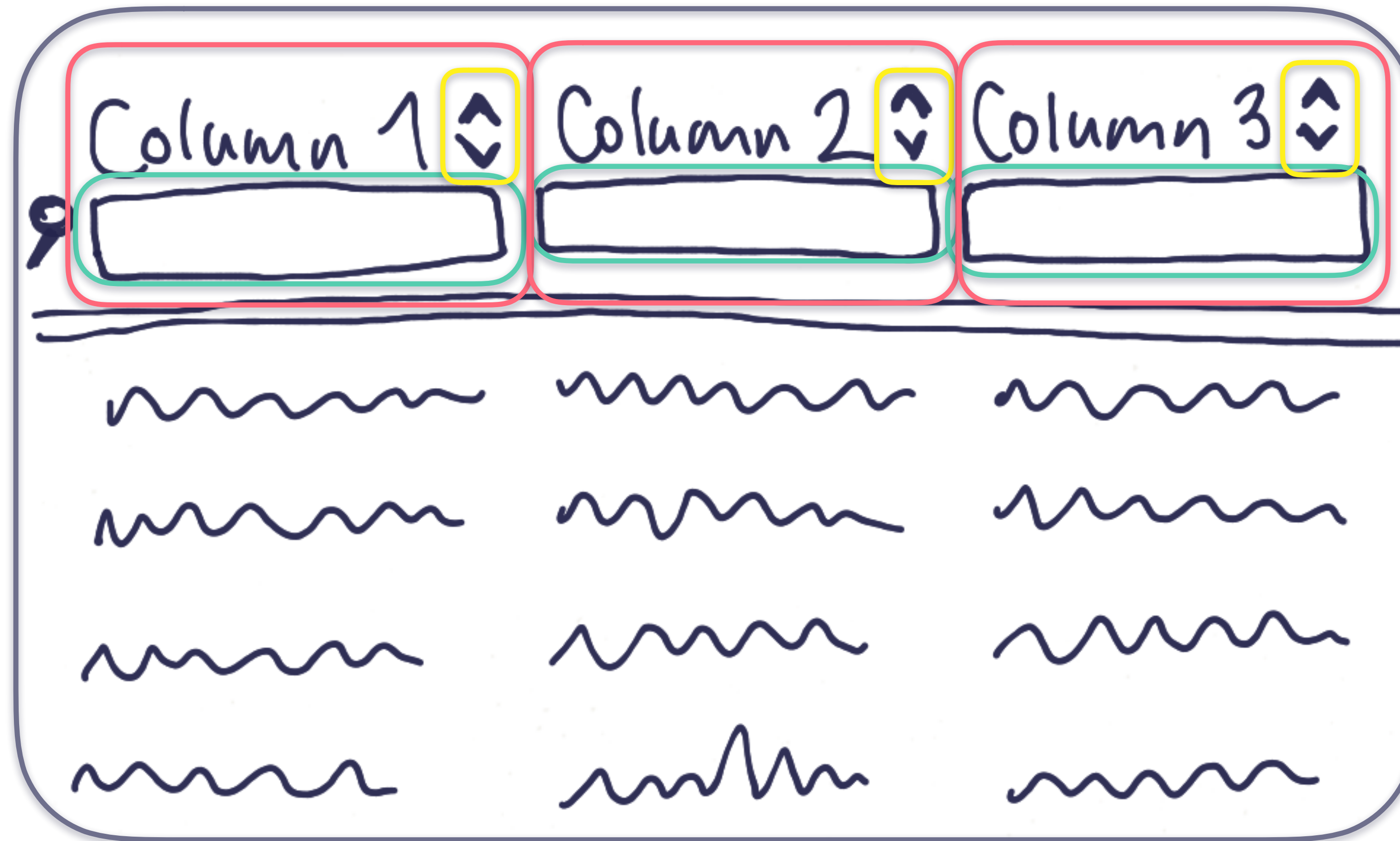
Single Responsibility Principle

Each Component should do
ONE thing **REALLY WELL**

If you need more functionality, write a new
component and compose them!

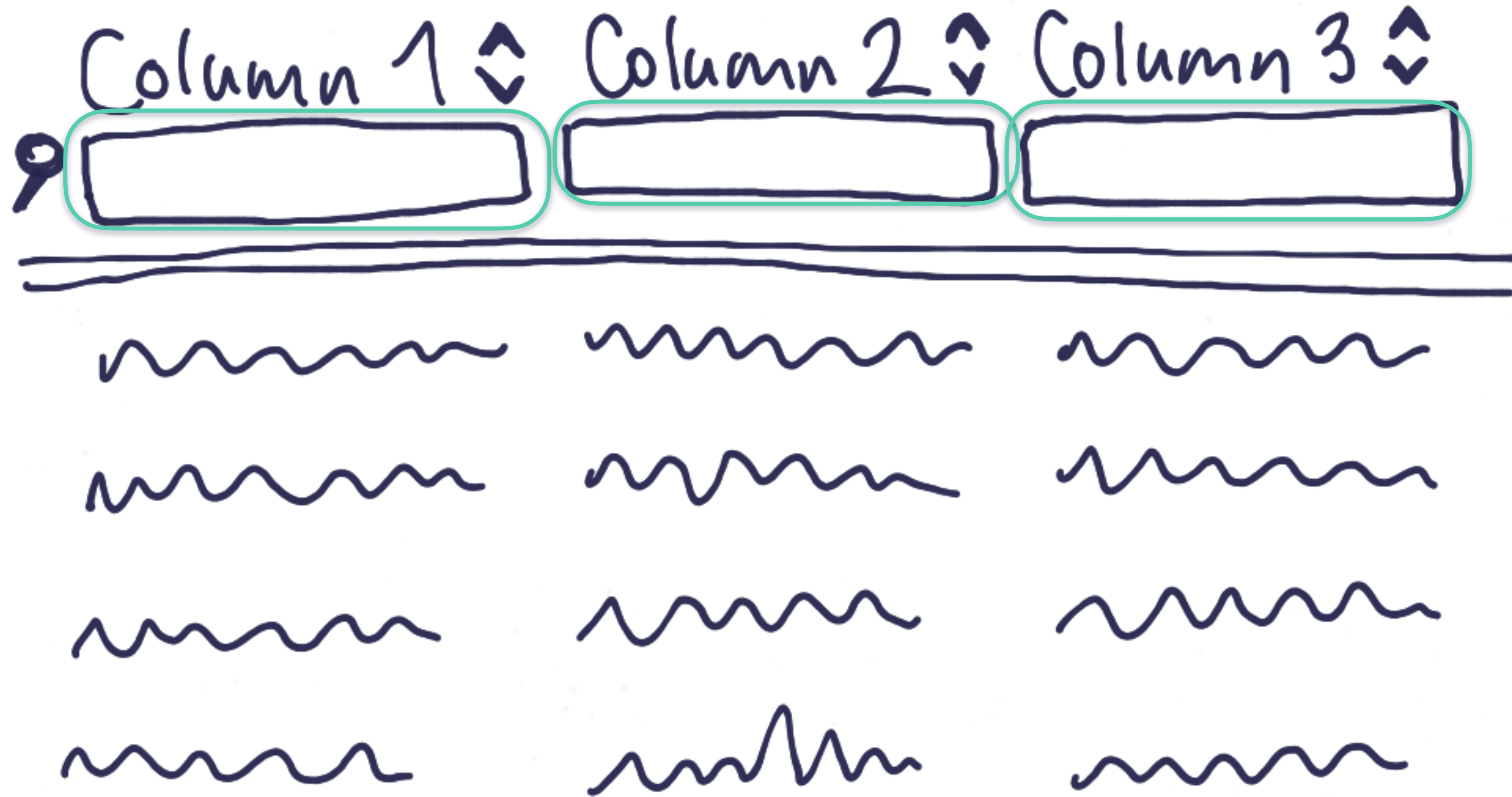
Benefit of HTML First:
HTML composes naturally!

How to find web components?



F
I
R
S
T

C
O
M
P
O
N
E
N
T



make it work

Filter Field Component

Step 1: make it work

What we need



GET /search?
col1=my+search+str&
col2=Option+1

How to get it

```
<form action="/search">
  <input type="text" name="col1" />
```

```
  <select name="col2">
    <option>Option 1</option>
    <option>Option 2</option>
    <option>Option 3</option>
  </select>
```

```
  <button type="submit">Search</button>
</form>
```



it works!

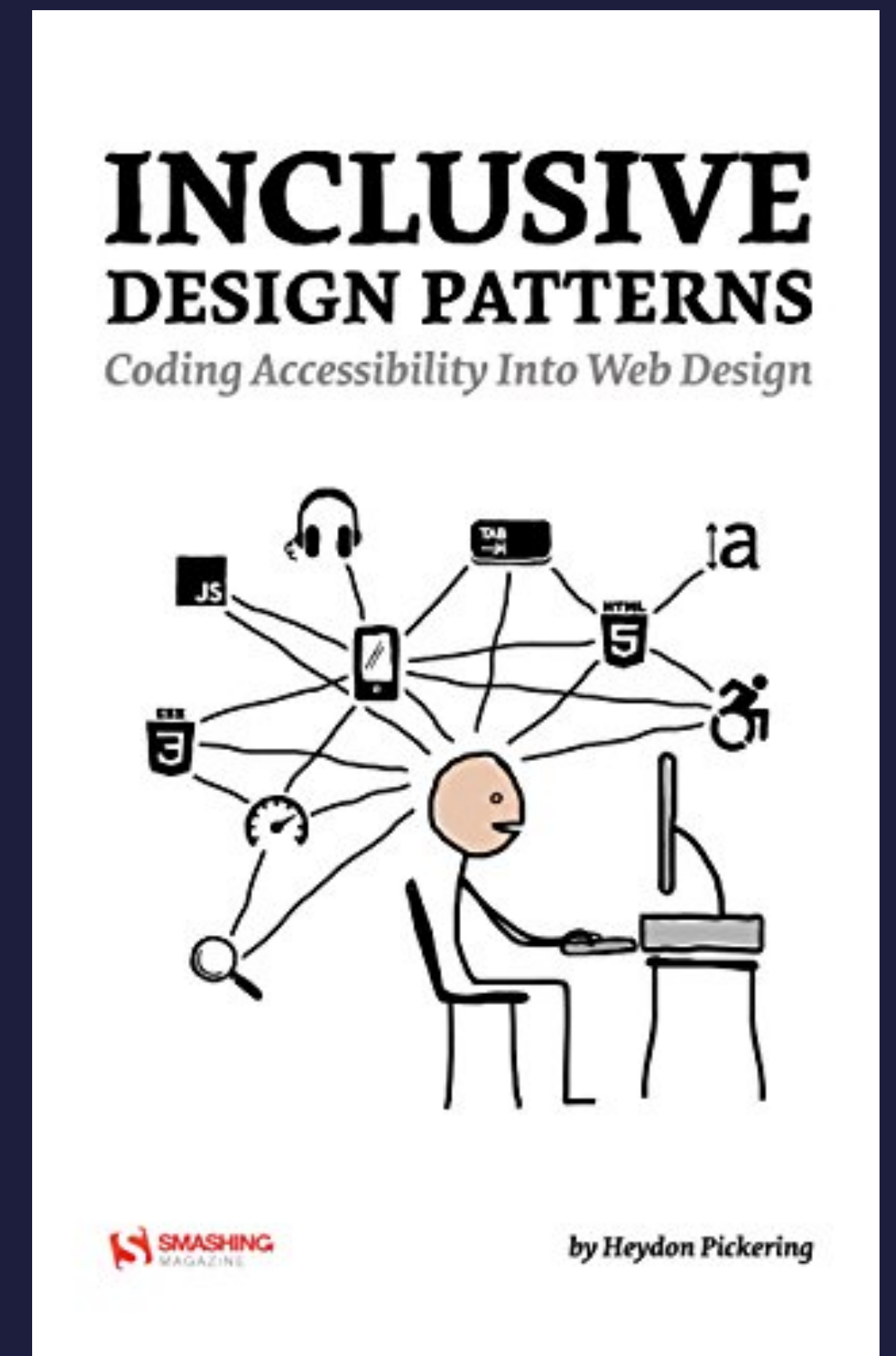
FOR ALL USERS??

Tip #1

If you want to make sure
you are accessible

**ACTUALLY USE A
SCREEN READER**

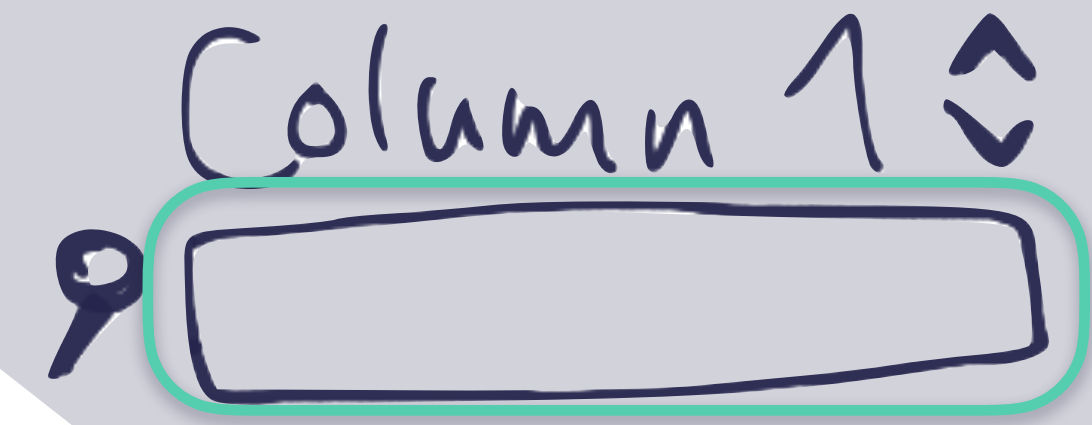
Read this book for
more tips!



Filter Field Component

Step 1.2: make it accessible

Let's test our component with a screen reader to see how it sounds!



Currently

```
<input type="text" name="col1" />
```



Edit Text.

*You are currently on a text field
inside of web content...*

Solution

```
<input type="text" name="col1"  
      aria-label="Filter Column 1" />
```



Filter Column 1, Edit Text.

*You are currently on a text field
inside of web content...*





Option 1

⌵

**HTML**

it works
FOR ALL USERS!

make it pretty??

**CSS is harder to
OVERRIDE than it is to
WRITE**

Tip #2

Use minimal CSS to make it
easy to override your styles
later!

Filter Field Component

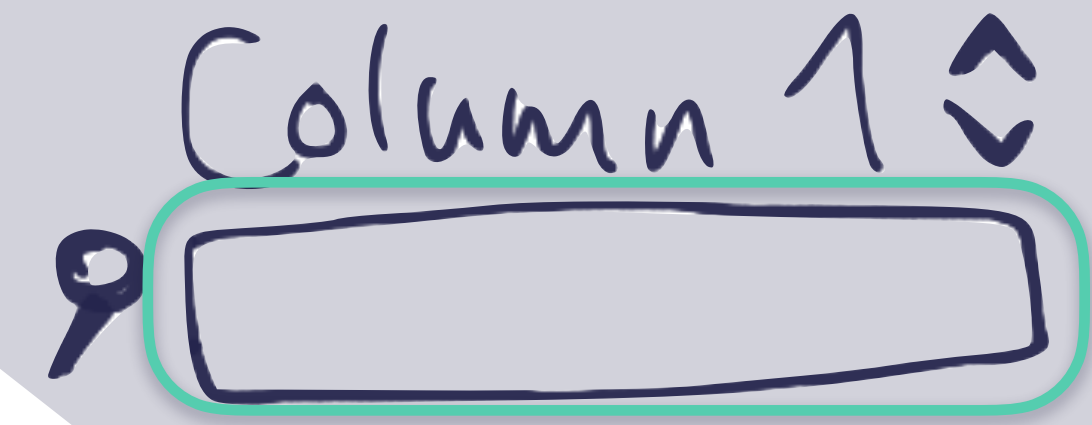
Step 2: make it pretty

Add a CSS class to our component to allow users to write their own styles later!

```
<input class="tabelle-input" type="text" name="col1" aria-label="Filter Column 1"/>
```

```
<select class="tabelle-input" name="col2" aria-label="Filter Column 1">
```

```
  ...  
</select>
```





? JS?

make it better?

✓ CSS

*someone else can
make it pretty!*

✓ HTML

*it works
FOR ALL USERS!*



 **JS**

no need!

 **CSS**

*someone else can
make it pretty!*

 **HTML**

*it works
FOR ALL USERS!*

S
E
C
O
N
D

C
O
M
P
O
N
E
N
T



make it work

Sort Button Component

Step 1: make it work

What we need



HTML Element?

☐ Sort Column 1 Ascending ☐ Sort Column 1 Descending



GET /search?sort=col1_asc

How to get it

```
<form action="/search">
```

```
<input id="col1_asc" type="radio"
      name="sort" value="col1_asc" />
```

```
<label for="col1_asc">Sort Column 1 Ascending</label>
```

```
<input id="col1_desc" type="radio"
      name="sort" value="col1_desc" />
```

```
<label for="col1_desc">Sort Column 2 Descending</label>
```

```
<button type="submit">Search</button>
</form>
```

Sort Button Component

Step 1.2: make it accessible

☐ Sort Column 1 Ascending ☐ Sort Column 1 Descending



Sort Column 1 Ascending, radio button, 1 of 2
You are currently on a radio button, 1 of 2, ...



Column 1 

☐ Sort Column 1 Ascending ☐ Sort Column 1 Descending



HTML

it works
FOR ALL USERS!

make it pretty

Tip #3

You can select an input element by clicking on it's HTML label

Sort Button Component

Step 2: make it pretty

Visually hide radio buttons and style their labels.

Move content offscreen so it is not visible but can still be read by a screen reader.

SCSS Placeholder Selector & Util CSS Class

```
%visually-hidden {  
  position: absolute !important;  
  height: 1px; width: 1px;  
  overflow: hidden;  
  clip: rect(1px 1px 1px 1px); /* IE6, IE7 */  
  clip: rect(1px, 1px, 1px, 1px);  
}  
  
.visually-hidden {  
  @extend %visually-hidden;  
}
```

Source: https://snook.ca/archives/html_and_css/hiding-content-for-accessibility

Sort Button Component

Step 2: make it pretty

Visually hide radio buttons and style their labels.

```
.arrow {  
  @extend %visually-hidden;  
}  
  
.arrow-asc {  
  @include icon-before('up.svg', #acacac);  
  cursor: pointer;  
}  
  
.arrow-desc {  
  @include icon-before('down.svg', #acacac);  
  cursor: pointer;  
}
```

```
<input class="arrow" id="col1_asc" ... />  
<label class="arrow-asc" for="col1_asc">  
  Sort Column 1 Ascending  
</label>  
  
<input class="arrow" id="col1_desc" ... />  
<label class="arrow-desc" for="col1_desc">  
  Sort Column 2 Descending  
</label>
```

So far, so good:

^ Sort Column 1 Ascending
v Sort Column 1 Descending

Sort Button Component

Step 2: make it pretty

- **Style checked, focused, and hover input states!**
- **Because our label follows our input, we can style this using pure CSS with a '+'!**

```
.arrow:checked + .arrow—asc::before,  
.arrow:checked + .arrow—desc::before {  
  background-color: #535353;  
}
```

```
.arrow:focus + .arrow—asc::before,  
.arrow:focus + .arrow—desc::before,  
.arrow:hover + .arrow—asc::before,  
.arrow:hover + .arrow—desc::before {  
  background-color: #6882cb;  
}
```

Result:

focused:

^ Sort Column 1 Ascending
v Sort Column 1 Descending

checked:

^ Sort Column 1 Ascending
v Sort Column 1 Descending

Sort Button Component

Step 2: make it pretty

Hide our labels visually

```
<input class="arrow"
      id="col1_asc" type="radio"
      name="sort" value="col1_asc" />
<label class="arrow—asc" for="col1_asc">
  <span class="visually-hidden">Sort Column 1 Ascending</span>
</label>

<input class="arrow"
      id="col1_desc" type="radio"
      name="sort" value="col1_desc" />
<label class="arrow—desc" for="col1_desc">
  <span class="visually-hidden">Sort Column 1 Descending</span>
</label>
```

base:



checked:



focused:



Sort Button Component

Step 2.2: Double-check accessibility



Sort Column 1 Ascending, radio button, 1 of 2

You are currently on a radio button, 1 of 2, ...



Column 1



? JS?

make it better?

✓ CSS

it's pretty!

✓ HTML

it works
FOR ALL USERS!

Column 1



JS

we're ok for now!



CSS

it's pretty!



HTML

it works
FOR ALL USERS!

We've written some
components...

But how can we **MAINTAIN**
them over time?

Tip #4

Use a templating engine as
an abstraction for your
component

Possible templating engines: JSX / complate, Handlebars

Benefits of HTML Templates for web components

Change ONCE, modify all instances

Easier to MAINTAIN components over time

PUBLISH templates as an npm library for use in other projects

We need a big mental shift:
From "Let's Build a Website" to
"Let's maintain a product which
other products use as a dependency"

- Brad Frost

input and select components

```
/* input.jsx */  
function Input ({ name, label }, ...kids) {  
  return <input class="tabelle-input"  
    type="text" name={name}  
    aria-label={label} />  
}
```



```
<Input name="col1"  
  label="Column 1" />
```

```
/* select.jsx */  
function Select ({ name, label }, ...kids) {  
  return <select class="tabelle-input"  
    name={name}  
    aria-label="Filter {label}">  
  
    {kids}  
  </select>
```



```
<Select name="col2"  
  label="Column 2">  
  <option>A</option>  
  <option>B</option>  
  ...  
  <option>Z</option>  
</Select>
```

arrows component

```
/* arrows.jsx */
function Input ({ name, label }, ...kids) {
  return <>
    <input class="arrow" id="{name}_asc" type="radio"
      name="sort" value="{name}_asc" />
    <label class="arrow-asc" for="{name}_asc">
      <span class="visually-hidden">
        Sort {label} Ascending
      </span>
    </label>

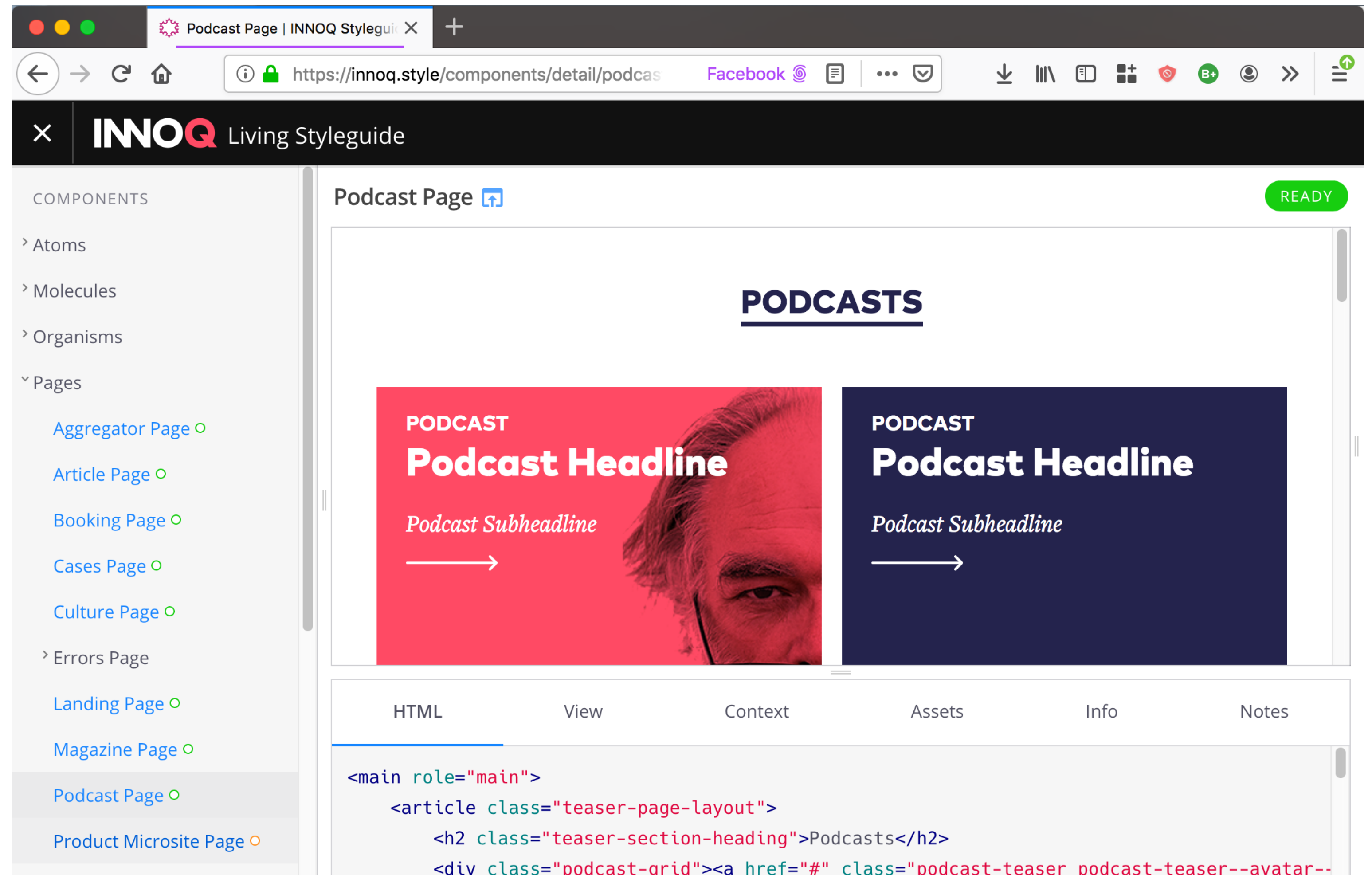
    <input class="arrow" id="{name}_desc" type="radio"
      name="sort" value="{name}_desc" />
    <label class="arrow-desc" for="{name}_desc">
      <span class="visually-hidden">
        Sort {label} Descending
      </span>
    </label>
  </>
}
```

→ **<Arrows name="col3"
label="Column 3" />**

Tip #5

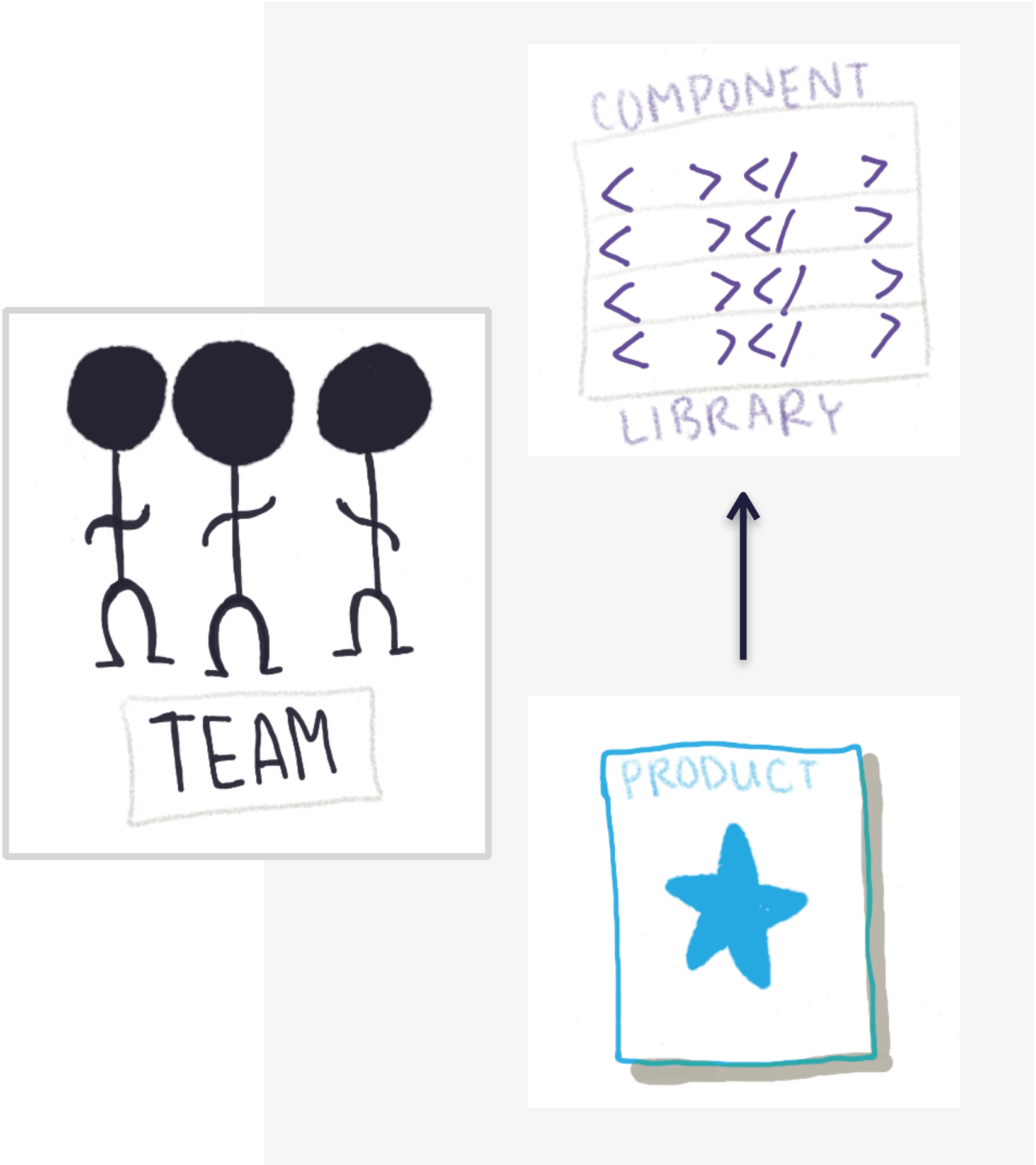
Develop your components using a pattern library or living styleguide
(patternlab.io, [fractal](https://fractal.design/))

BONUS POINTS:
publish your pattern library as reusable templates



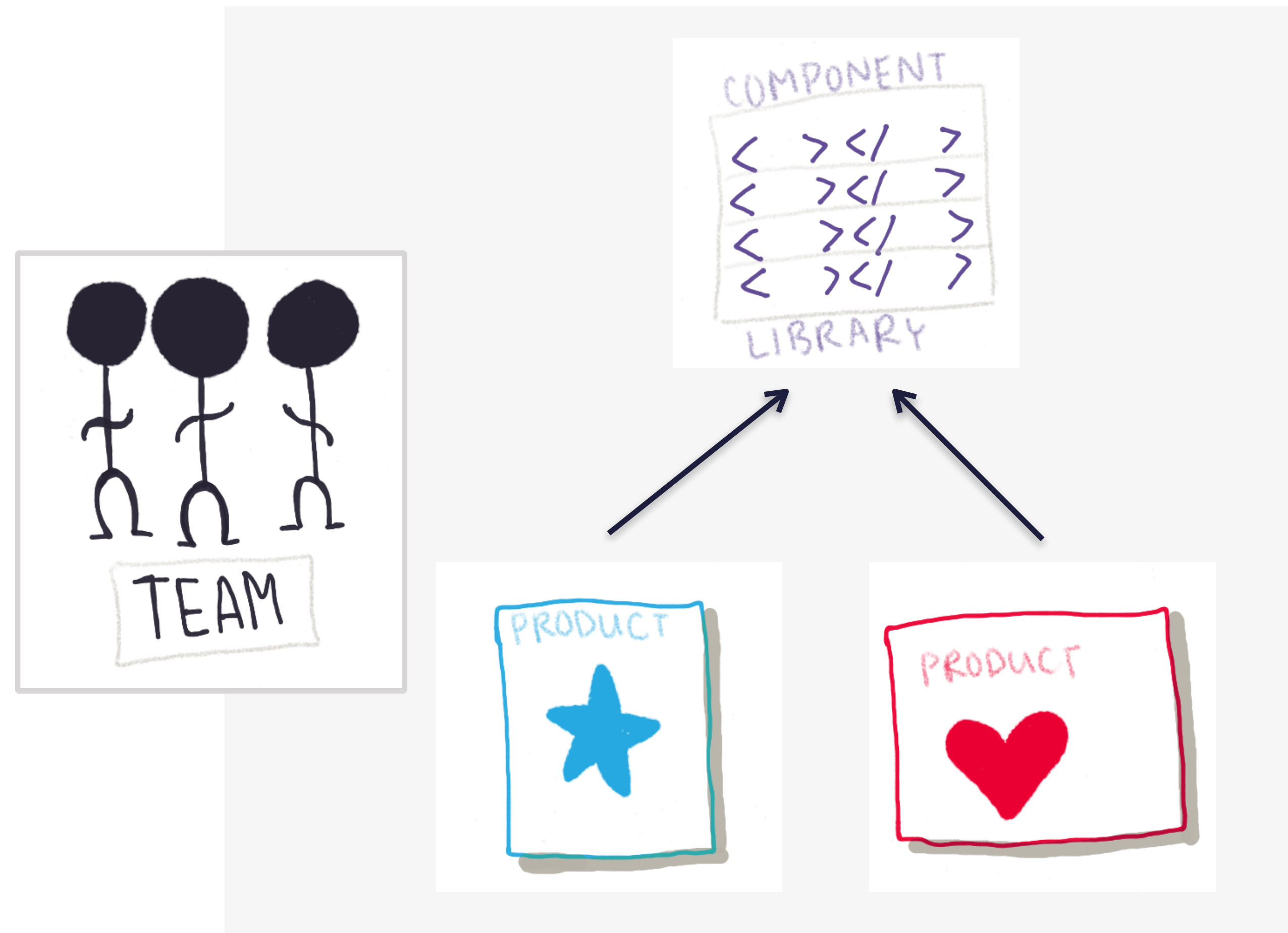
How does this work in a team?

A simple team



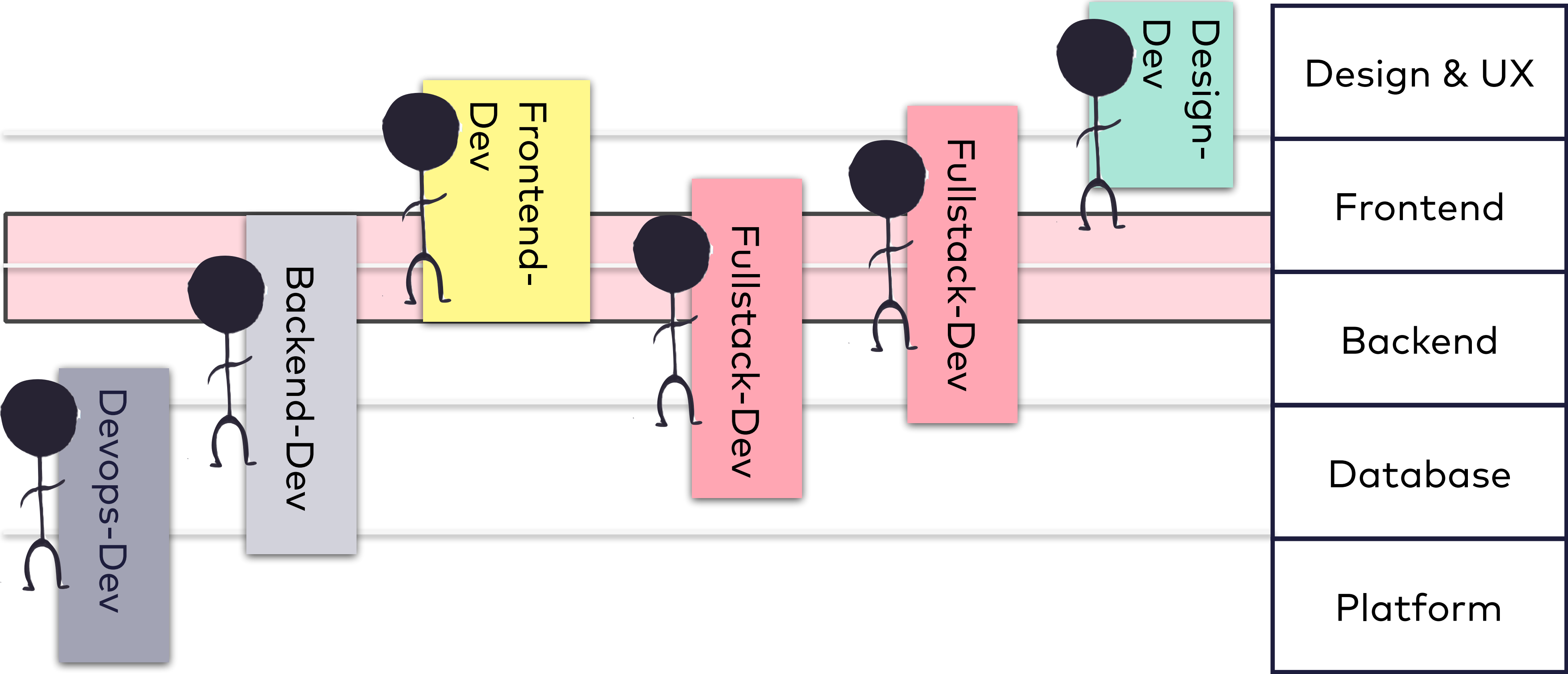
A new product
with the same
design system?

But what
happens when
we have to split
our teams?



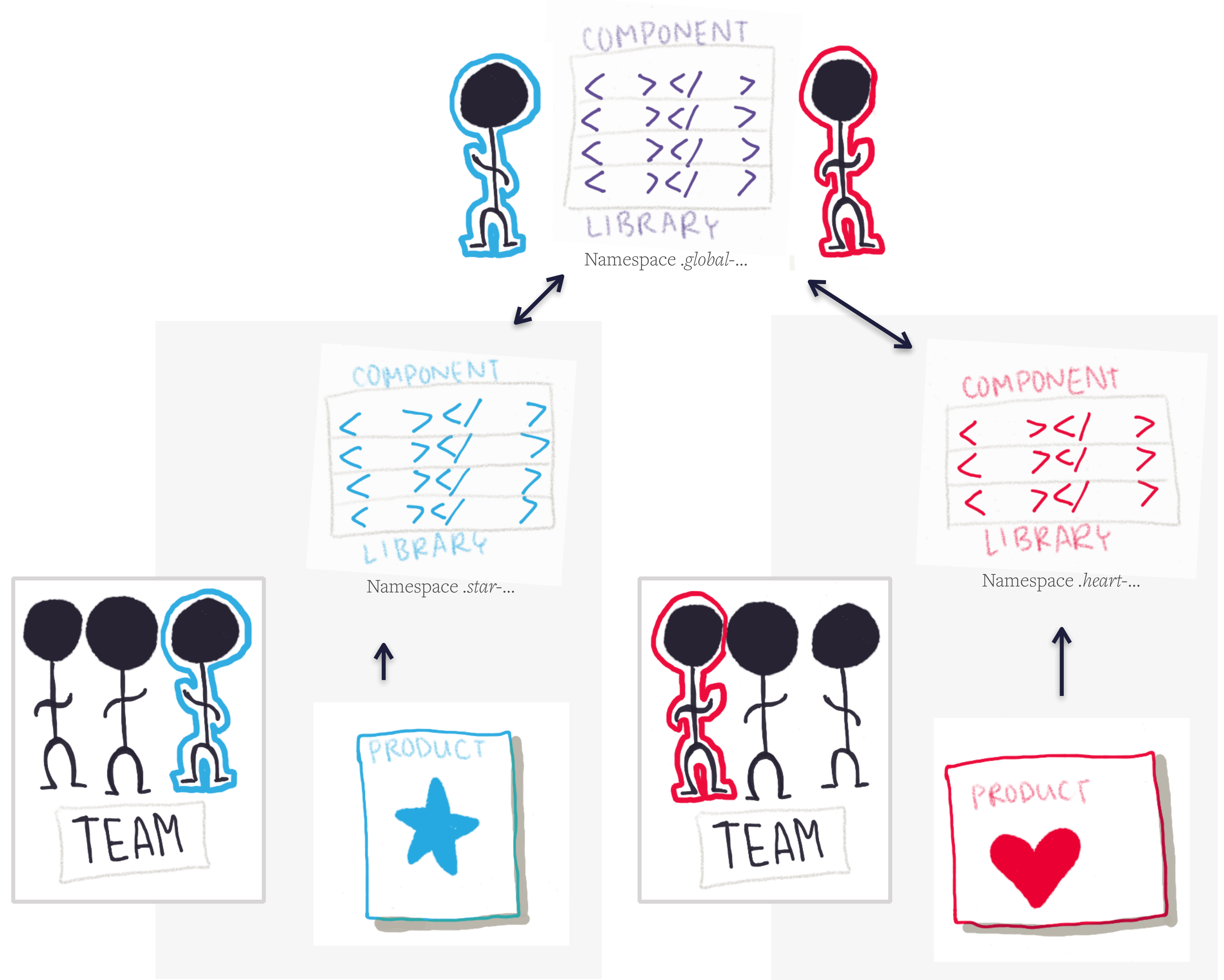
Cross-functional Team

Web components provide a *language* between backend and frontend



Collaboration with split teams

AVOID throwing
components over
the wall!



**There are loads of
different team
constellations
but in all of them
we want...**

**to RELEASE new versions of our component
library quickly as we develop new ideas**

**to UPGRADE to new versions of our
component library quickly WITHOUT worrying
about breaking changes**

Tip #6

When publishing a library of templates, keep the components **backward compatible** to make it easy to upgrade!

How to make a component backward compatible

**When adding new parameters, make them
OPTIONAL or add a default value**

**Instead of breaking a component, create a
COPY of the existing implementation and give
it a NEW NAME**

**Avoid DEPENDING too strongly on the
structure of the component when using it in
other components or in UI Tests**

...now to our next
component...

T
H
I
R
D

C
O
M
P
O
N
E
N
T



make it work

Table Header Component

Step 1: make it work

```
<div>  
  Column 1  
  <Arrows name="col1" label="Column 1" />  
  <Input name="col1" label="Column 1" />  
</div>
```

Column 1





Table Header Component

Step 1.2: make it accessible

Column 1
^
v

Column 2
^
v

Column 3
^
v



Column 1

- *Sort Column 1 Ascending, radio button 1 of 6*
- *Sort Column 1 Ascending*
- *Sort Column 1 Descending, radio button 2 of 6*
- *Sort Column 1 Descending*
- *Filter Column 1, edit text*
- **Column 2**
- *Sort Column 2 Ascending, radio button 3 of 6*
- *Sort Column 2 Ascending*
- *Sort Column 2 Descending, radio button 4 of 6*
- ...

Tip #7

Wrap input fields in a
role=group to add a
navigation level for a screen
reader

Note: The built-in HTML Element *fieldset* provides similar behavior but is extremely difficult to style

Table Header Component

Step 1.2: make it accessible



```
<div>  
  Column 1  
  <Arrows name="col1" label="Column 1" />  
  <Input name="col1" label="Column 1" />  
</div>
```



```
<div role="group" aria-labelledby="col1_group">  
  <span id="col1_group">Column 1</span>  
  <Arrows name="col1" label="Column 1" />  
  <Input name="col1" label="Column 1" />  
</div>
```



Table Header Component

Step 1.2: make it accessible

Column 1
^
v

Column 2
^
v

Column 3
^
v



Column 1, group

VO+→

Column 2, group

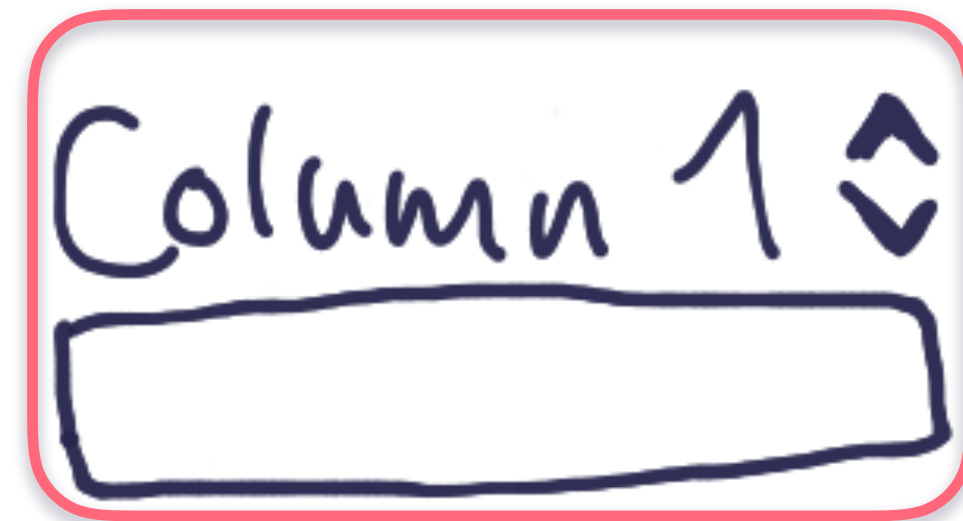
VO
+
↓



Column 2

- Sort Column 2 Ascending, radio button 3 of 6
- Sort Column 2 Ascending
- Sort Column 2 Descending, radio button 4 of 6
- Sort Column 2 Descending
- Filter Column 2, edit text

Navigation via groups with a screen reader!



Column 1

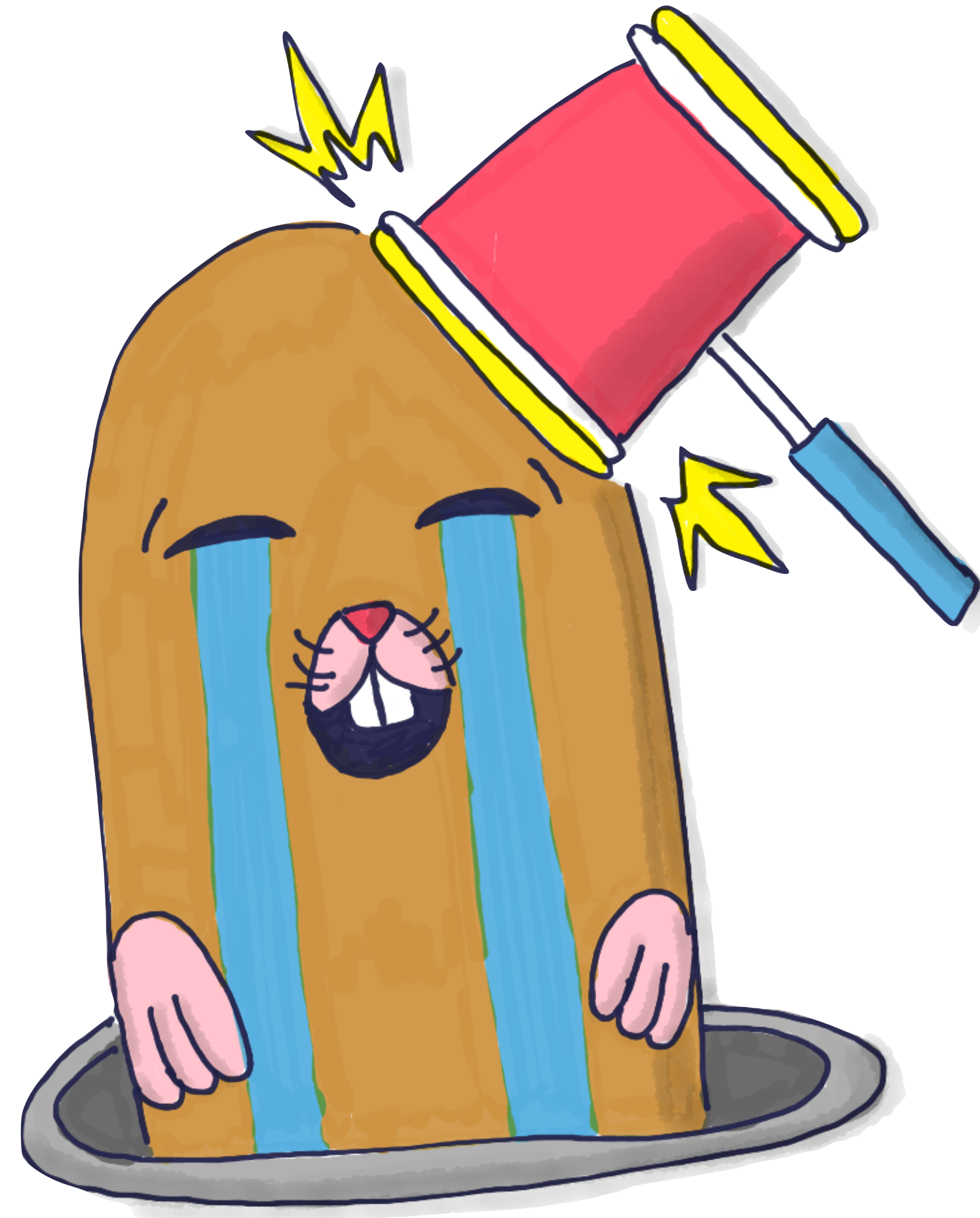


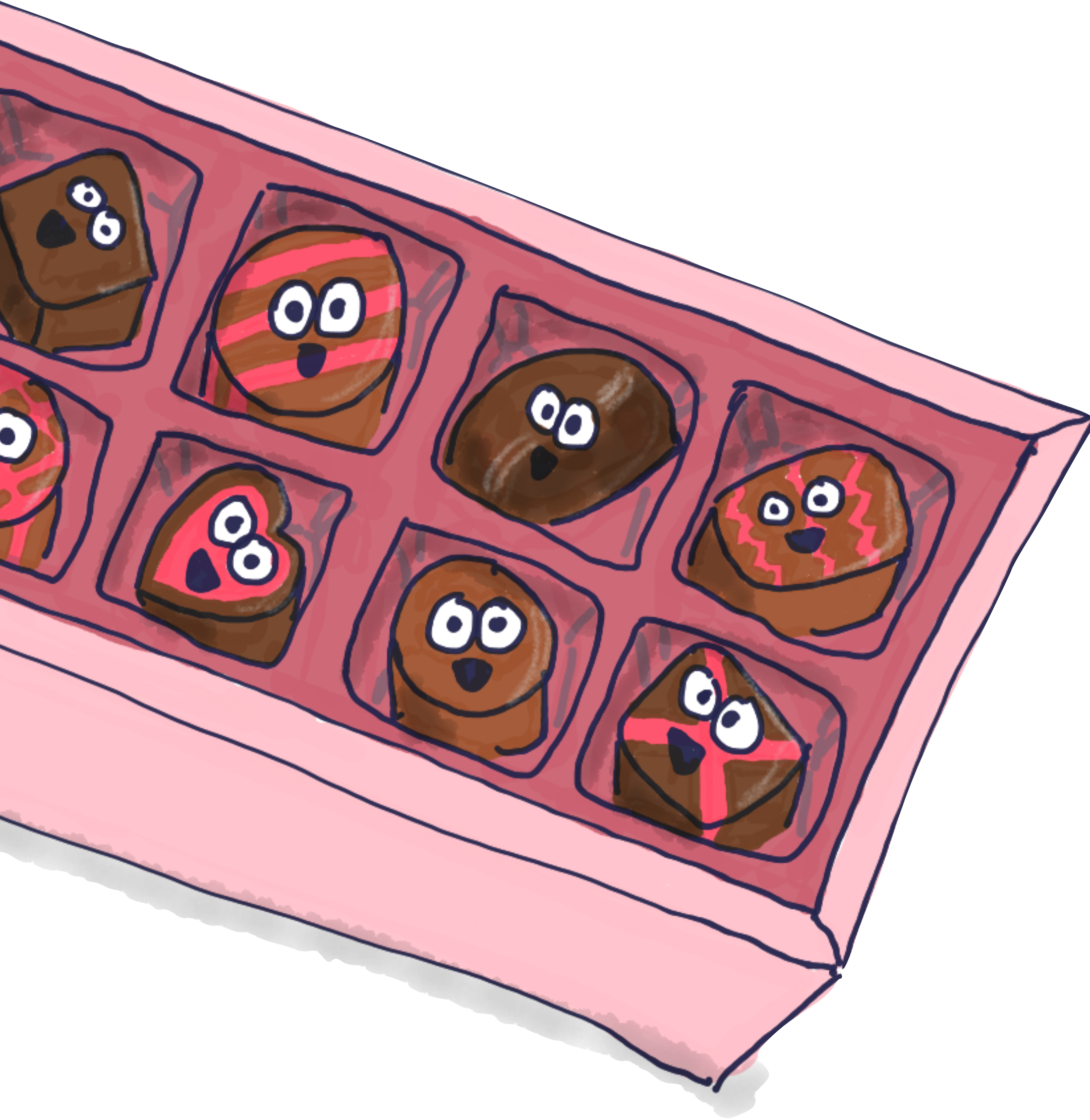
HTML

it works
FOR ALL USERS!

make it pretty

R.I.P
Whack-A-Mole
CSS





Tip #8

Design CSS

like a box of chocolates

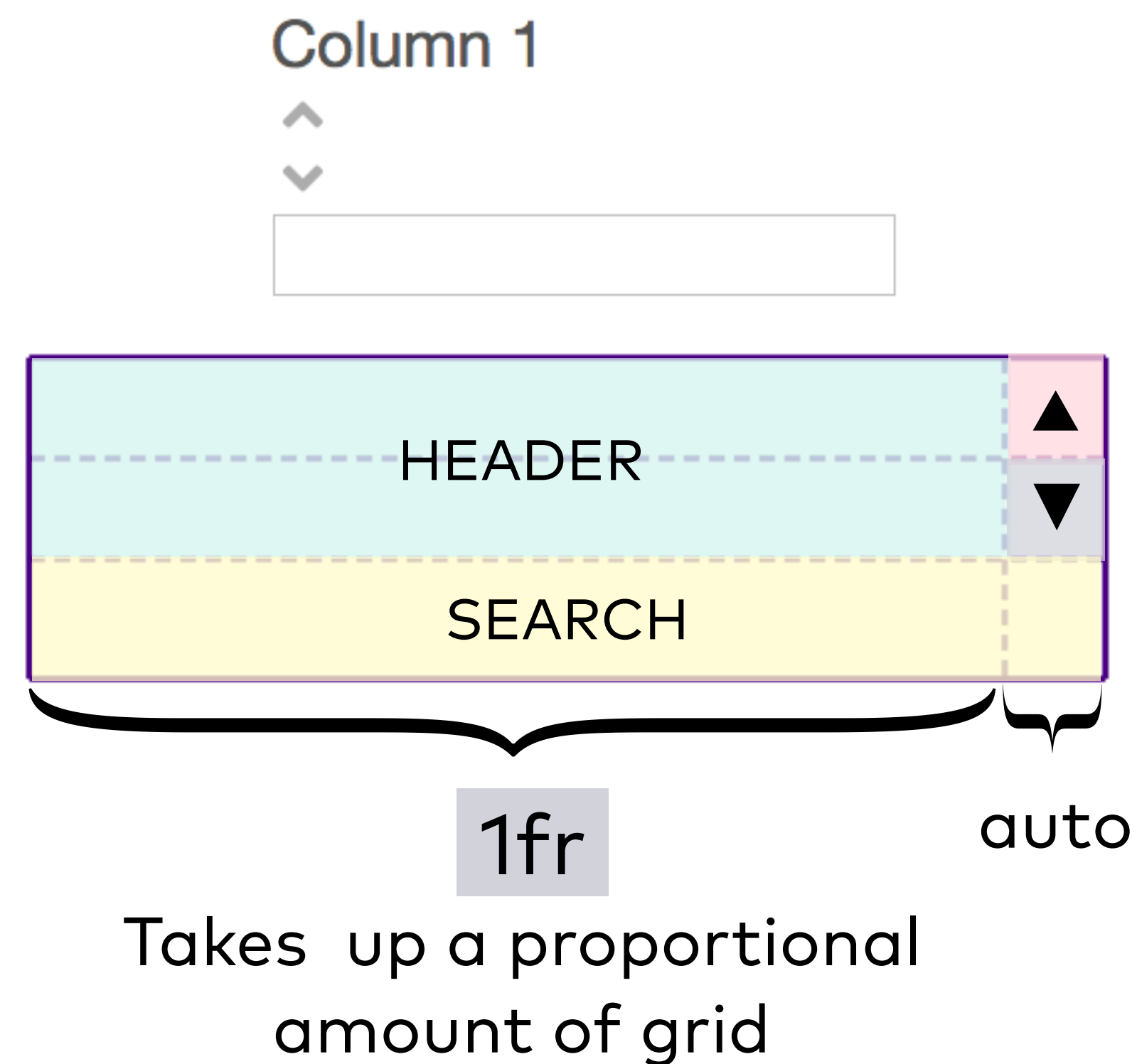
Design your container with
love and *place* your
components inside it.

(use Flexbox or CSS Grid)

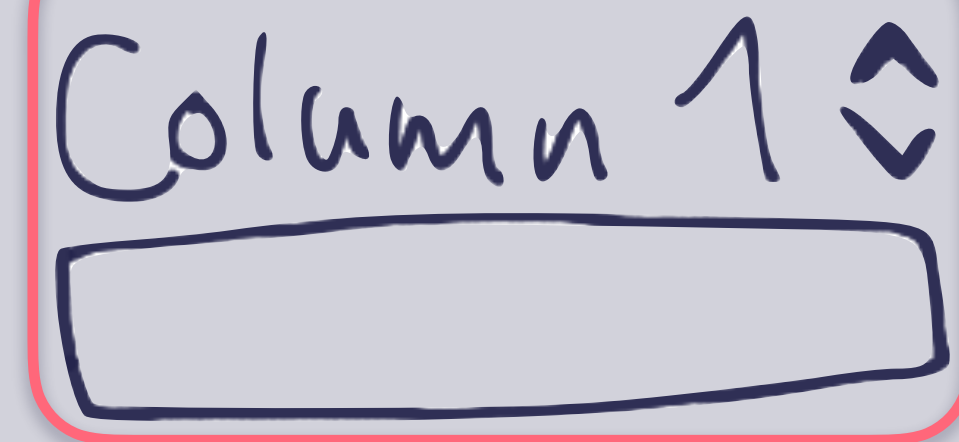
Table Header Component

Step 2: make it pretty

Define a CSS Grid for Layout



```
.tabelle-header {
  display: grid;
  grid-template-areas:
    "header arrow-asc"
    "header arrow-desc"
    "search search";
  grid-template-columns: 1fr auto;
}
```



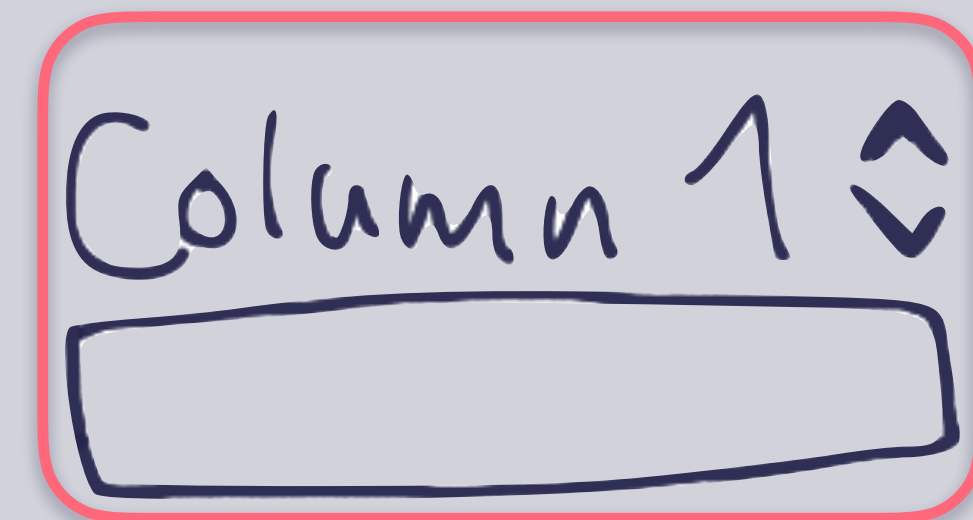
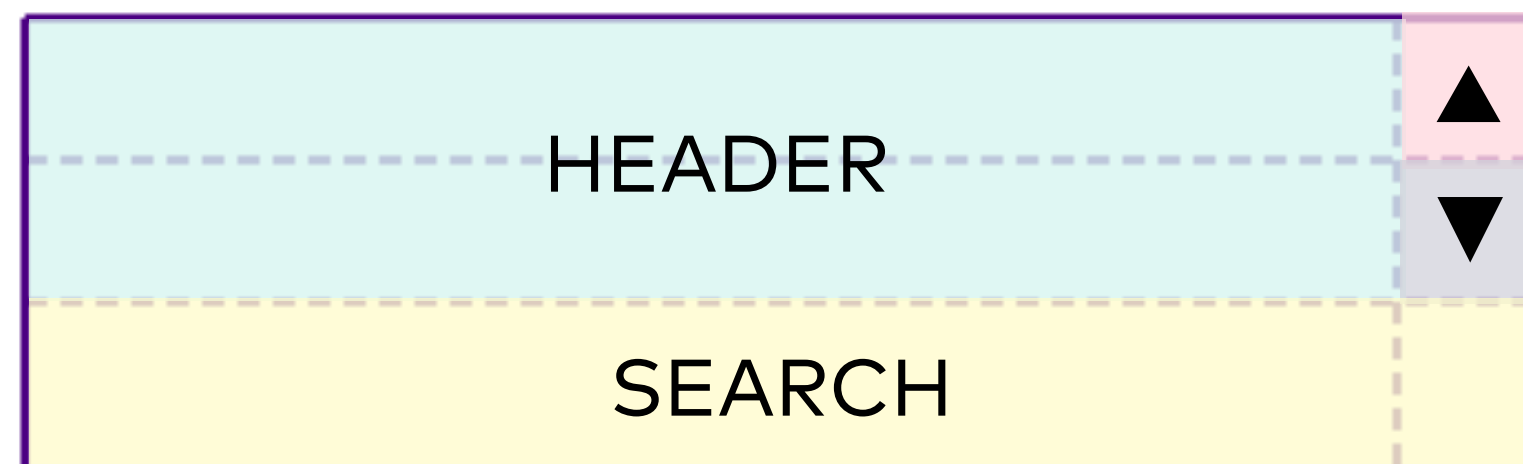


Table Header Component

Step 2: make it pretty

Place children items in grid



```
.tabelle-header {  
  grid-template-areas:  
    "header arrow-asc"  
    "header arrow-desc"  
    "search search";  
}
```

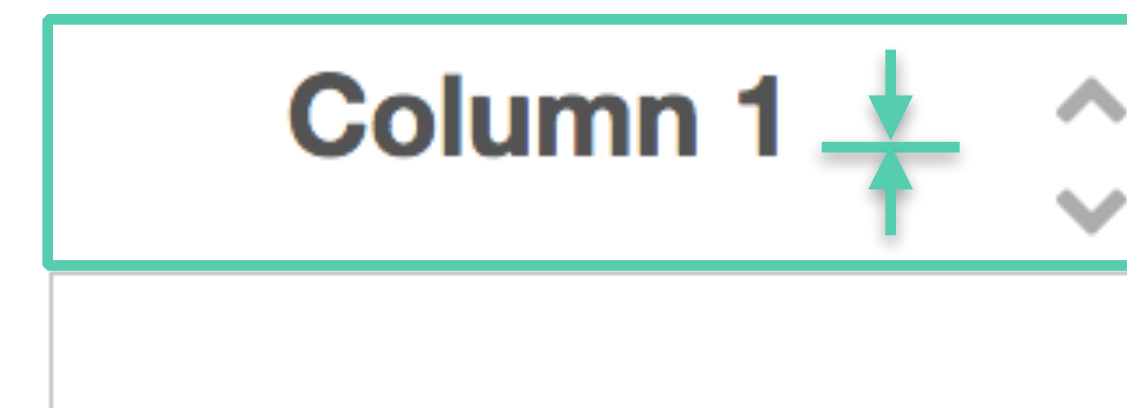
```
.tabelle-header {  
  .header {  
    grid-area: header;  
  }  
  .arrow-asc {  
    grid-area: arrow-asc;  
  }  
  .arrow-desc {  
    grid-area: arrow-desc;  
  }  
  .tabelle-input {  
    grid-area: search;  
  }  
}
```

Table Header Component

Step 2: make it pretty

Add new styling classes

```
<table>
  <th class="tabelle-header"
    role="group" aria-labelledby="col2g">
    <span class="header" id="col2g">Column 1</span>
    ...
  </th>
</table>
```



Almost there!!!



Tip #9

With CSS Grid or Flexbox, vertical alignment is *finally* easy!

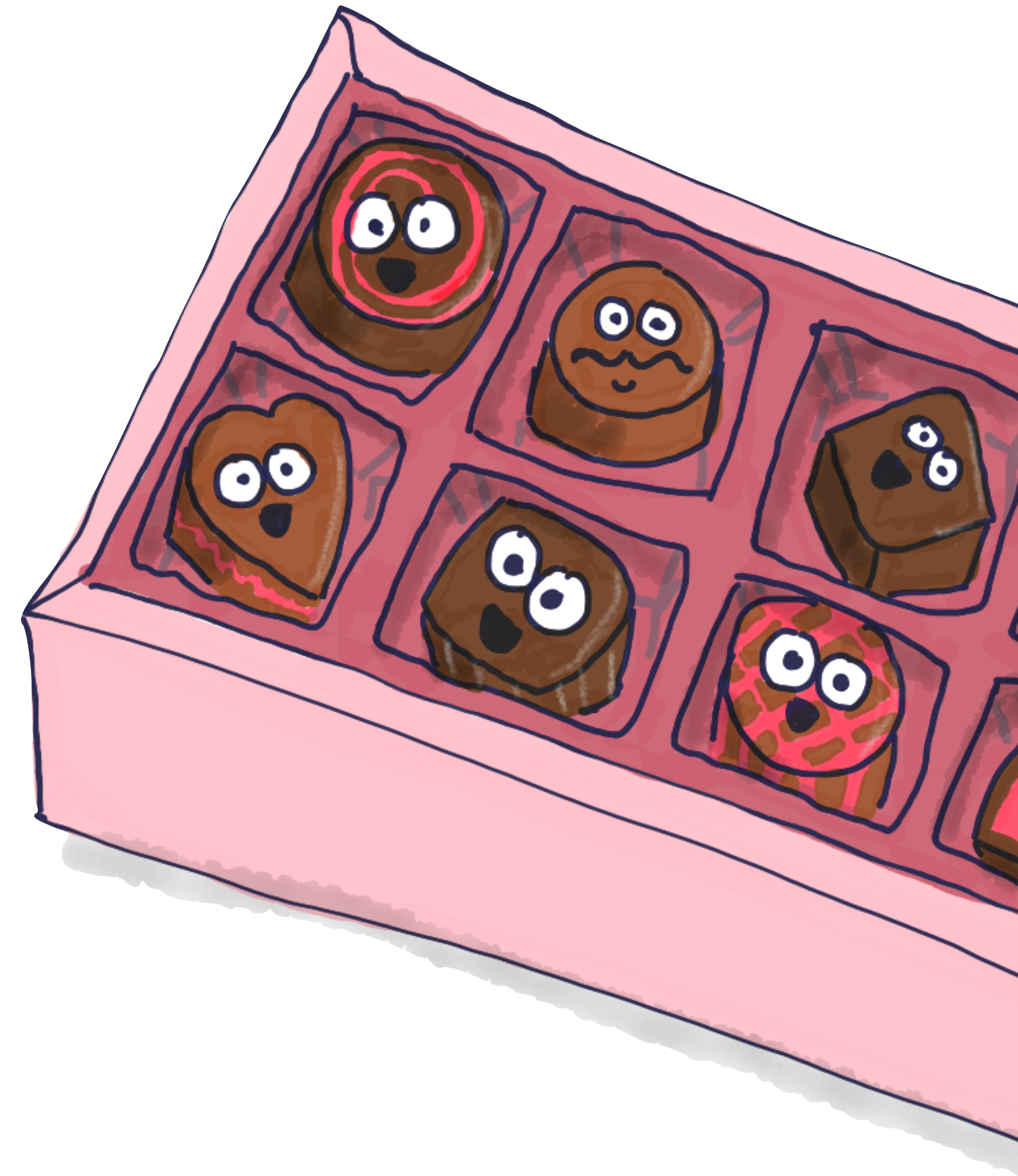


Table Header Component

Step 2: make it pretty

Center table header vertically



```
.tabelle-header {  
  .header {  
    align-self: center;  
  }  
}
```



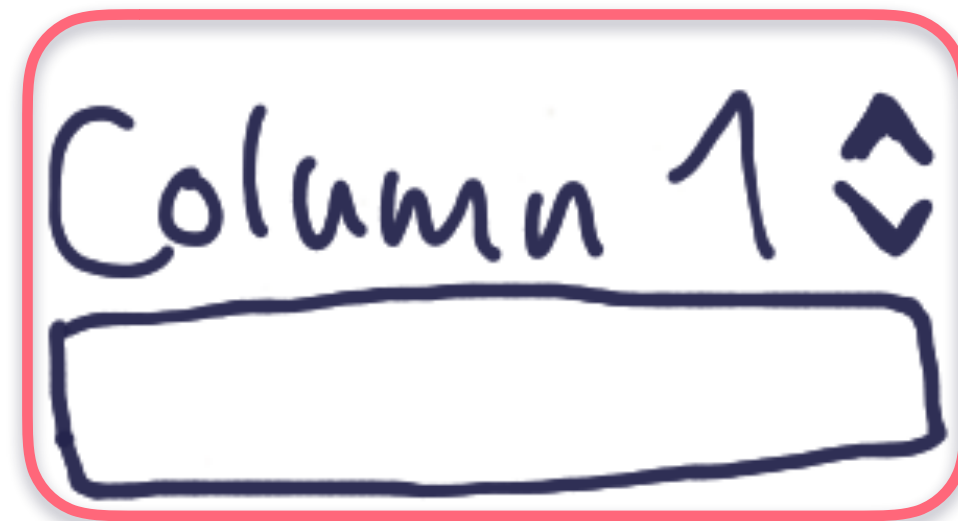


Table Header Component

Our HTML Template for reuse



```
/* tabelle-header.jsx */
function TabelleHeader ({ name, label }, ...kids) {
  return <div class="tabelle-header"
    role="group" aria-labelledby="{name}_group">
    <span class="header" id="{name}_group" >
      {label}
    </span>
    <Arrows name={name} label={name} />
    {kids
      ? <Select name={name} label={name}>{kids}<Select>
      : <Input name={name} label={name} />}
    </div>
  }
}
```



? JS?

make it better?

✓ CSS

it's pretty!

✓ HTML

it works
FOR ALL USERS!



JS

no need!



CSS

it's pretty!



HTML

it works
FOR ALL USERS!

L
A
S
T

C
O
M
P
O
N
E
N
T



make it work

Tabelle

Step 1: make it work

```
<form action="/search">
  <table>
    <thead>
      <tr>
        <th>
          <TabelleHeader name="col1" label="Column 1" />
        </th>
        <th>
          <TabelleHeader name="col2" label="Column 2" />
        </th>
        <th>
          <TabelleHeader name="col3" label="Column 3" />
        </th>
      </tr>
    </thead>
    <tbody>...</tbody>
  </table>
  <button type="submit">Perform Search</button>
</form>
```

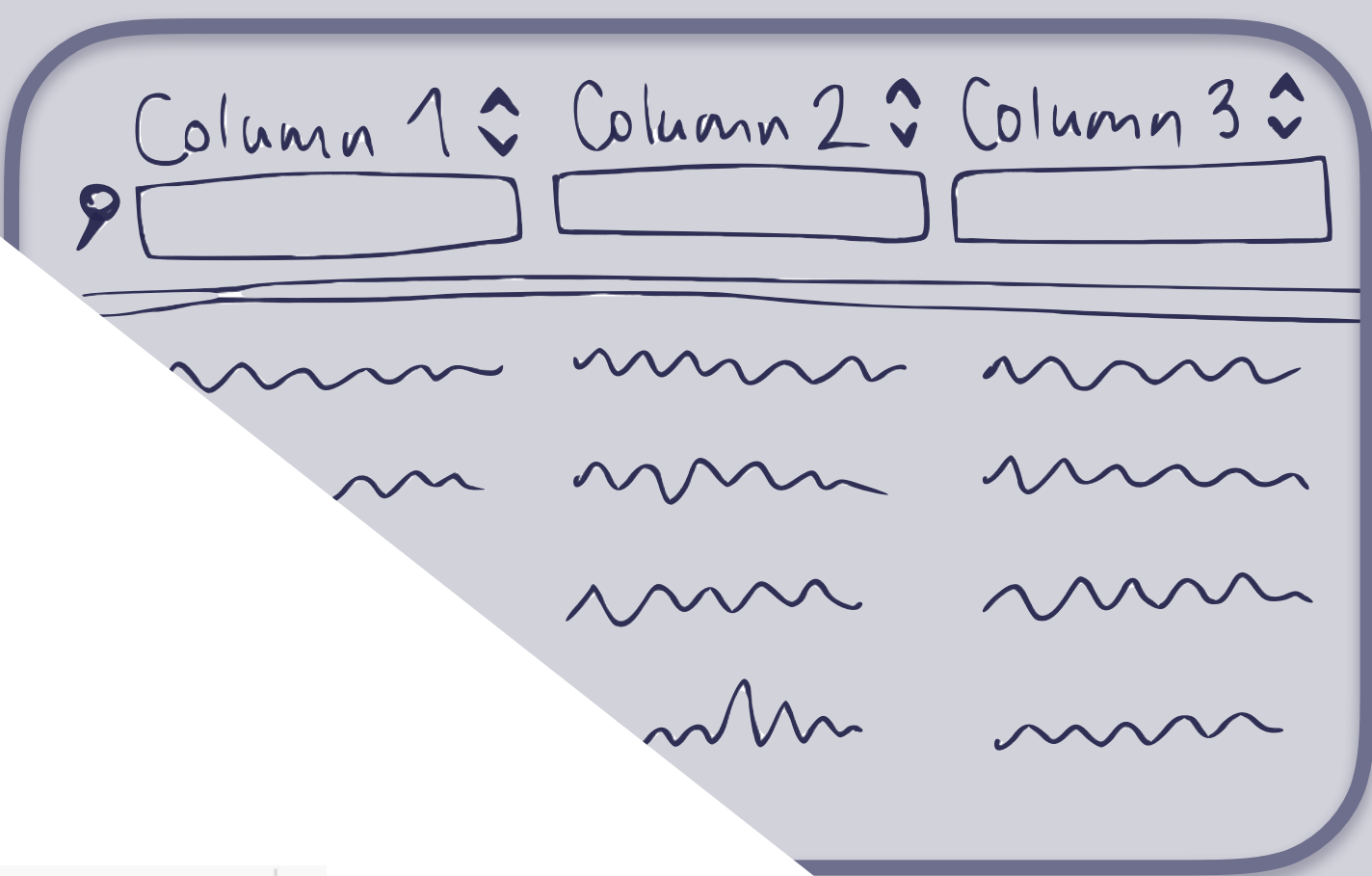
Wrap in a form so that we can submit user filter queries

Tabelle

Step 1: make it work



The magic of FORMS



⬅

➡

🔄

🏠

localhost:3000

⋮

🔒

⬇

⌵

📄

💬

🌐

⛶

ABP

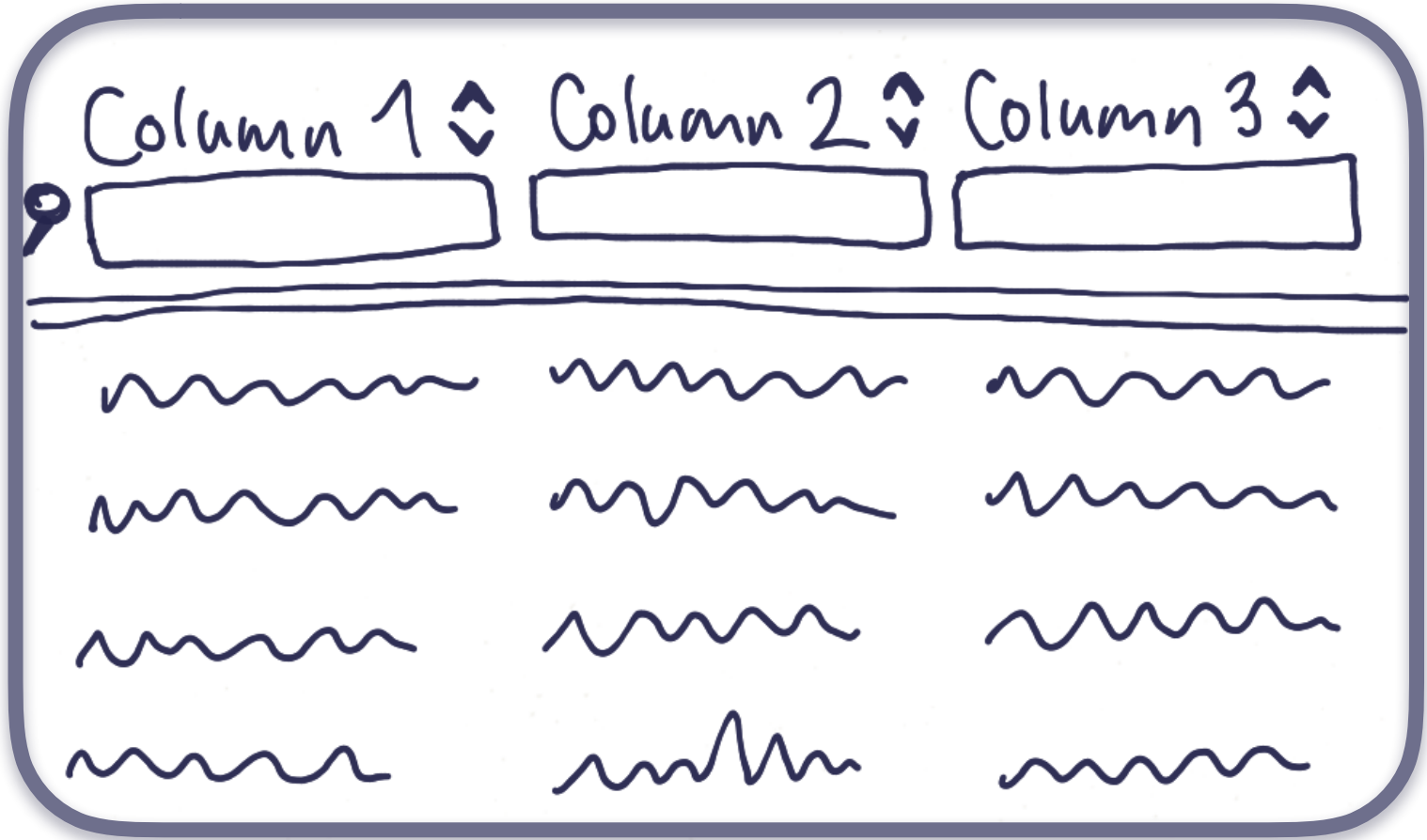
🛡

➡➡

⋮

People

Name	Phone	Email	Address
Jeremy Kutch V	185-342-4903 x729	renayshields@bogan.name	7843 Haywood Drive Moenmouth, 53084-5375 Liberia
Noelia Treutel	(721) 756-2848 x444	stewartbecker@okon.biz	827 Carita Spring South Jerold, 65969 Taiwan
Julia Daugherty Sr.	693.904.6223	murray@pfefferwalter.org	75038 Hermann Keys Lake Kathyrn, 05745 Finland
Bradly Wuckert	882.469.7537 x5484	malcolmsmitham@gradybrekke.org	86628 Zulauf Meadow Terenceton, 41074 Lesotho
Delia O'Hara	392.556.7103	genarobuckridge@mcdermott.biz	39098 Stefan Rest North Janette, 31321 Palau



localhost:3000

Name	Phone	Email	Address
Jeremy Kutch V	185-342-4903 x729	renayshields@bogan.name	7843 Haywood Drive Moenmouth, 53084-5375 Liberia
Noelia Treutel	(721) 756-2848 x444	stewartbecker@okon.biz	827 Carita Spring South Jerold, 65969 Taiwan
Julia Daugherty Sr.	693.904.6223	murray@pfefferwalter.org	75038 Hermann Keys Lake Kathryn, 05745 Finland
Bradly Wuckert	882.469.7537 x5484	malcolmsmitham@gradybrekke.org	86628 Zulauf Meadow Terenceton, 41074 Lesotho
Della O'Hara	392.556.7103	genarobuckridge@mcdermott.biz	39098 Stefan Rest North Janette, 31321 Palau



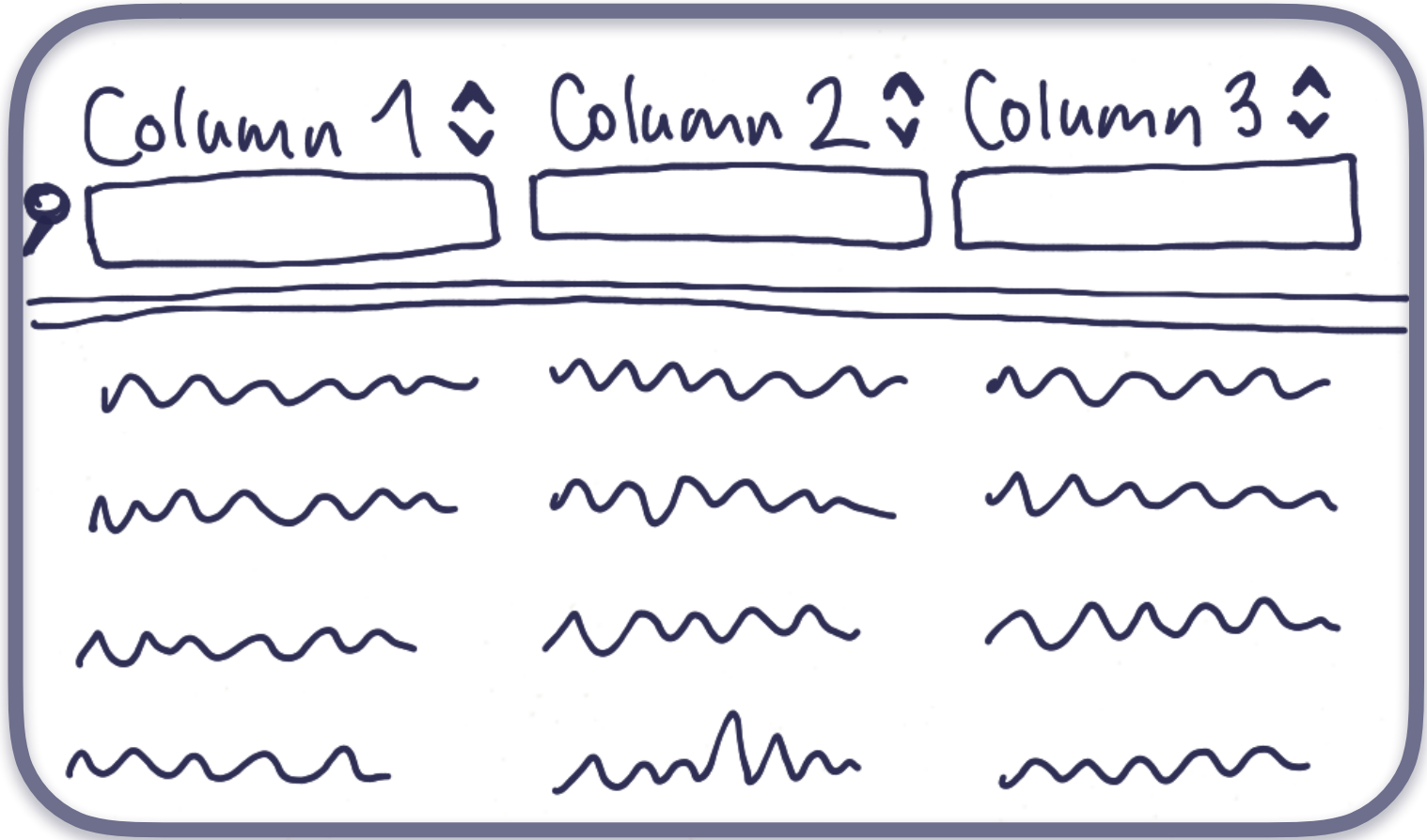
CSS?

make it pretty?



HTML

*it works
FOR ALL USERS!*



localhost:3000

Name	Phone	Email	Address
Jeremy Kutch V	185-342-4903 x729	renayshields@bogan.name	7843 Haywood Drive Moenmouth, 53084-5375 Liberia
Noelia Treutel	(721) 756-2848 x444	stewartbecker@okon.biz	827 Carita Spring South Jerold, 65969 Taiwan
Julia Daugherty Sr.	693.904.6223	murray@pfefferwalter.org	75038 Hermann Keys Lake Kathryn, 05745 Finland
Bradly Wuckert	882.469.7537 x5484	malcolmsmitham@gradybrekke.org	86628 Zulauf Meadow Terenceton, 41074 Lesotho
Della O'Hara	392.556.7103	genarobuckridge@mcdermott.biz	39098 Stefan Rest North Janette, 31321 Palau



CSS

*someone else can
make it pretty!*



HTML

*it works
FOR ALL USERS!*

make it better

Tip #10

Submit forms
asynchronously and replace
your DOM with the result
from the server

Tabelle

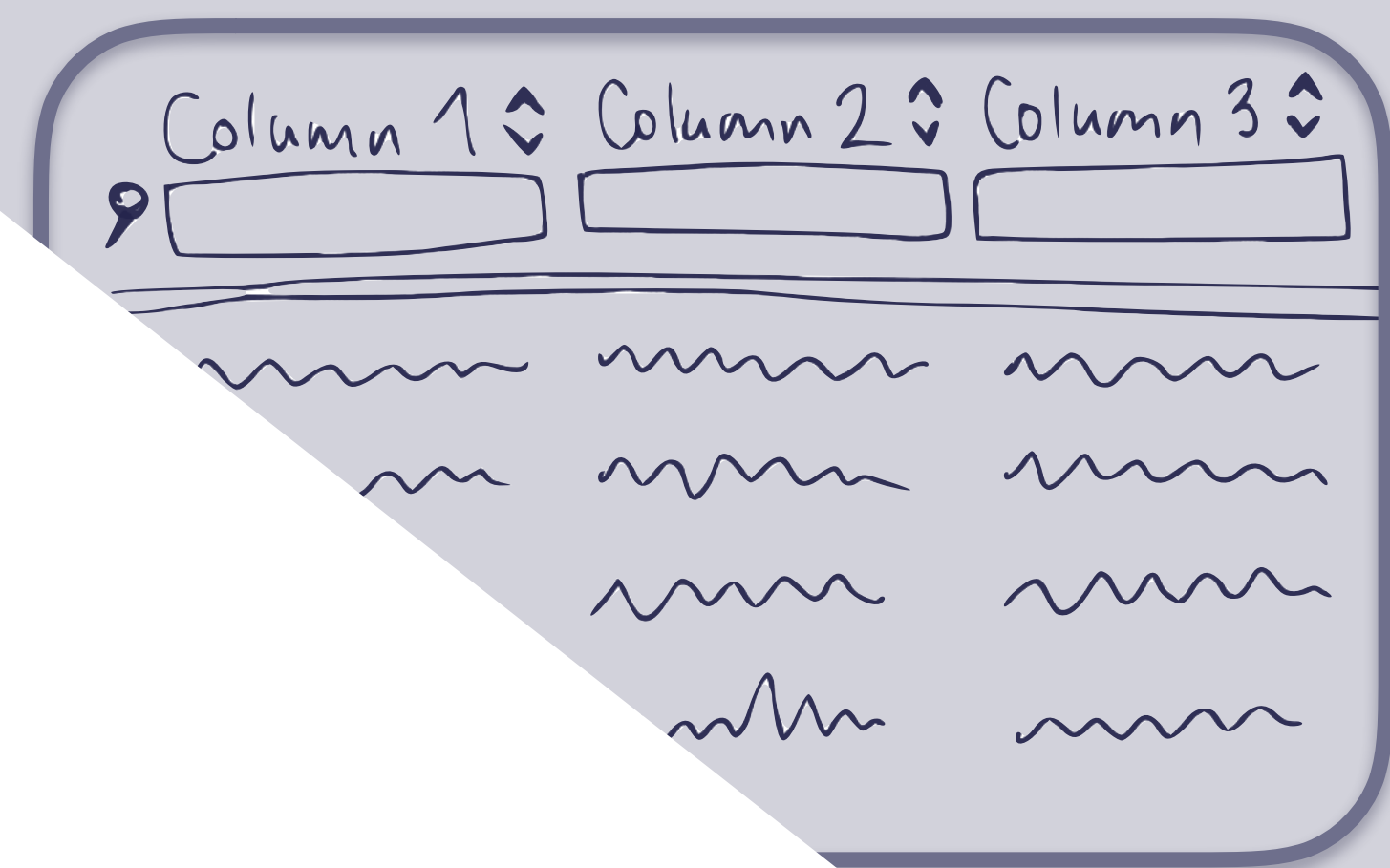
Step 3: make it better

Submitting forms asynchronously

```
// stringify form data as `application/x-www-form-urlencoded`  
function serializeForm (form) { ... }  
  
function submit (form) {  
  const uri = serializeForm(form)  
  return fetch(uri)  
    .then(response => response.text())  
}
```

⚠️ *Only the application happy path is being considered here.*

Source: <https://github.com/FND/uitil>



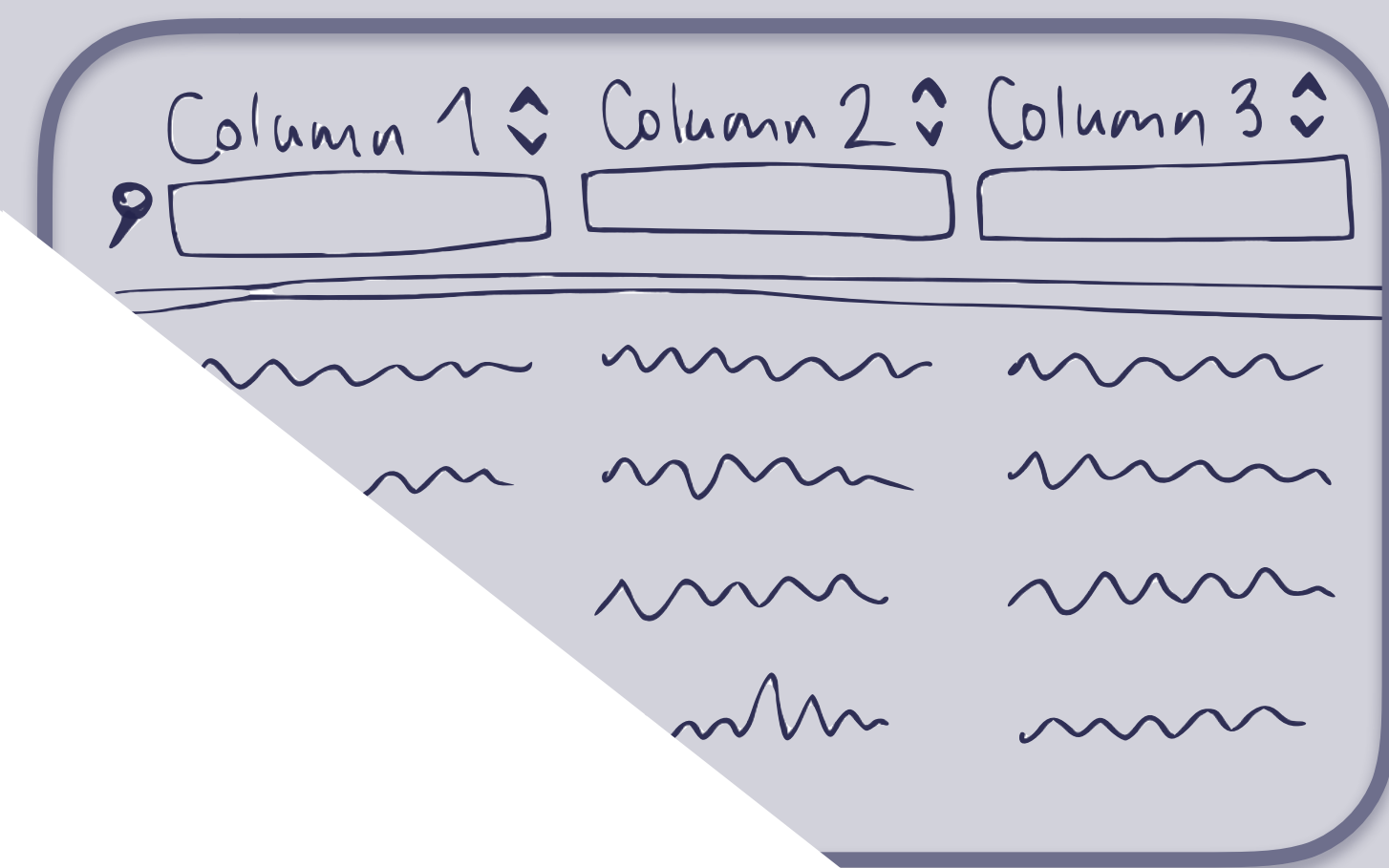
Tabelle

Step 3: make it better

Replacing the tbody with response HTML

```
function template2dom (htmlString, selector) {  
  let tmp = document.createElement('template')  
  tmp.innerHTML = htmlString.trim()  
  return tmp.content.querySelector(selector)  
}  
  
function replaceTbody (tbody, htmlResponse) {  
  let newTbody = template2dom(htmlResponse, 'tbody')  
  tbody.innerHTML = newTbody.innerHTML  
}
```

⚠ Only the application happy path is being considered here.



Tabelle

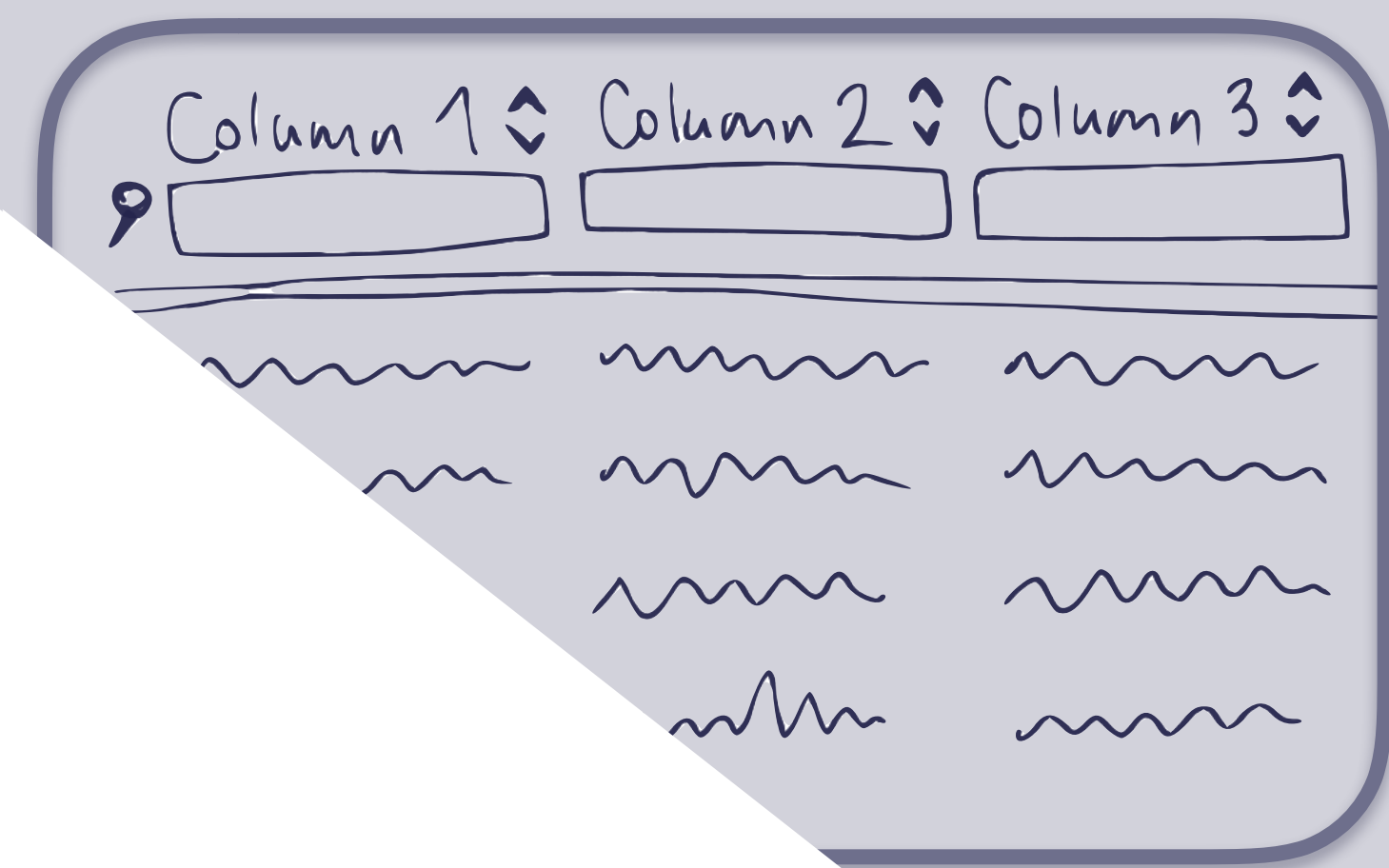
Step 3: make it better

Override default submit behavior

```
function doSubmit (component, form) {  
  let tbody = component.querySelector('tbody')  
  submit(form)  
    .then(html => replaceTBody(tbody, html))  
}  
  
function initializeSubmit (component, form) {  
  form.addEventListener('submit', ev => {  
    doSubmit(component, form)  
    ev.preventDefault()  
  })  
}
```

⚠️ *Only the application happy path is being considered here.*

Who calls the initialize function?



Historically:

adding a component
dynamically used to require
BOTH the HTML Markup and
a function to initialize it

Tip #11

**Use Custom Elements to define
a custom HTML Element AND
define how to initialize it.**

The browser will then initialize
your component for you!

Tabelle

Step 3: make it better

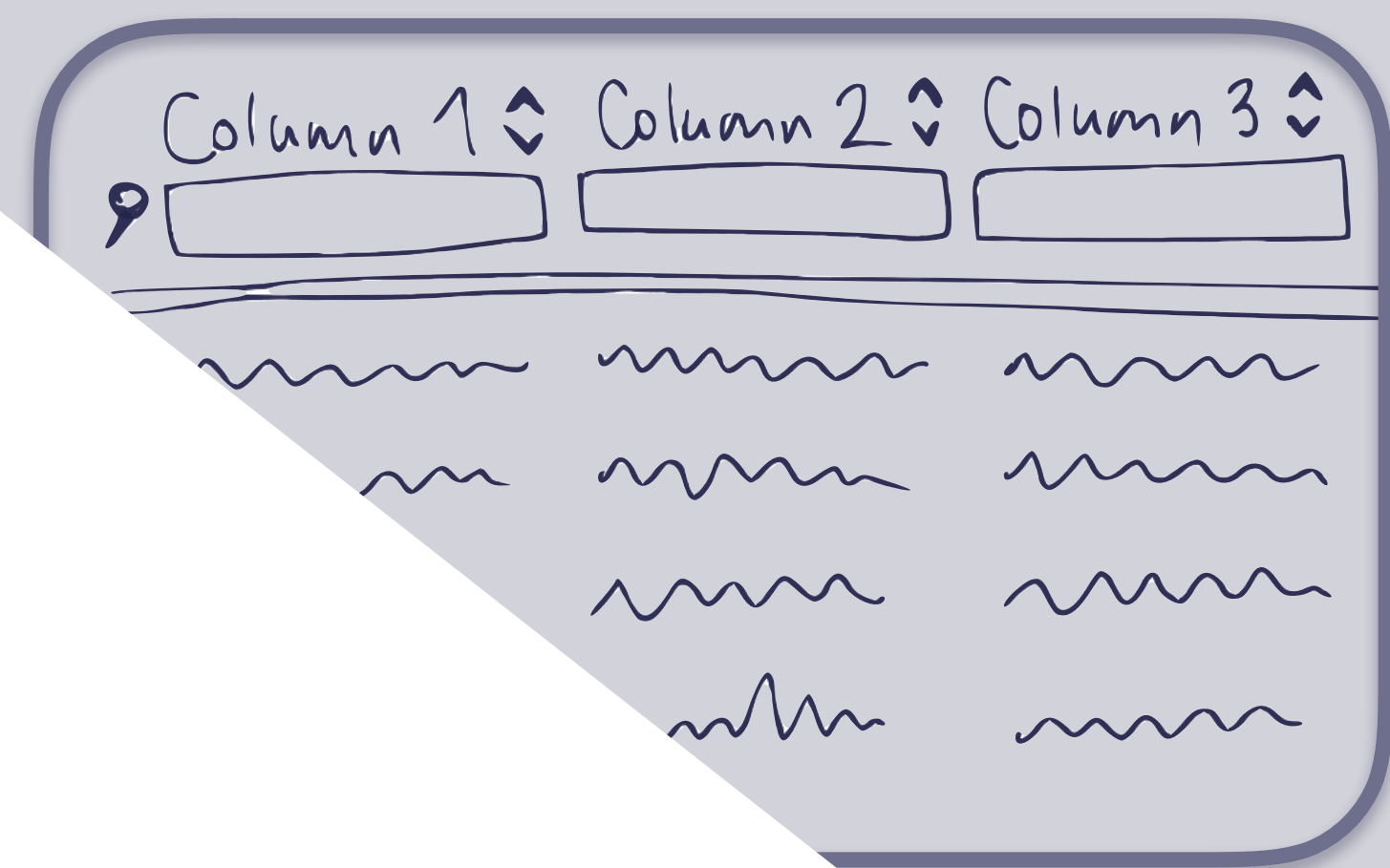
Custom Elements allow you to define how the component needs to be initialized.

```
class Tabelle extends HTMLElement {  
  connectedCallback () {  
    let form = this.form  
    initializeSubmit(this, form)  
  }  
  get form () {  
    return this.querySelector('form')  
  }  
}  
customElements.define('ta-belle', Tabelle)
```

```
<ta-belle>  
...  
</ta-belle>
```



As soon as **ta-belle** appears in the DOM, `connectedCallback` will be called!

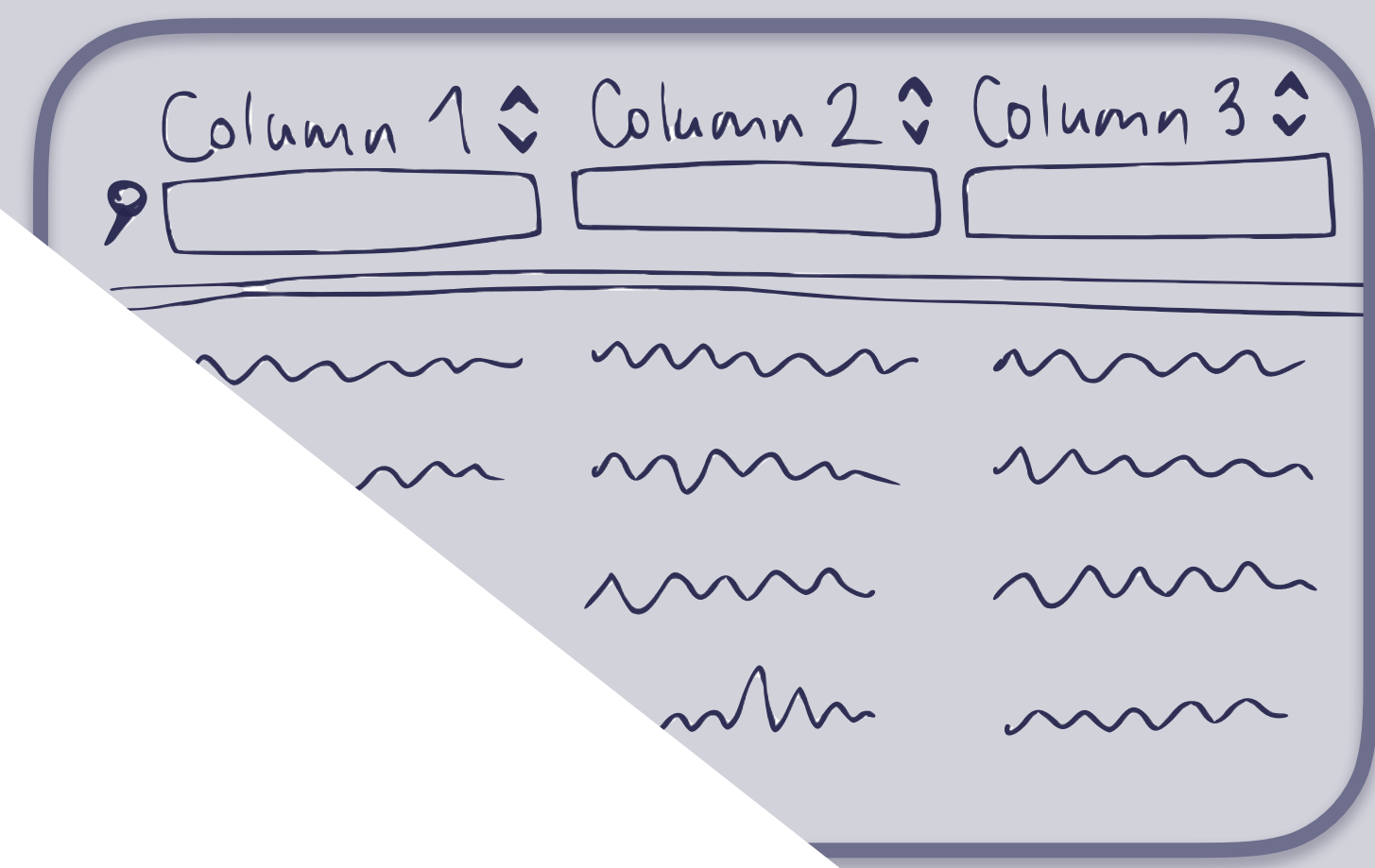


Tabelle

Step 3: make it better

Custom Elements provide scope:
this is bound to the HTML Element itself
and it's native JavaScript API

```
class Tabelle extends HTMLElement {  
  connectedCallback () {  
    let form = this.form  
    initializeSubmit(this, form)  
  }  
  get form () {  
    return this.querySelector('form')  
  }  
}  
customElements.define('ta-belle', Tabelle)
```



Tabelle

Step 3: make it better

Make the implementation more dynamic by triggering submits when an input element changes

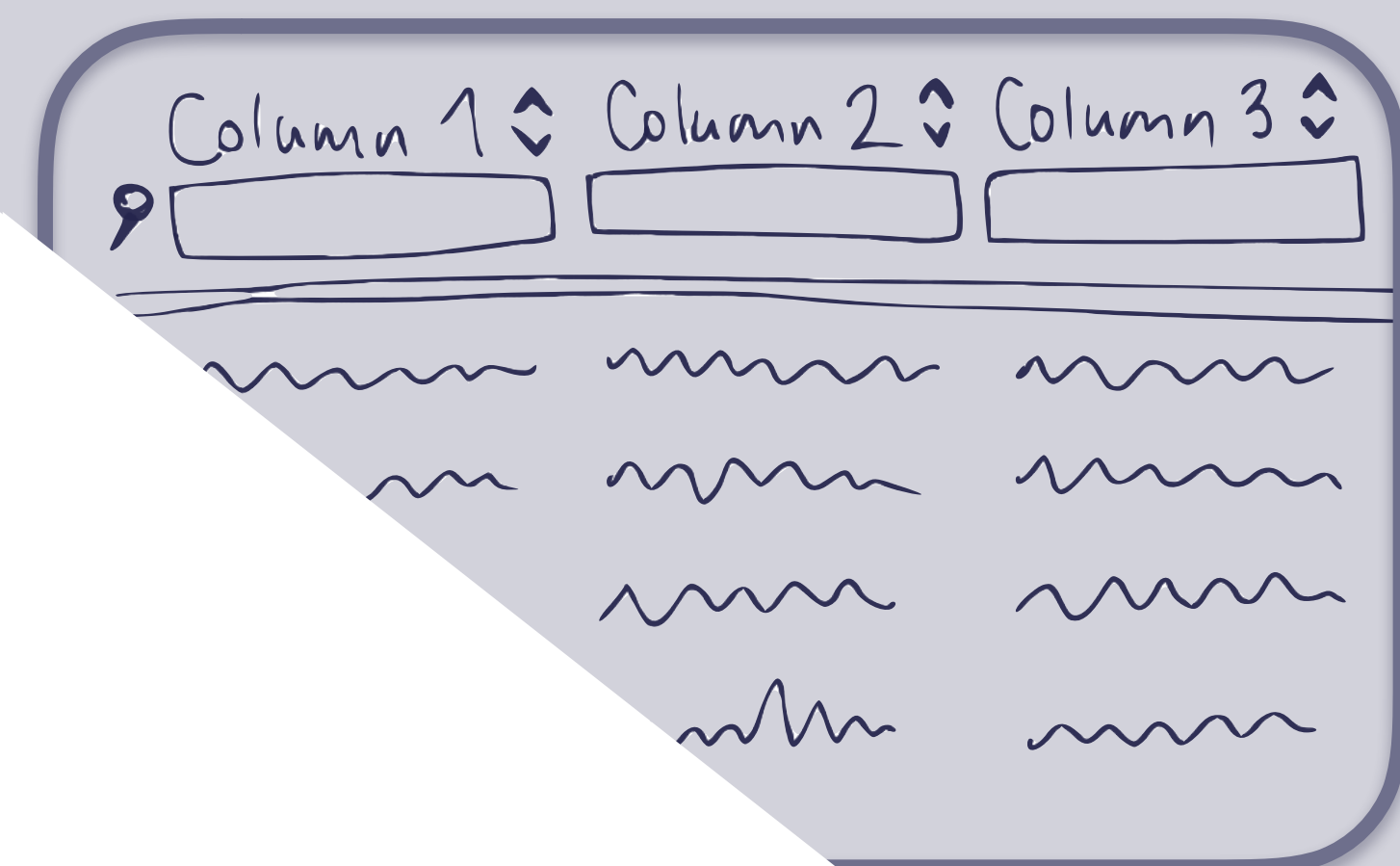
```
import { debounce } from 'util'

class Tabelle extends HTMLElement {
  connectedCallback () {
    ...
    form.addEventListener('change', () => doSubmit(this, form))
    form.addEventListener('keyup',
      debounce(300, ev => doSubmit(this, form)))
  }
}
```

Make sure to debounce 'keyup's for input[type=text] elements!

<https://davidwalsh.name/javascript-debounce-function>

⚠ Only the application happy path is being considered here.



...and much more...

TabelleRailsExample

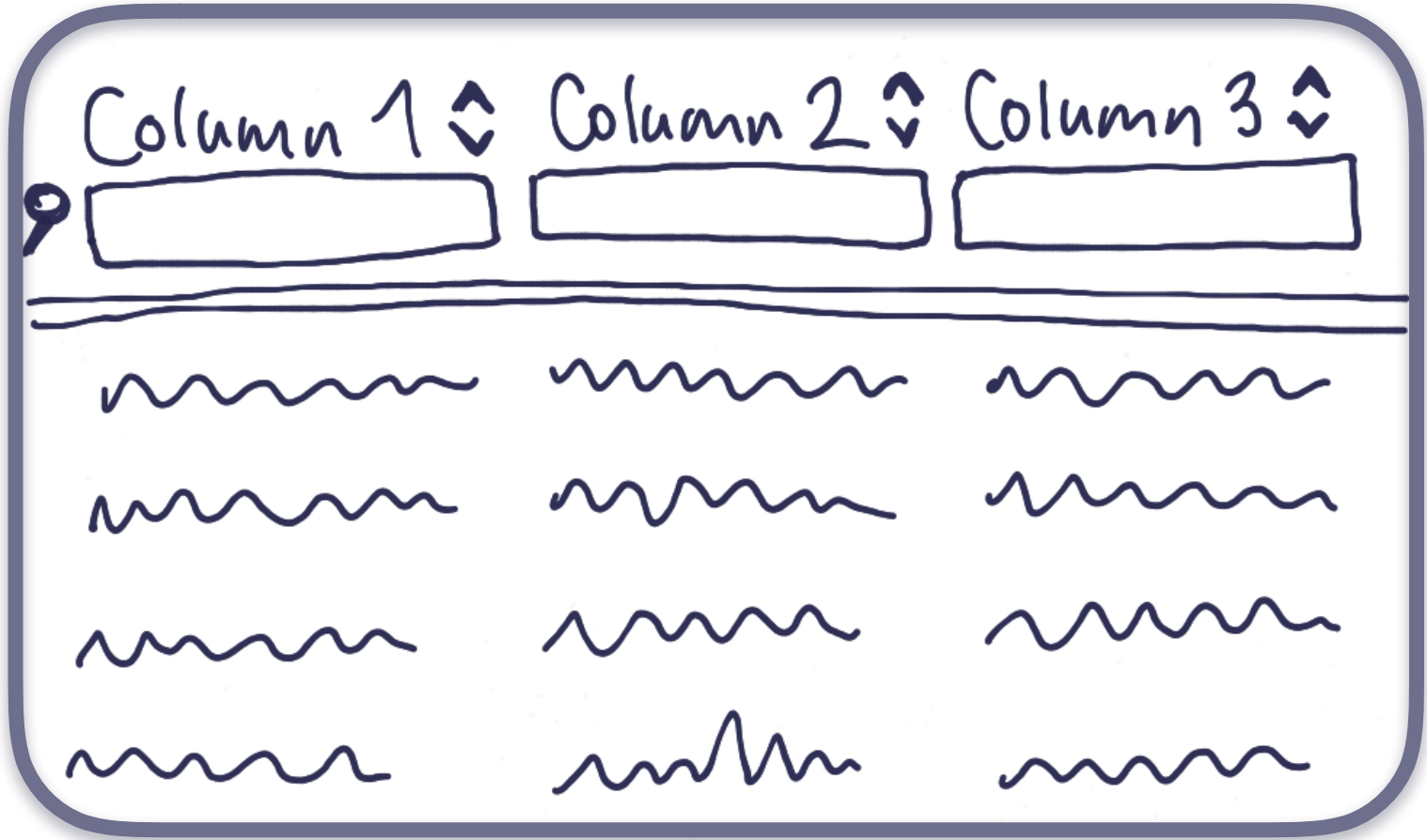
https://tabelle-rails-example.herokuapp.com

Work

ABP

People

Name	Phone	Email	Address
			4561 Kling Courts New Chester, 58948-2023 Iraq
Kit Glover	509.257.2793 x4916	angie@robel.info	578 Ruecker Via Domenicchester, 95234-3854 Nicaragua
Florrie Skiles	1-751-059-1176	mariano@casper.io	29524 Mertz Ridges Meldatown, 16774-7278 Indonesia
Graig Dibbert	133.422.1767	kristalebert@mayertgorczany.net	92301 Leandro Fields West Loni, 90077-3488 Jamaica
Cecilia Von	390-242-0512 x16166	merle@marks.io	170 Alix Causeway New Brantview, 38570 Czech Republic
Latonya Reichert	(806) 153-2423	allen@nicolas.co	7544 Rigoberto Pines Malcolmfurt, 67893-5783 Japan
Rueben Cummings	604.534.9325 x525	garry@hammesmcclure.org	187 Schuster Parks Lake Alfonzo, 42853 Albania
Nadine Hayes	(990) 978-5937 x56483	wesherzog@farrell.co	163 Kimberly Plains Port Vincenzo, 58659 Mexico

A screenshot of a web browser displaying a table of people data. The browser's address bar shows the URL "https://tabelle-rails-example.herokuapp.com". The table has four columns: "Name", "Phone", "Email", and "Address". The data is as follows:

Name	Phone	Email	Address
Willard Mertz DVM	508-768-5842	buster@windler.io	4561 Kling Courts New Chester, 58948-2023 Iraq
Kit Glover	509.257.2793 x4916	angle@robel.info	578 Ruecker Via Domenicchester, 95234-3854 Nicaragua
Florrie Skiles	1-751-059-1176	mariano@casper.io	29524 Mertz Ridges Meldatown, 16774-7278 Indonesia
Graig Dibbert	133.422.1767	kristalebert@mayertgorczany.net	92301 Leandro Fields West Loni, 90077-3488 Jamaica
Cecilia Von	390-242-0512 x16166	merle@marks.io	170 Alix Causeway New Brantview, 38570 Czech Republic
Latonya Reichert	(806) 153-2423	allen@nicolas.co	7544 Rigoberto Pines Malcolmfurt, 67893-5783 Japan
Rueben Cummings	604.534.9325 x525	garry@hammesmcclure.org	187 Schuster Parks Lake Alfonzo, 42853 Albania
Nadine Hayes	(990) 978-5937 x56483	wesherzog@farrell.co	163 Kimberly Plains Port Vincenzo, 58659 Mexico



JS

faster & dynamic



CSS

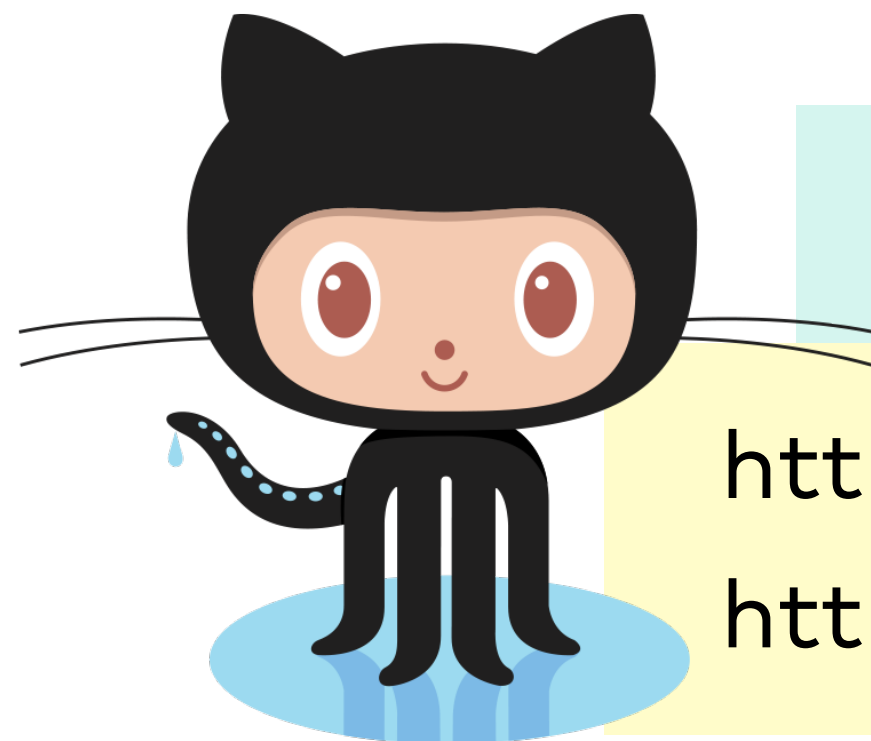
it's pretty!



HTML

it works
FOR ALL USERS!

Thank you!



<https://tabelle-rails-example.herokuapp.com/>

<https://github.com/innoq/tabelle>

https://github.com/innoq/tabelle_rails_example

