

Webudvikling Frontend

CSS Grid, JavaScript & the DOM

Webudvikling

kea
KØBENHAVNS ERHVERVSAKADEMI

Agenda

- CSS Grid
- JavaScript basics
 - Variabler, Data Types, Template String,
 - Objects, Arrays & loops
 - if-else Statements
 - DOM Manipulation

The screenshot shows a web application titled "Person Array" displaying four profiles in a 2x2 grid. Each profile consists of a photo, a name, a title, and an email link.

Profile	Name	Title	Email
Top Left	Birgitte Kirk Iversen	Senior Lecturer	bki@mail.dk
Top Right	Martin Aagaard Nøhr	Lecturer	mnor@mail.dk
Bottom Left	Rasmus Cederdorff	Senior Lecturer	race@mail.dk
Bottom Right	Dan Okkels Brendstrup	Lecturer	dob@mail.dk

- CSS Grid med personer
- Styling og Media queries
- “Dynamisk” indhold genereret via JavaScript
- Variabler
- Array med objekter med properties
- For-of-loop
- DOM-Manipulation
- Template String / Backtick String

CSS Layouts

Introduction to CSS layout

Normal flow

The display property

Flexbox

Grid

Floats

Positioning

Table layout

Multiple-column layout

Grid & Flexbox

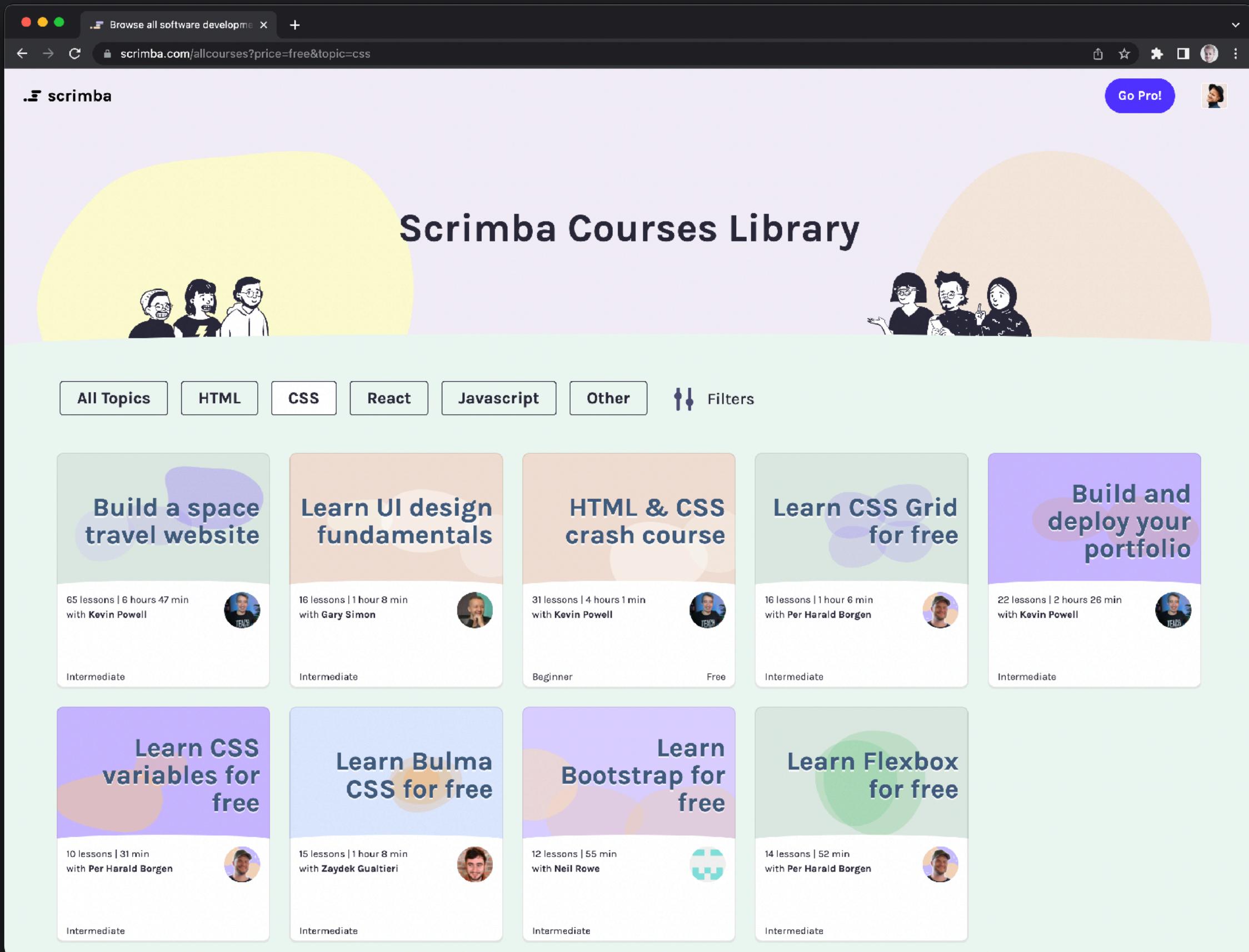
... two CSS layout models that can be used to create layouts with just a couple of CSS rules.

... responsive and flexible layouts.

Scrimba Courses

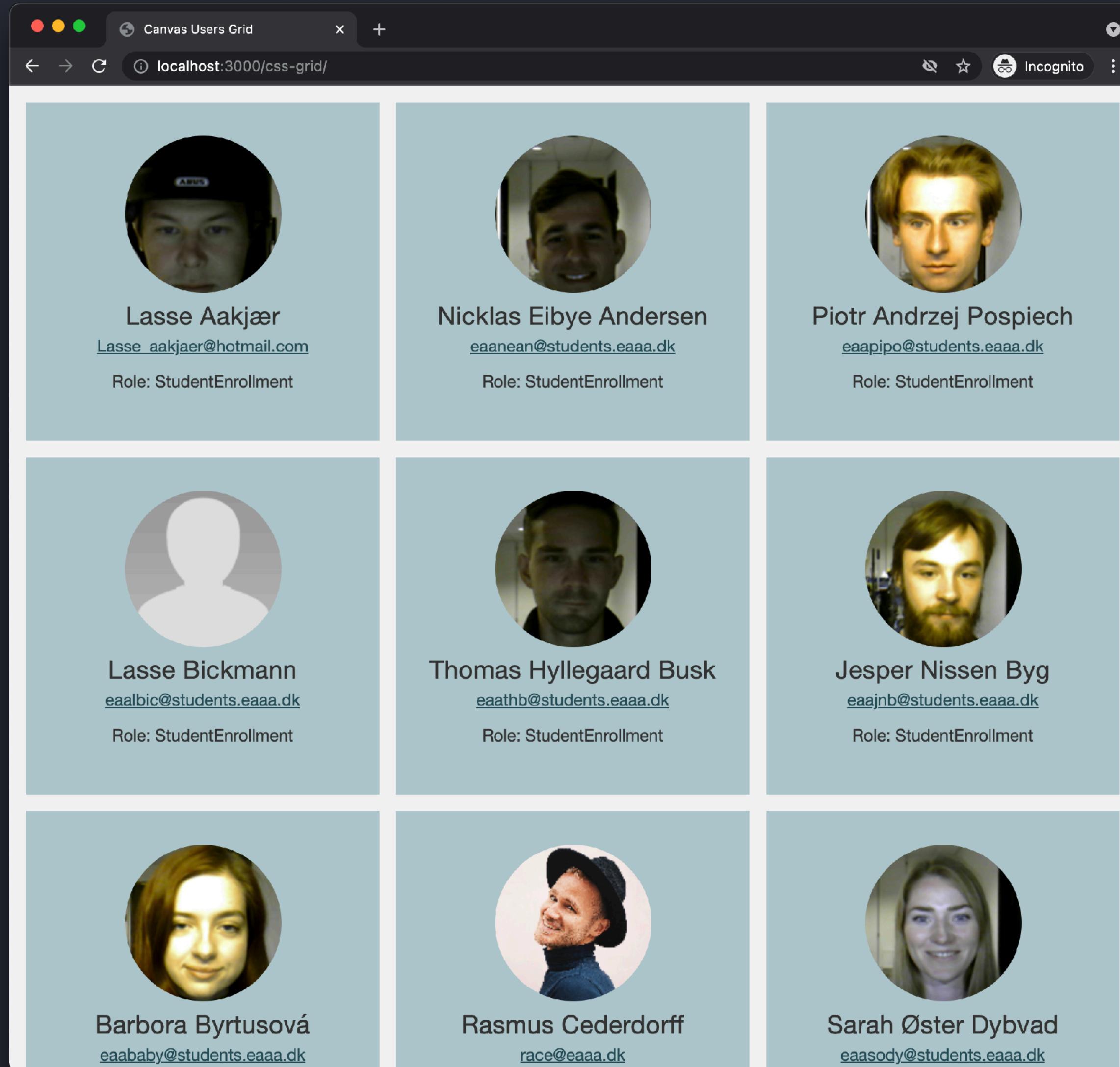
Learn CSS Grid for free: <https://scrimba.com/learn/cssgrid>

Learn Flexbox for free: <https://scrimba.com/learn/flexbox>



<https://scrimba.com/allcourses>

CSS Grid



```
<section id="users" class="grid"></section>
```

```
/* ----- grid styling ----- */
```

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

```
@media (min-width: 600px) {  
  .grid {  
    grid-template-columns: 1fr 1fr;  
  }  
}
```

```
@media (min-width: 992px) {  
  .grid {  
    grid-template-columns: 1fr 1fr 1fr;  
  }  
}
```

CSS Grid

The screenshot shows a web page titled "Person Array". The content is organized into a 2x2 grid of cards, each containing a portrait photo and details about a person:

- Birgitte Kirk Iversen**: Senior Lecturer, bki@mail.dk
- Martin Aagaard Nøhr**: Lecturer, mnor@mail.dk
- Rasmus Cederdorff**: Senior Lecturer, race@mail.dk
- Dan Okkels Brendstrup**: Lecturer, dob@mail.dk

A developer tools sidebar is visible on the right, showing the DOM structure and CSS styles for the grid container.

```
<section id="content" class="grid-container"></section>
```

```
/* ----- grid container styling ----- */
.grid-container {
  display: grid;
  grid-template-columns: 1fr;
  gap: 1em;
}

@media (min-width: 600px) {
  .grid-container {
    grid-template-columns: 1fr 1fr;
  }
}

@media (min-width: 992px) {
  .grid-container {
    grid-template-columns: 1fr 1fr 1fr;
  }
}
```

```
section#content.grid-container 910x740.25
cederdorff.github.io/web-diplom-frontend/array-persons/
```

```
Elements > 1 | ⚙ | ⋮ | X
```

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body class="page">
    <header>...</header>
    ... <section id="content" class="grid-container">...</section> (grid) = $0
      <!-- main js file -->
      <script src="js/main.js"></script>
    </body>
</html>
```

```
html body.page section#content.grid-container
```

```
Styles Computed Layout Event Listeners >
```

```
Filter :hover .cls + □ □
```

```
element.style {
```

```
@media (min-width: 600px)
  .grid-container {
    grid-template-columns: 1fr 1fr; main.css:132
  }
  .grid-container { main.css:125
    display: grid; □□
    grid-template-columns: 1fr;
    gap: 1em;
  }
  section { user agent stylesheet
    display: block;
  }
```

```
Inherited from body.page
```

```
body { main.css:12
  color: var(--text-color-dark);
  padding: 0;
  margin: 0;
```

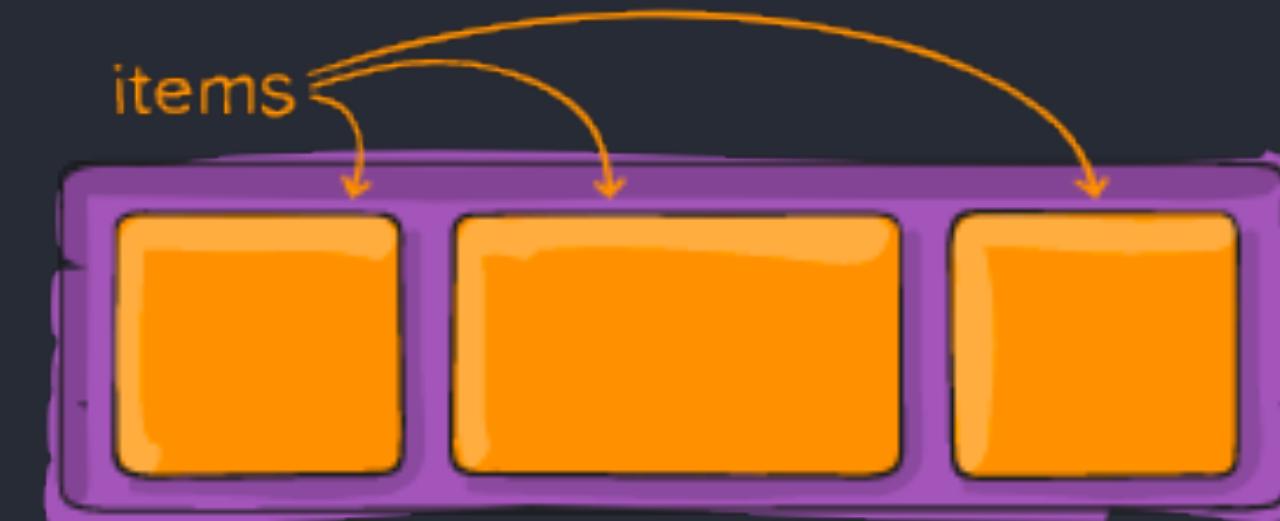
CSS GRID



```
/* ----- grid container styling ----- */
.grid-container {
  display: grid;
  grid-template-columns: 1fr;
  gap: 1em;
}

@media (min-width: 600px) {
  .grid-container {
    grid-template-columns: 1fr 1fr;
  }
}

@media (min-width: 992px) {
  .grid-container {
    grid-template-columns: 1fr 1fr 1fr;
  }
}
```



```
.grid-container > article {
  box-shadow: 1px 1px 8px □rgba(0, 0, 0, 0.25);
  padding: 1.5em;
  background-color: var(--white);
  transition: 0.5s;
  animation: fadeIn 0.5s;
}

.grid-container > article:hover {
  box-shadow: 0 8px 16px 0 var(--green);
}

.grid-container > article img {
  width: 100%;
  height: 200px;
  object-fit: cover;
}
```

Grid Container

CSS Grid Layout

w3schools.com/css/css_grid.asp

HTML CSS JAVASCRIPT

Display Property

An HTML element becomes a grid container when its `display` property is set to `grid` or `inline-grid`.

Example

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

[Try it Yourself »](#)

Example

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

[Try it Yourself »](#)

All direct children of the grid container automatically become *grid items*.

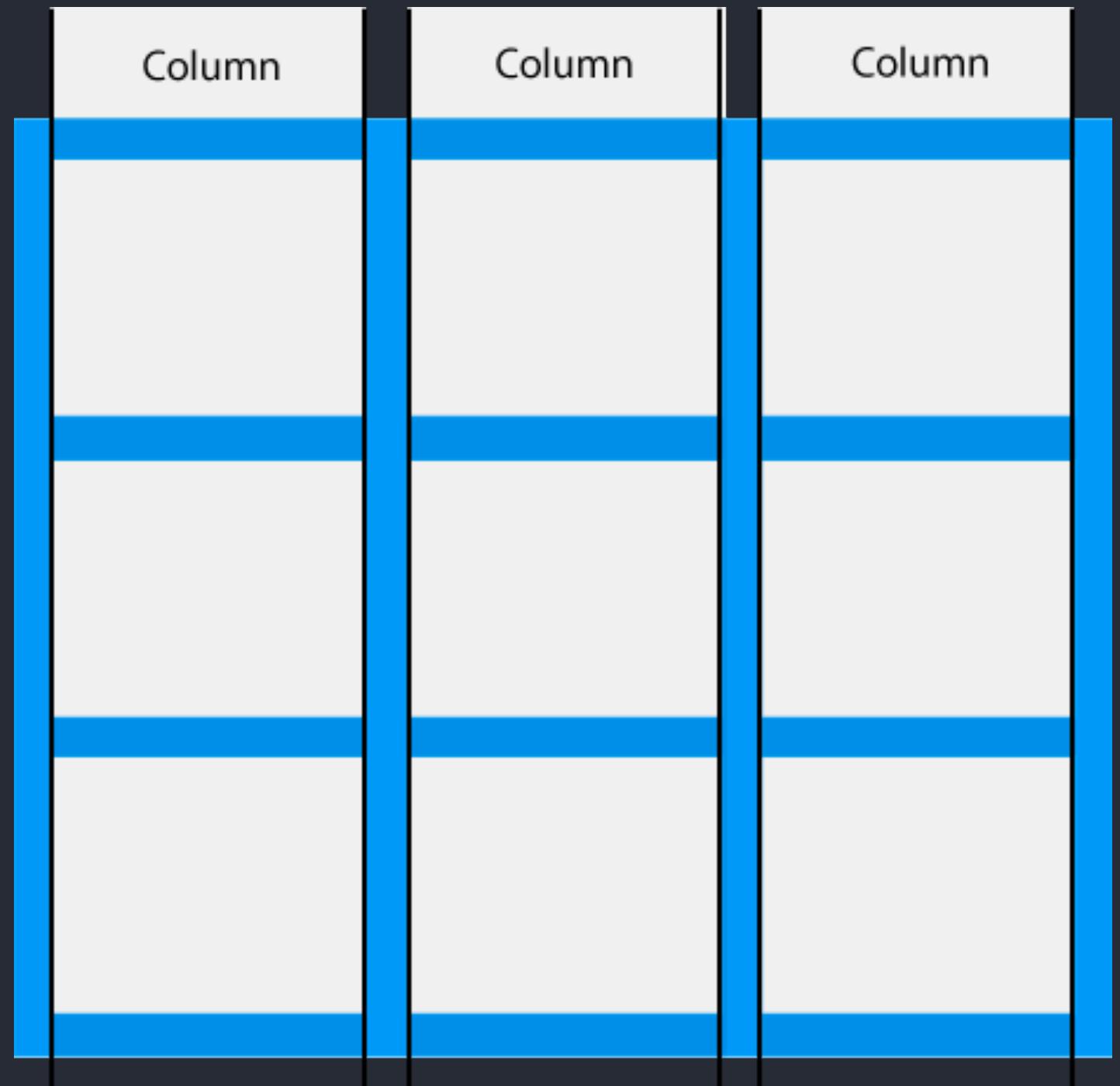
Example

```
<div class="grid-container">  
  <div class="grid-item">1</div>  
  <div class="grid-item">2</div>  
  <div class="grid-item">3</div>  
  <div class="grid-item">4</div>  
  <div class="grid-item">5</div>  
  <div class="grid-item">6</div>  
  <div class="grid-item">7</div>  
  <div class="grid-item">8</div>  
  <div class="grid-item">9</div>  
</div>
```



Grid Columns

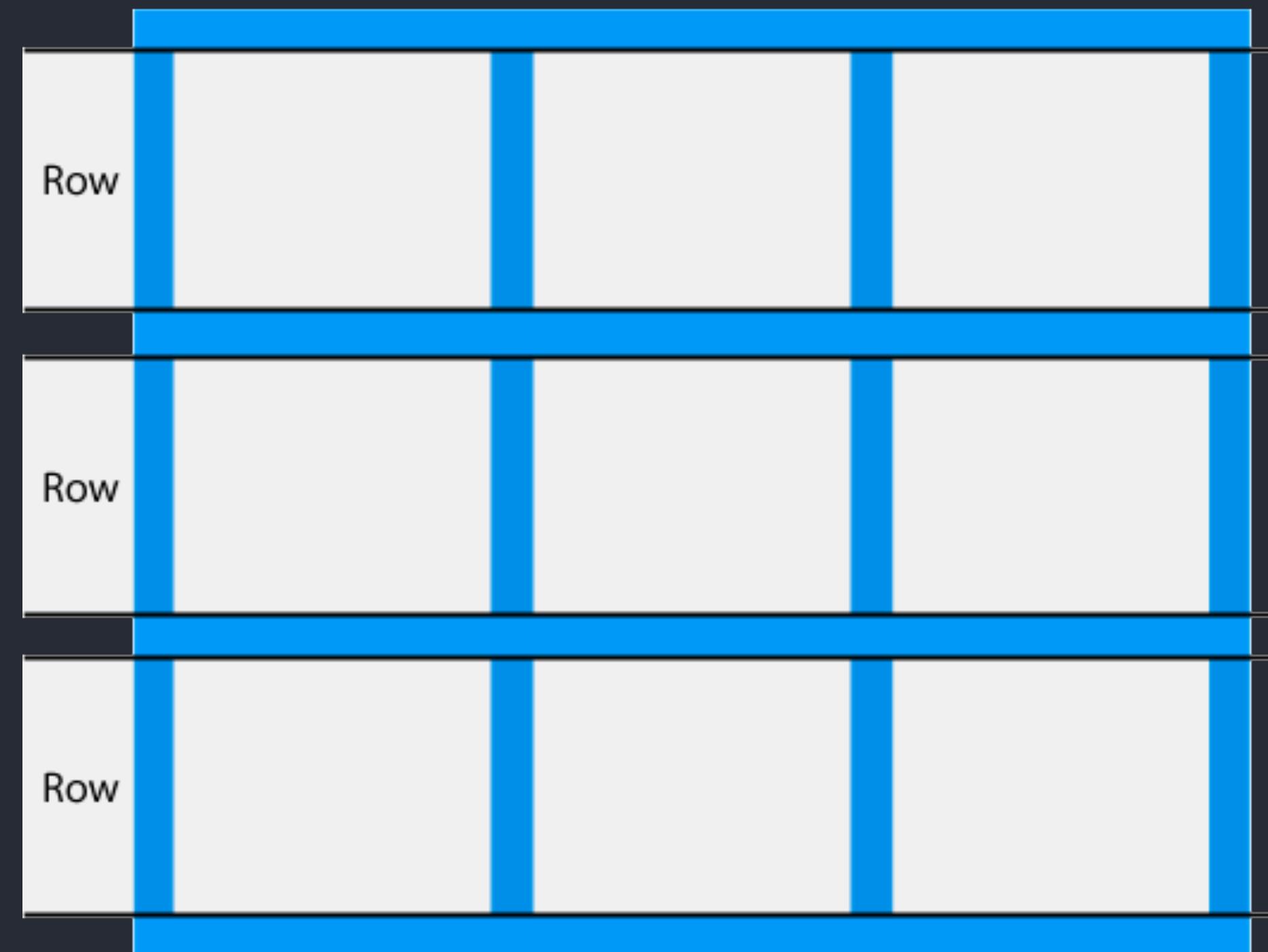
Vertical Lines



```
.grid-container {  
    display: grid;  
    grid-template-columns: auto auto auto auto;  
}
```

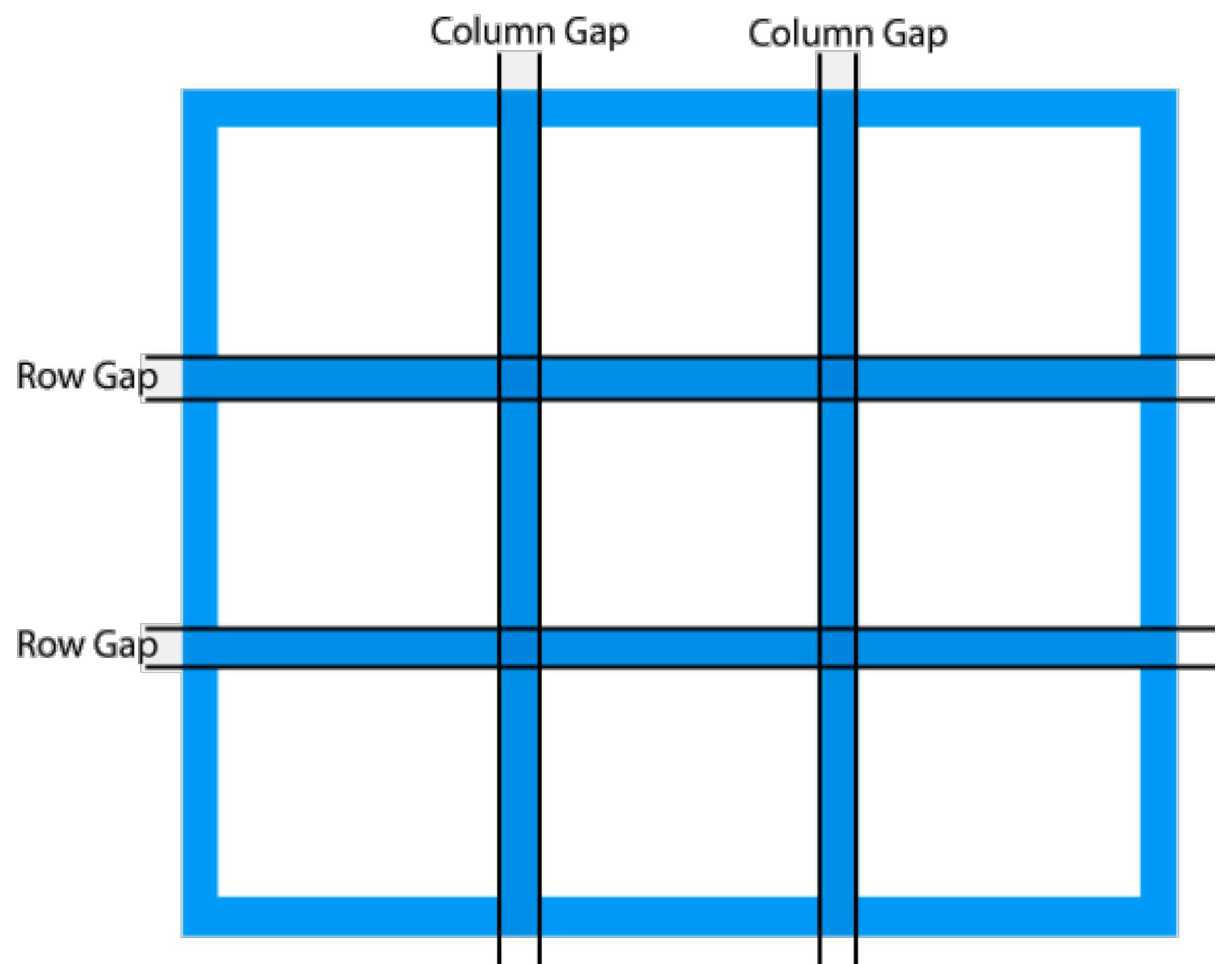
Grid Rows

Horisontal Lines



```
.grid-container {  
    display: grid;  
    grid-template-rows: 100px 300px;  
}
```

Grid Gap



The `column-gap` property sets the gap between the columns:

```
.grid-container {  
    display: grid;  
    column-gap: 50px;  
}
```

The `row-gap` property sets the gap between the rows:

```
.grid-container {  
    display: grid;  
    row-gap: 50px;  
}
```

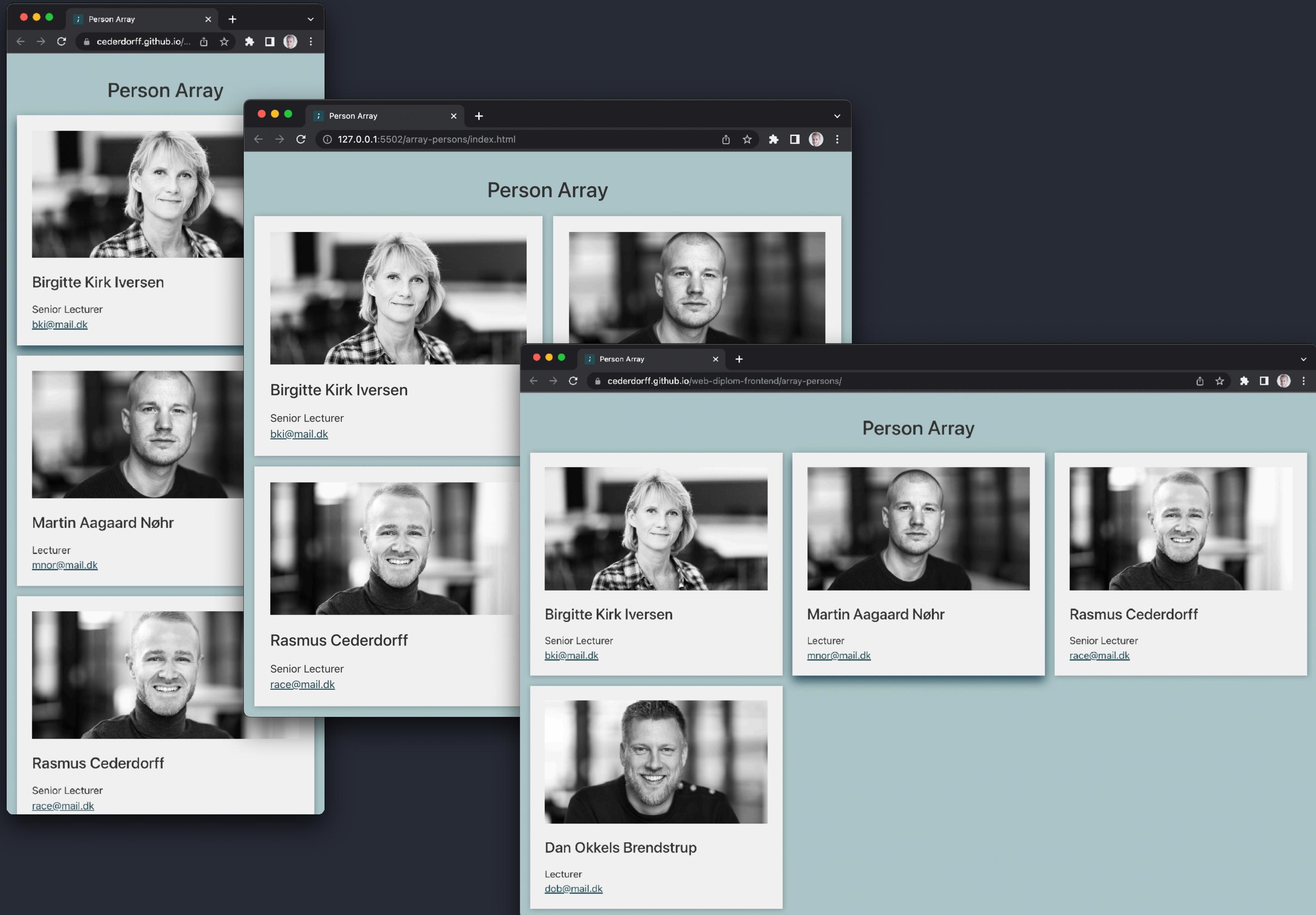
The `gap` property is a shorthand property for the `row-gap` and the `column-gap` properties:

```
.grid-container {  
    display: grid;  
    gap: 50px 100px;  
}
```

The `gap` property can also be used to set both the row gap and the column gap in one value:

```
.grid-container {  
    display: grid;  
    gap: 50px;  
}
```

CSS Grid & media queries

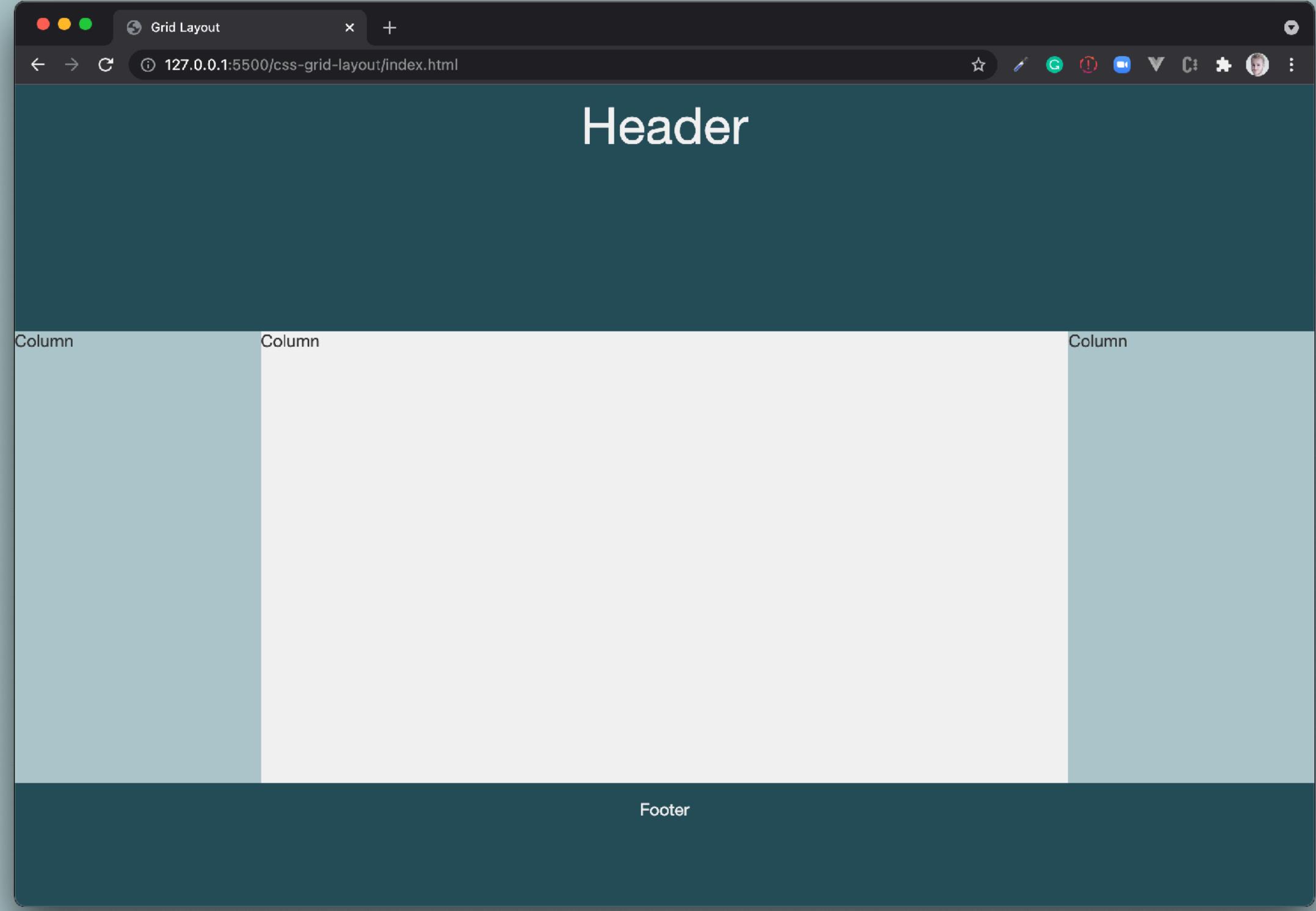


array-persons

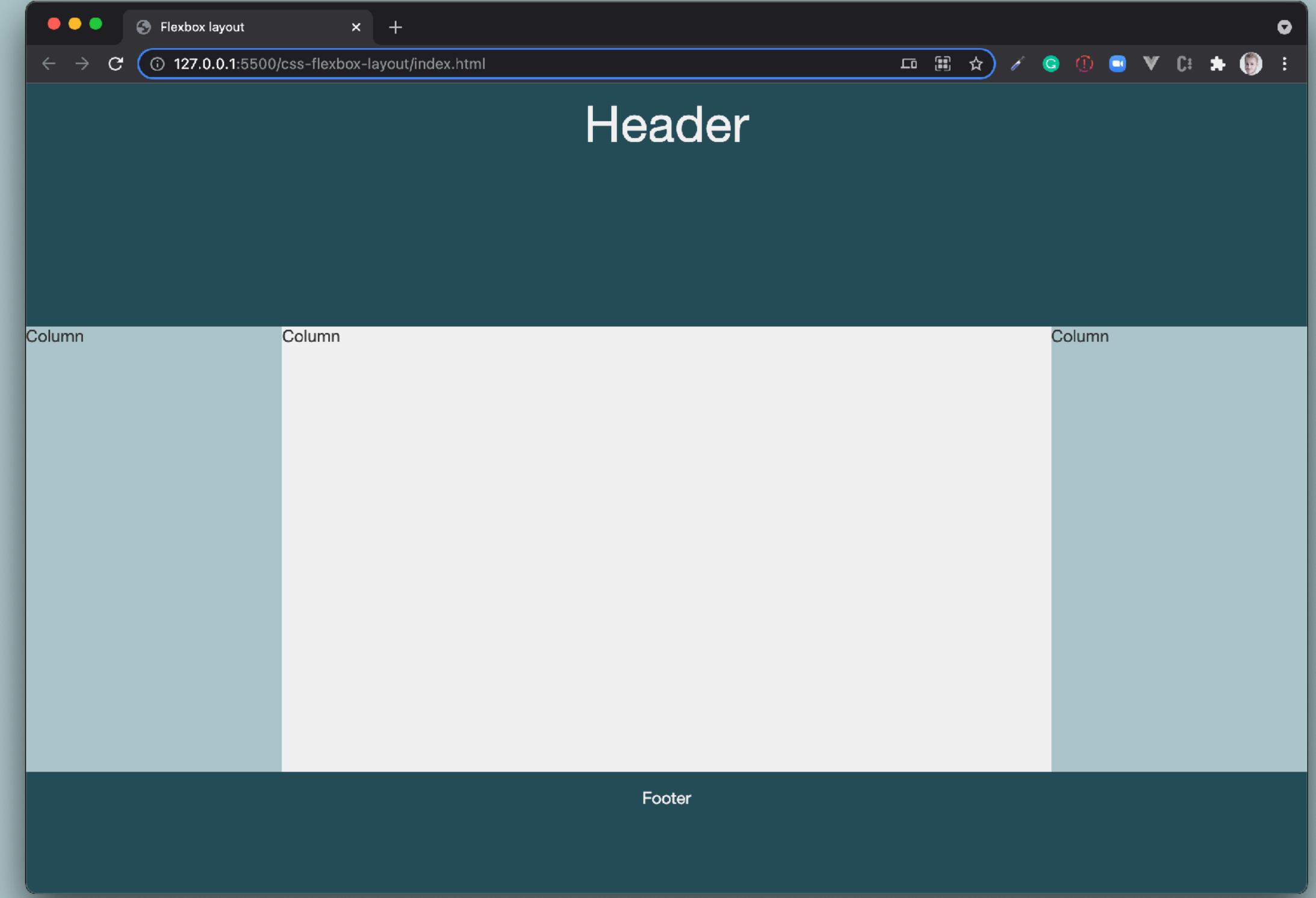
```
/* ----- grid container styling ----- */
.grid-container {
  display: grid;
  grid-template-columns: 1fr;
  gap: 1em;
}

@media (min-width: 600px) {
  .grid-container {
    grid-template-columns: 1fr 1fr;
  }
}

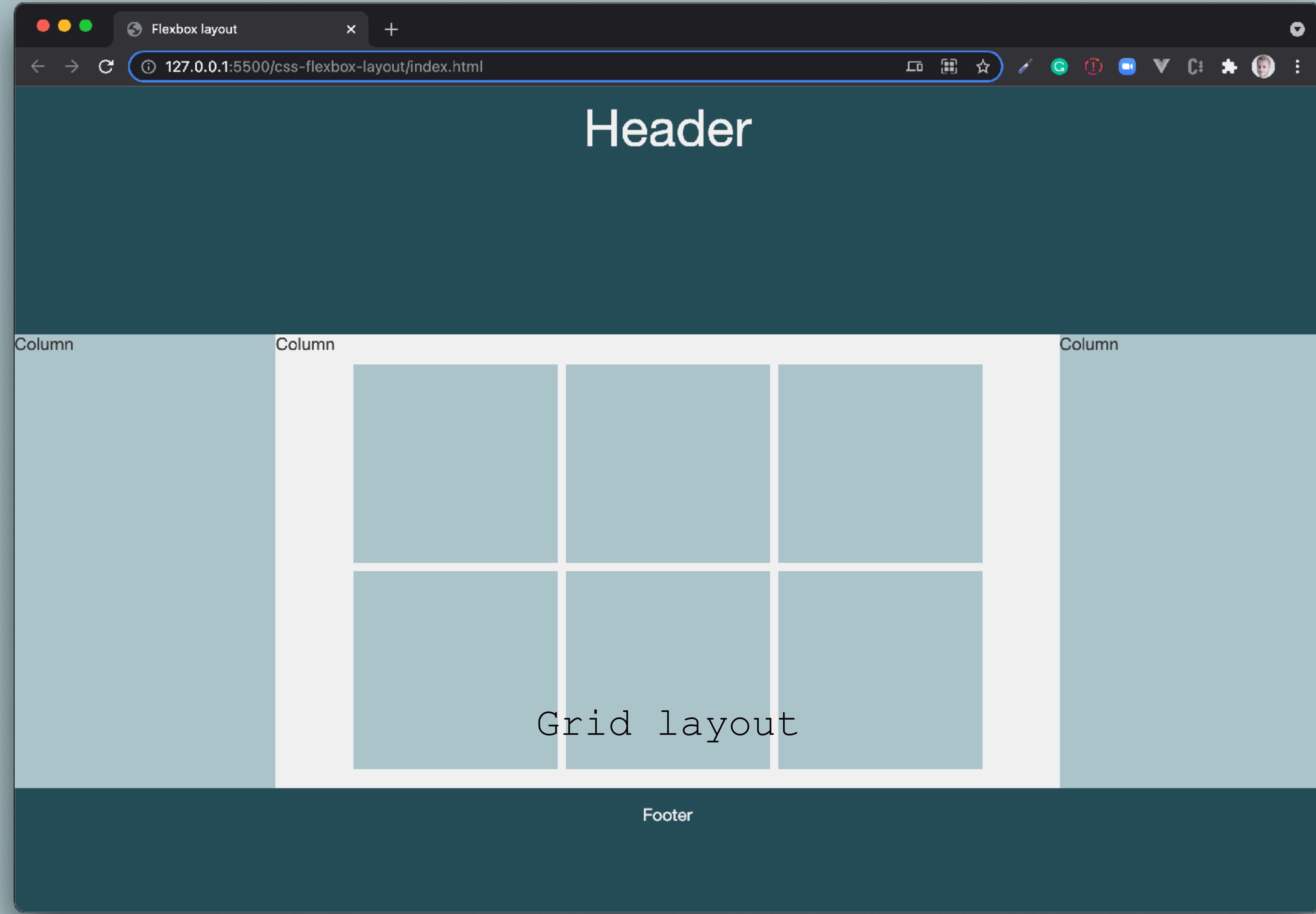
@media (min-width: 992px) {
  .grid-container {
    grid-template-columns: 1fr 1fr 1fr;
  }
}
```



css-grid-layout



css-flexbox-layout



Flexbox layout

CSS Templates

w3schools.com/css/css_templates.asp

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP HOW TO W3.CSS JAVA JQUERY C++

CSS Multiple Columns
CSS User Interface
CSS Variables
CSS Box Sizing
CSS Media Queries
CSS MQ Examples
CSS Flexbox

CSS Responsive
RWD Intro
RWD Viewport
RWD Grid View
RWD Media Queries
RWD Images
RWD Videos
RWD Frameworks
RWD Templates

CSS Grid
Grid Intro
Grid Container
Grid Item

CSS SASS
SASS Tutorial

CSS Examples
CSS Templates
CSS Examples
CSS Quiz
CSS Exercises
CSS Certificate

CSS References
CSS Reference
CSS Selectors

CSS Layout Templates

We have created some responsive starter templates with CSS.

You are free to modify, save, share, and use them in all your projects.

Header, equal columns and footer:

Try it (using float) »

Try it (using flexbox) »

Try it (using grid) »

Header, unequal columns and footer:

Try it (using float) »

Try it (using flexbox) »

Try it (using grid) »

Topnav, content and footer:

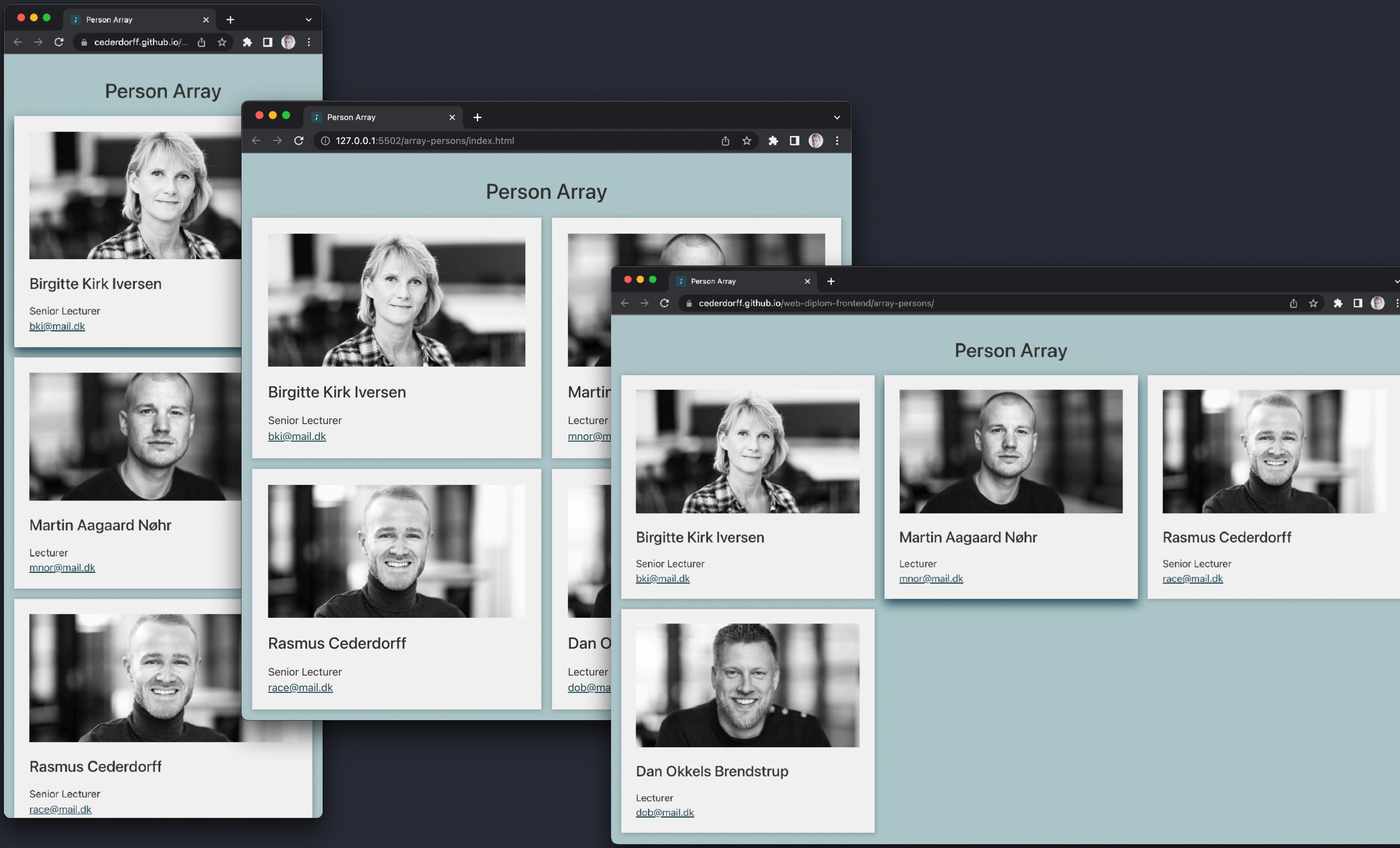
Sidenav and content:

NetSikker inkluderet
sikrer dig hjælp ved digitale krænkelser
Se vilkår på telenor.dk

NEW
We just launched W3Schools videos
Explore now

COLOR PICKER

LIKE US



Person Array with CSS Grid

<https://race.notion.site/Person-Array-with-CSS-Grid-667a504e43d14c17a8a6f2d1c4a4975c>

JavaScript

The image is a collage of various JavaScript-related content. In the center, the word "JavaScript" is written in large, bold, white letters. Surrounding it are several smaller, semi-transparent snippets of code and text boxes containing programming concepts:

- Variables (let & const), Arrays & Objects**
- Data types**
- Template String/ Backtick String**
- Loops**
- Operators and control structures**
- JavaScript load in HTML & DOM Manipulation**
- Array Methods**
- Functions**
- Event Listeners**

Some snippets include:

- const afterSomeTime = (time) => new Promise(resolve => { setTimeout(() => { resolve(true); }, time); });
- const callAfterSomeTime = (callback, time) => afterSomeTime(time).then(callback);
- callAfterSomeTime().then(() => console.log('Hello after 1500ms'), 1500);
- const getData = async (url) => fetch(url);
- document.querySelector('#submit').addEventListener('click', function() { const name = document.querySelector('#name').value;

Webudvikling

