

Webudvikling Frontend

CSS Flexbox & Intro to JavaScript

Webudvikling

kea
KØBENHAVNS ERHVERVSAKADEMI

Agenda

- Demo
- Responsive Web Design
- CSS Layouts & Flexbox
- JavaScript Intro

Demo

The screenshot shows a web browser window with the title "HTML Semantic - Cookie Recipe". The URL is "127.0.0.1:5502/html-semantics-recipe-flexbox/index.html". The page content is a cookie recipe for "Amerikanske chocolate chip cookies". It includes a header, a main flex container with two columns: "Ingredienser" and "Sådan gør du", and a footer. A sidebar on the left lists ingredients. The "Sådan gør du" column contains steps and a photo of cookies. The browser's developer tools are open, showing the HTML structure and the CSS rules for the flex container.

```
<!DOCTYPE html>
<html lang="da">
  <head>...</head>
  <body>
    <header>...</header>
    ... <main class="flex-container">...</main> (flex) = $0
      <!-- app.js file -->
      <script src="app.js"></script>
      <!-- Code injected by live-server -->
      <script>...</script>
    </body>
</html>
```

```
html body main.flex-container
Styles Computed Layout Event Listeners »
Filter :hover .cls + □ □
element.style {
}
.flex-container {
  display: flex; □□
  flex-wrap: wrap;
}
main {
  display: block;
}
main {
  display: block;
}
Inherited from body
```

- Flexbox Layout
- Flexbox Responsive Layout
- Google Font
- Best Practice
- Normalize.css

CSS Best Practices

Organizing your CSS

Top 10 Web Development Best Practices

14 CSS Best Practices for Beginners

CSS Best Practices

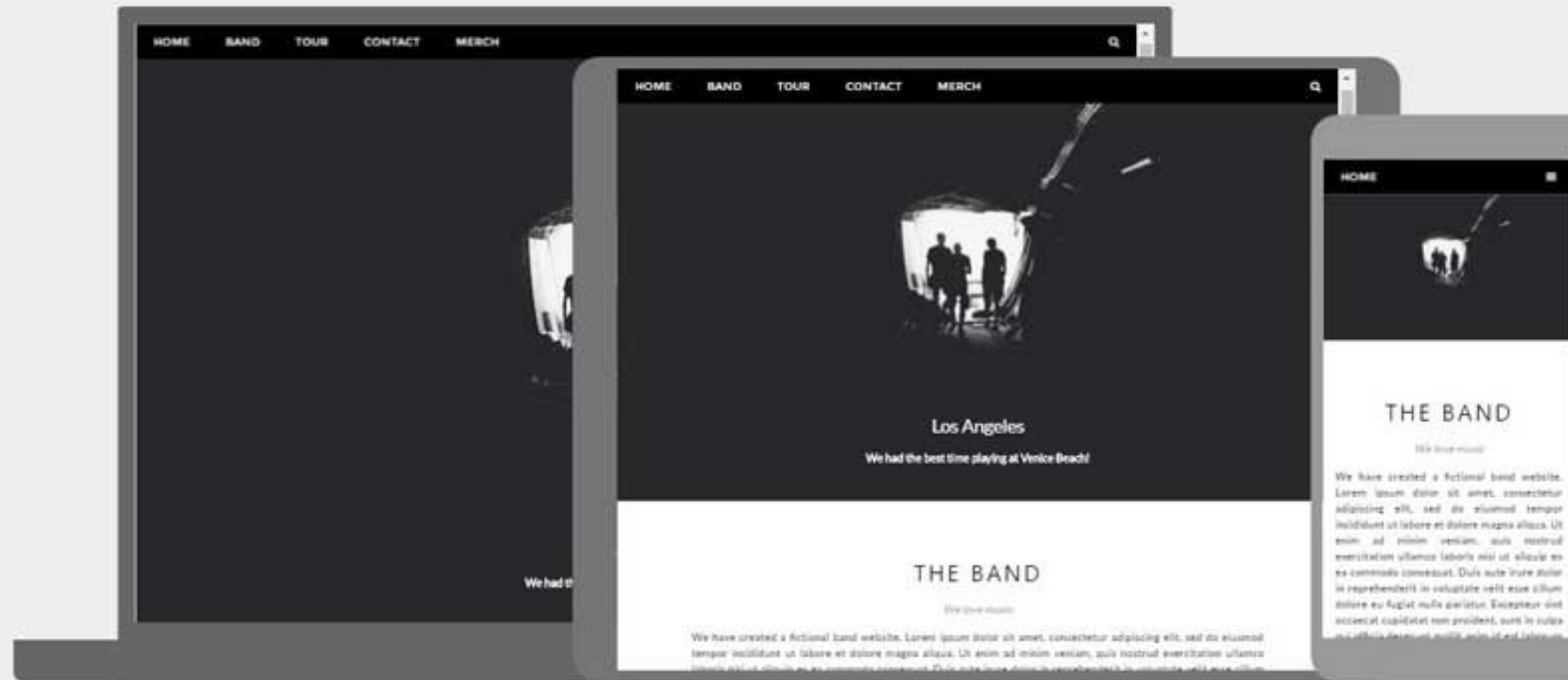
1. Strive for consistency
2. Organise, structure & comment
3. Keep it simple stupid (selectors: tag name, ids and classes)
4. Reset CSS and/or Normalise CSS
5. External Stylesheet over inline and internal

Responsive Web Design



- Create Web Design that looks good across all devices!
- Automatically adjust for different screen sizes and viewports.
- Resize, hide, shrink, or enlarge.

Resize, hide, shrink, enlarge



https://www.w3schools.com/w3css/tryw3css_templates_band.htm
[All W3School Templates](#)

Responsive Images



HTML Responsive Web Design

Not Secure | w3schools.com/html/html_r...

HTML CSS JAVASCRIPT

Responsive Images

Responsive images are images that scale nicely to fit any browser size.

Using the width Property

If the CSS `width` property is set to 100%, the image will be responsive and scale up and down:

```

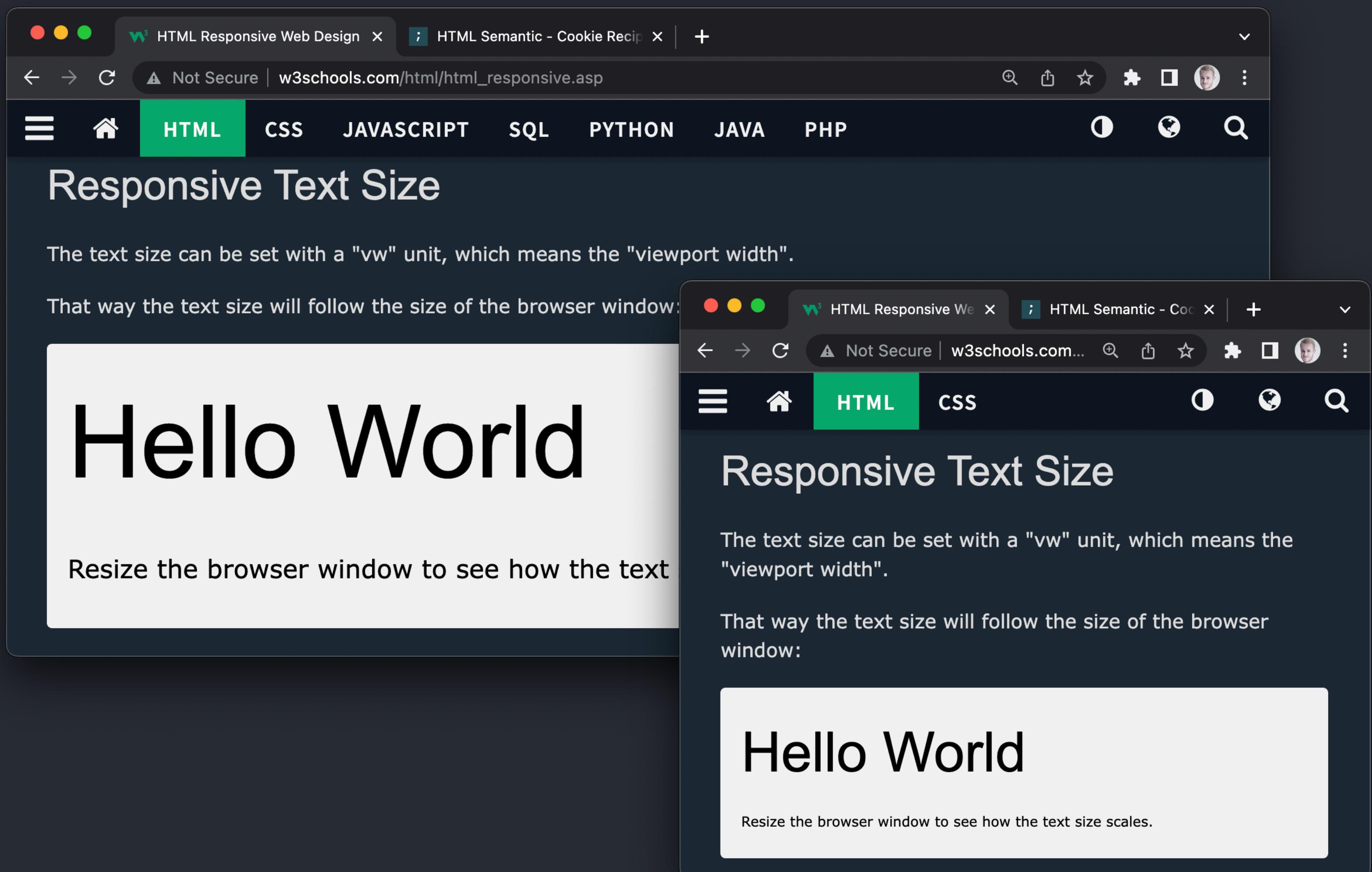
```

```
img {  
    max-width: 100%;  
    height: auto;  
}
```

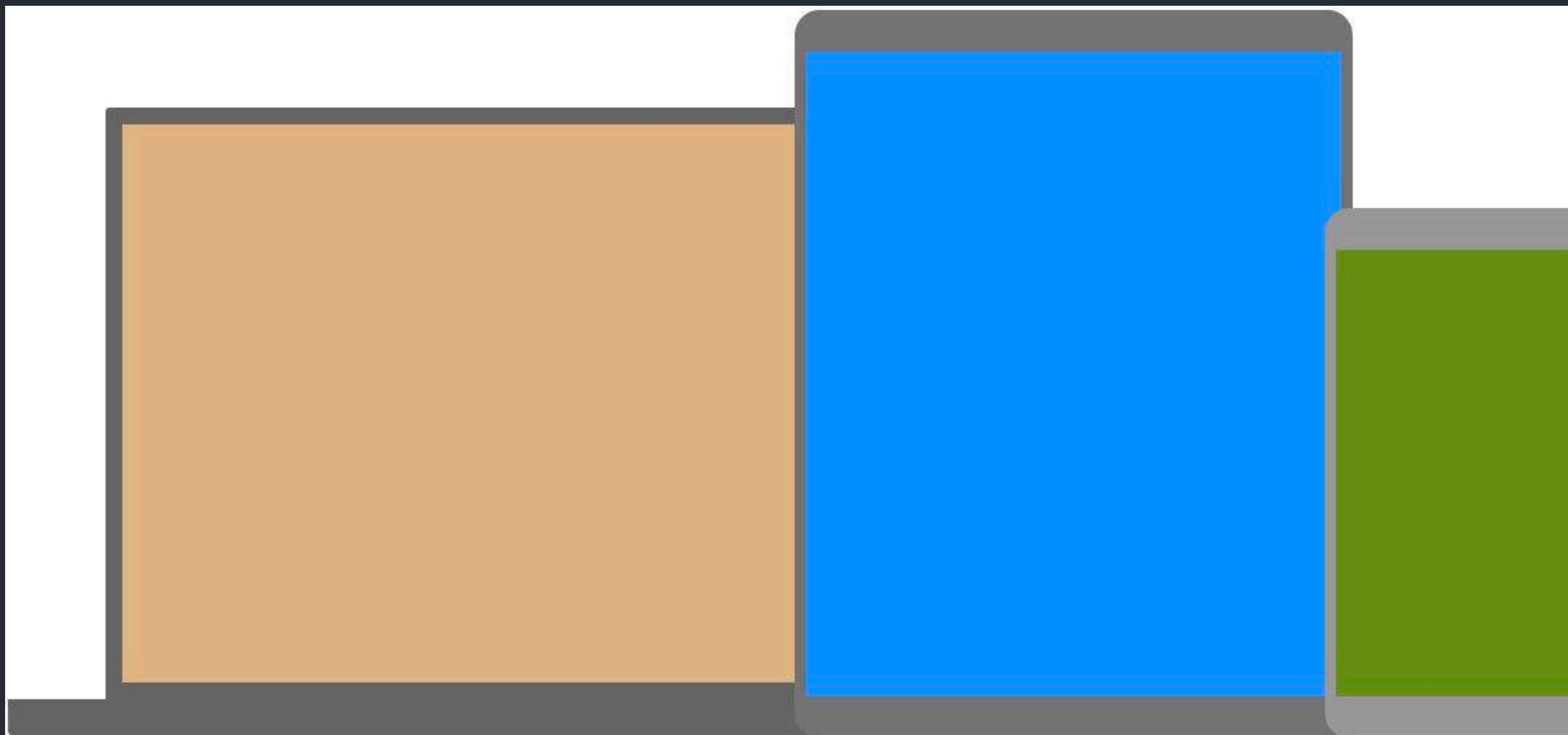
Responsive Text Size

```
<h1>Cookies</h1>
```

```
h1 {  
    font-size: 10vw;  
}
```



Media Queries



```
/* Set the background color of body to tan */
body {
    background-color: tan;
}

/* On screens that are 992px or less, set the background color to blue */
@media screen and (max-width: 992px) {
    body {
        background-color: blue;
    }
}

/* On screens that are 600px or less, set the background color to olive */
@media screen and (max-width: 600px) {
    body {
        background-color: olive;
    }
}
```

Media Queries



The screenshot shows a browser window with the title "Responsive Web Design Media". The address bar contains "w3schools.com/css/css_rwd_mediaqueries.asp". The navigation bar includes links for HTML, CSS (which is highlighted), JAVASCRIPT, SQL, and PYTHON. Below the navigation, there is a snippet of CSS code:

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

A green button labeled "Try it Yourself »" is visible. The main content area has a heading "Example" and a paragraph: "If the browser window is 600px or smaller, the background color will be lightblue:". A code editor-like view shows the same CSS snippet. Below this, there is a section titled "Add a Breakpoint" with a note: "Earlier in this tutorial we made a web page with rows and columns, and it was responsive, but it did not look good on a small screen." It also states: "Media queries can help with that. We can add a breakpoint where certain parts of the design will behave differently on each side of the breakpoint." At the bottom, there are two icons: a large one labeled "Desktop" and a smaller one labeled "Phone".

```
.header h1 {  
  font-size: 1.3em;  
  margin-top: 0;  
  margin: 10px 0 0;  
}
```

```
@media (min-width: 768px) {  
  .header h1 {  
    font-size: 1.5em;  
    margin: 0;  
  }  
}
```

```
@media (min-width: 992px) {  
  .header h1 {  
    font-size: 1.8em;  
  }  
}
```

Media Queries & Menus

The screenshot shows a browser window with two tabs: "Responsive Web Design Media" and "CSS3 Media Queries - Example". The main content area displays a "Media Queries For Menus" section. It includes a description of how media queries are used to create a responsive navigation menu that changes design based on screen size. Below this, there are two examples: "Large screens:" showing a horizontal navigation bar with four links ("Home", "Link 1", "Link 2", "Link 3"), and "Small screens:" showing a vertical dropdown menu where the "Home" link is at the top and "Link 1", "Link 2", and "Link 3" are stacked below it. A large callout box labeled "Example" contains the CSS code for the menu.

```
/* The navbar container */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Navbar links */
.topnav a {
  float: left;
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* On screens that are 600px wide or less, make the menu links stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .topnav a {
    float: none;
    width: 100%;
  }
}
```

```
/* The navbar container */
.topnav {
  overflow: hidden;
  background-color: #333;
}

/* Navbar links */
.topnav a {
  float: left;
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* On screens that are 600px wide or less, make the menu links stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .topnav a {
    float: none;
    width: 100%;
  }
}
```

Media Queries For Columns

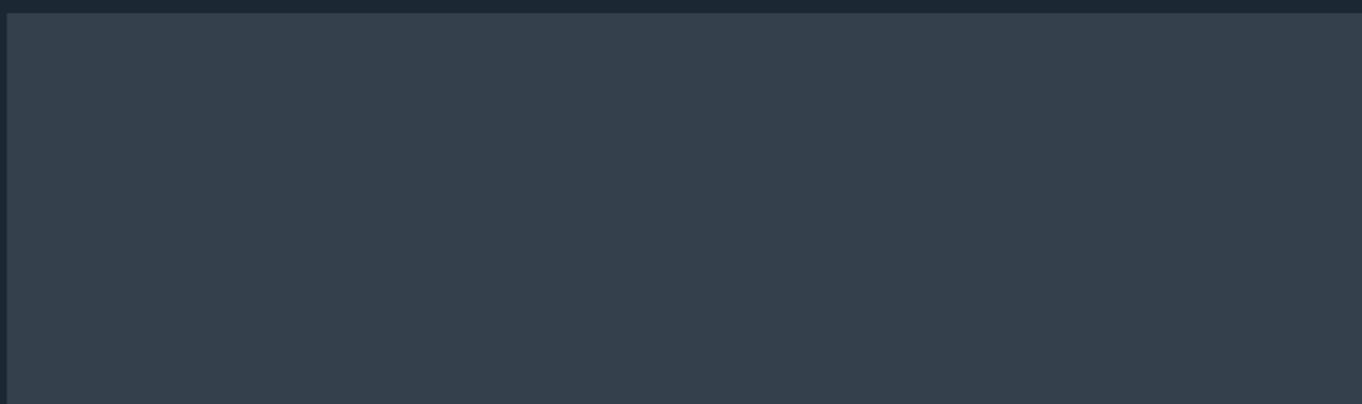
Large screens:



Medium screens:



Small screens:



```
/* Create four equal columns that floats next to each other */
.column {
    float: left;
    width: 25%;
}

/* On screens that are 992px wide or less, go from four columns to two columns */
@media screen and (max-width: 992px) {
    .column {
        width: 50%;
    }
}

/* On screens that are 600px wide or less, make the columns stack on top of each other instead */
@media screen and (max-width: 600px) {
    .column {
        width: 100%;
    }
}
```

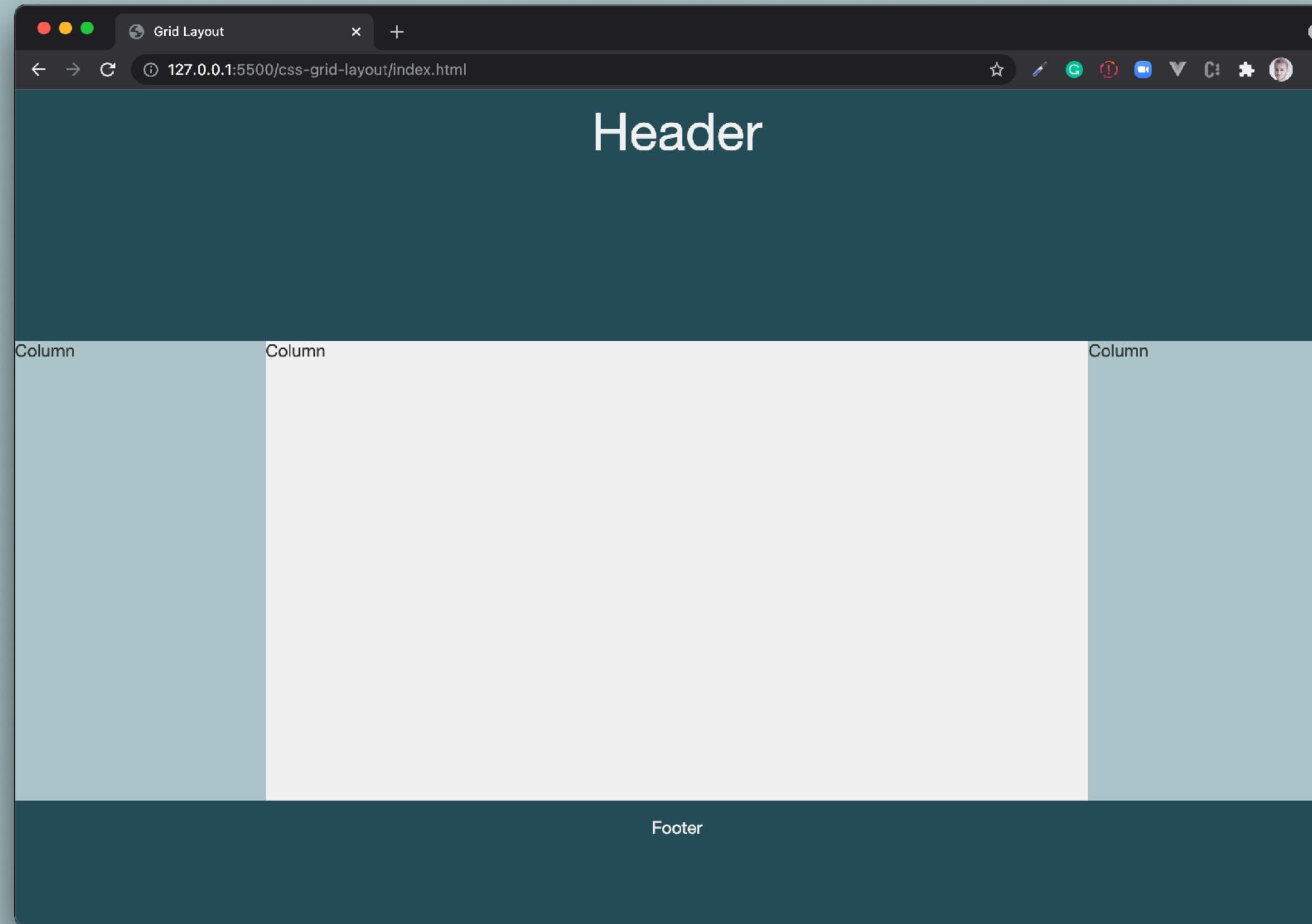
Design for Mobile First

Start with mobile, then add CSS for tablet,
desktop and other screen sizes.

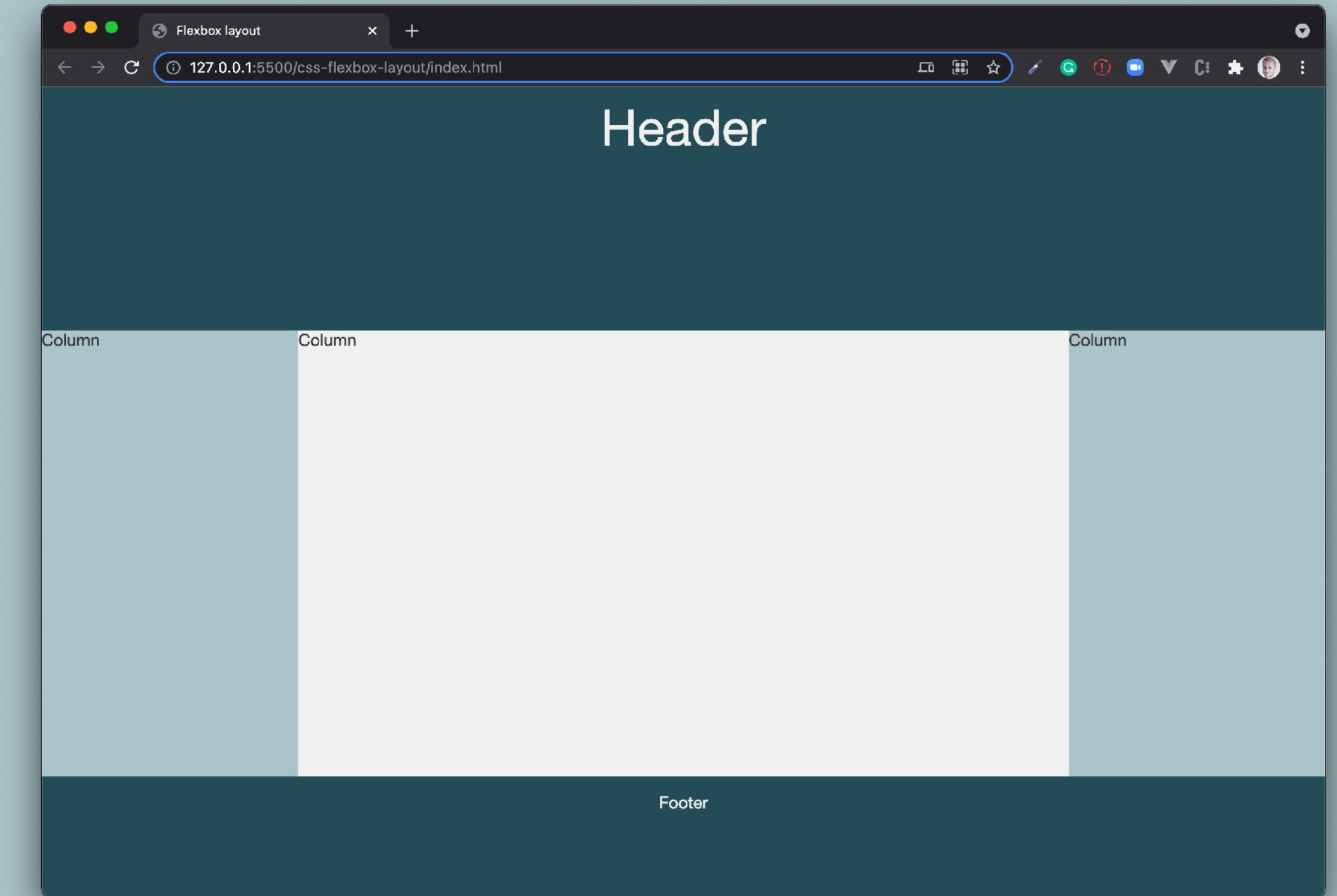
min-width over max-width

https://www.w3schools.com/css/css_rwd_mediaqueries.asp

Grids



css-grid-layout



css-flexbox-layout

Responsive Web Design Templates

https://www.w3schools.com/css/css_rwd_templates.asp

Media Queries

- Implementer media queries for min-width: 600px, min-width: 992px og min-width: 1200px
- Tilpas din h1 og andre headings, så de bliver større, jo større skærmen (browserbredden) er.
- Tilpas positionering (float) af elementer alt efter skærmstørrelsen.
- Tænk Mobile First.

```
.header h1 {  
    font-size: 1.3em;  
    margin-top: 0;  
    margin: 10px 0 0;  
}  
  
@media (min-width: 768px) {  
    .header h1 {  
        font-size: 1.5em;  
        margin: 0;  
    }  
}  
  
@media (min-width: 992px) {  
    .header h1 {  
        font-size: 1.8em;  
    }  
}
```

CSS Layouts

Introduction to CSS layout

Normal flow

The display property

Flexbox

Grid

Floats

Positioning

Table layout

Multiple-column layout

Grid & Flexbox

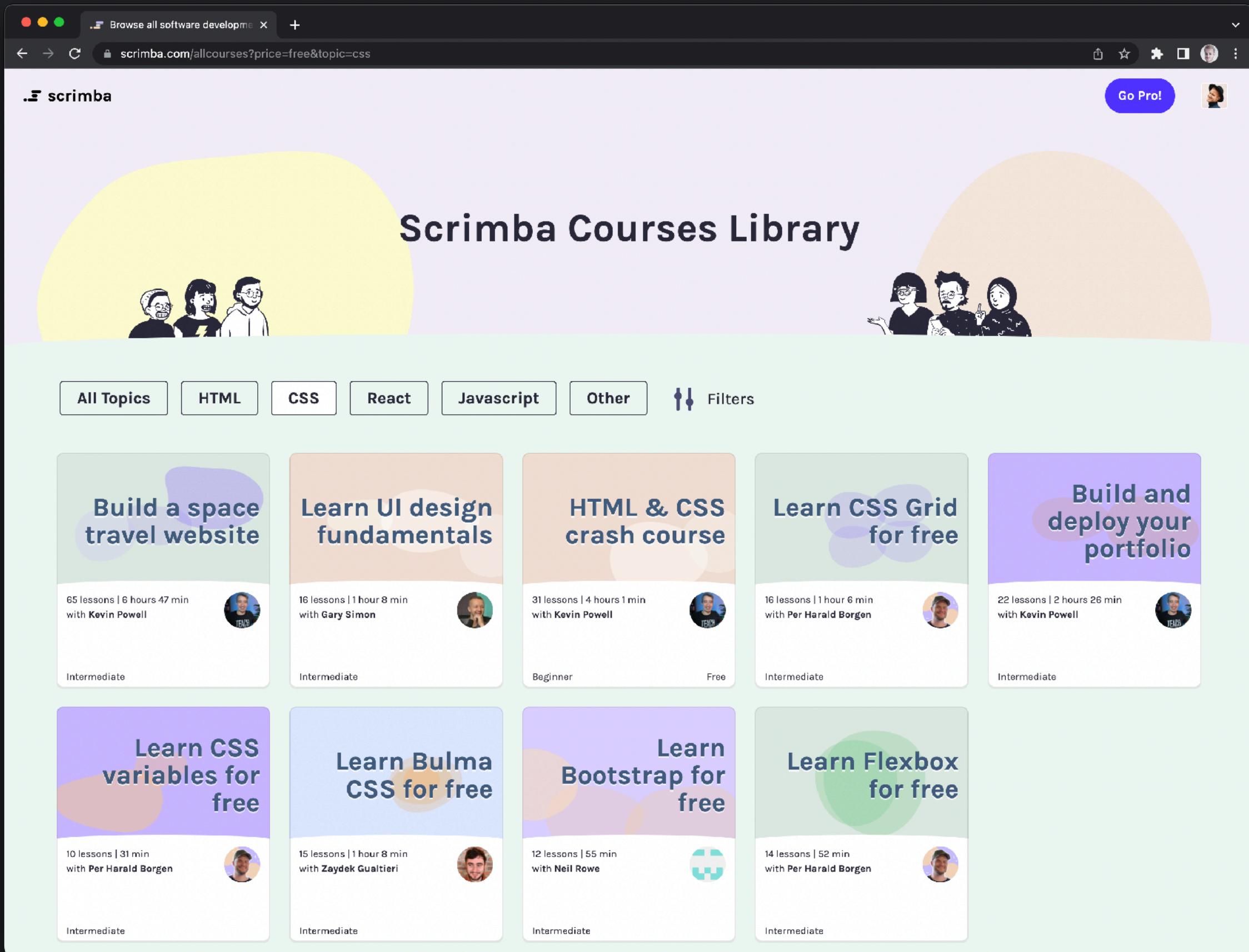
... two CSS layout models that can be used to create layouts with just a couple of CSS rules.

... responsive and flexible layouts.

Scrimba Courses

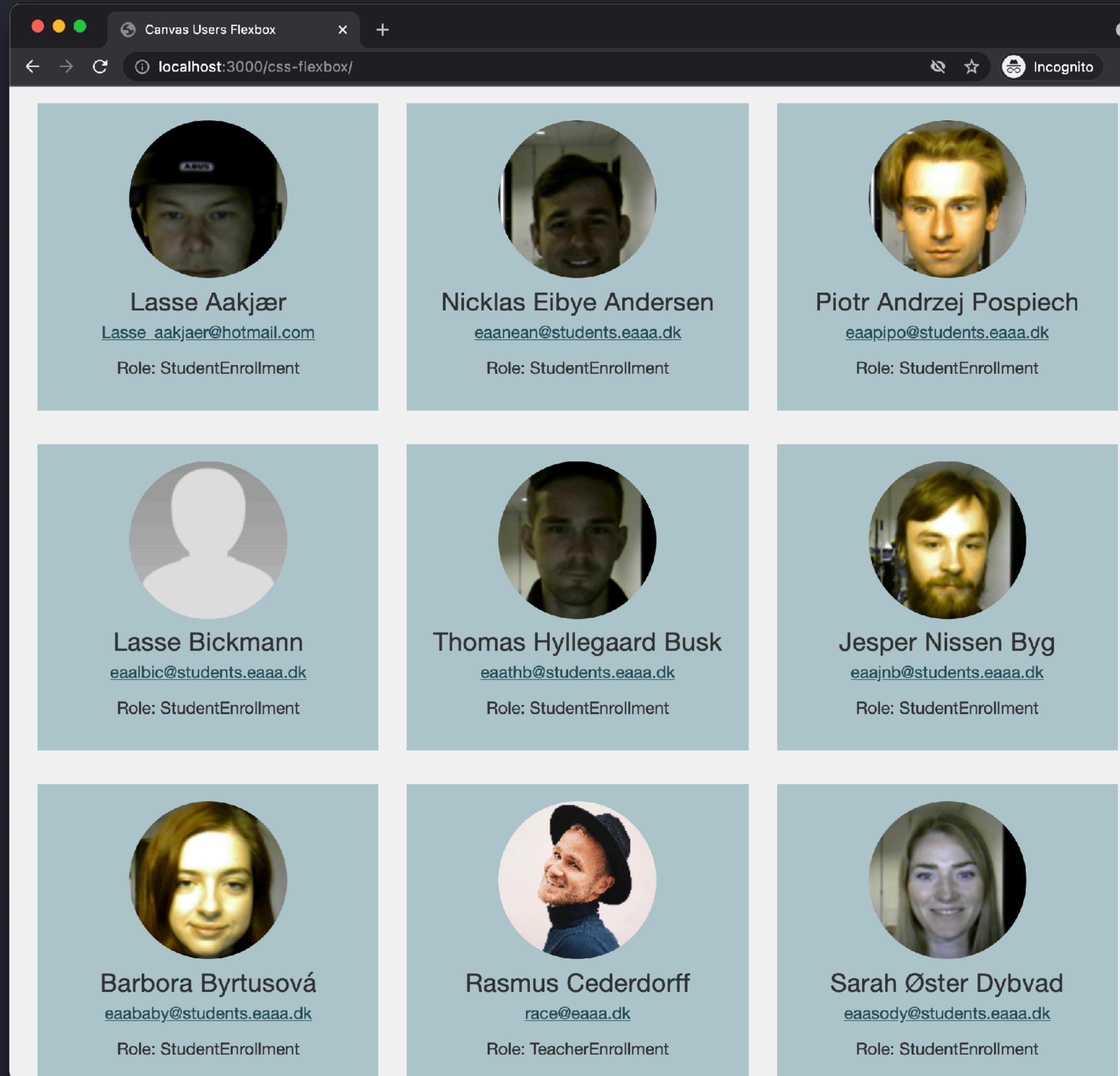
Learn CSS Grid for free: <https://scrimba.com/learn/cssgrid>

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



<https://scrimba.com/allcourses>

CSS Flexbox



```
<section id="users" class="flexbox">...</section>

/* ----- flexbox styling ----- */

.flexbox {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-evenly;
}

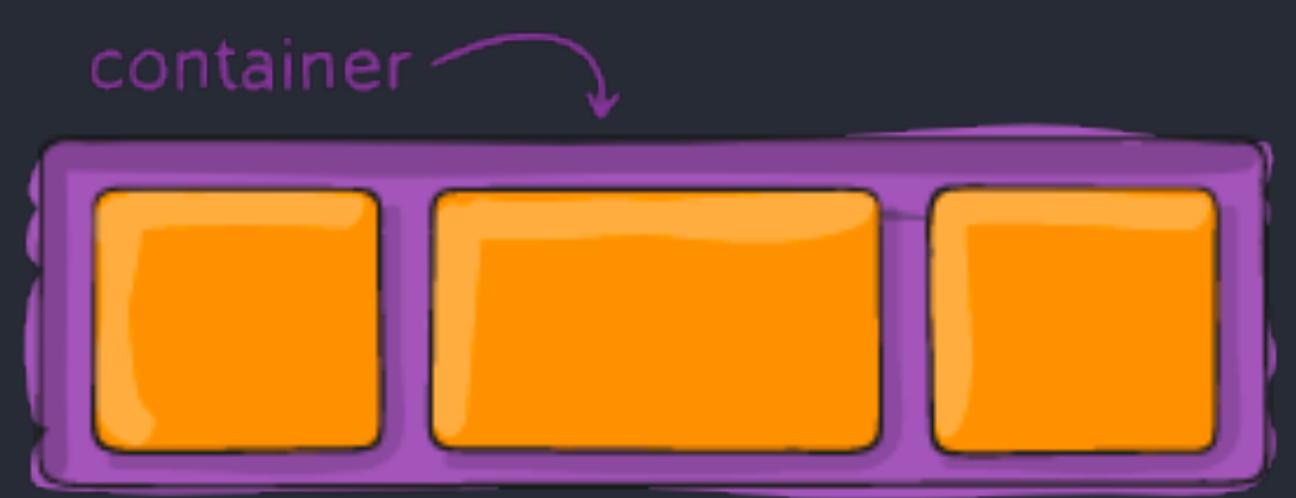
.flexbox article {
  margin: 1em;
  padding: 1em;
  text-align: center;
  background-color: var(--light-green);
  transition: 0.3s;
  flex-basis: 100%; /*default size of an element before the remaining space is distributed*/
  flex-grow: 1; /*ability for a flex item to grow. 1 = all items equally large*/
}

@media (min-width: 600px) {
  .flexbox article {
    flex-basis: 40%;
  }
}

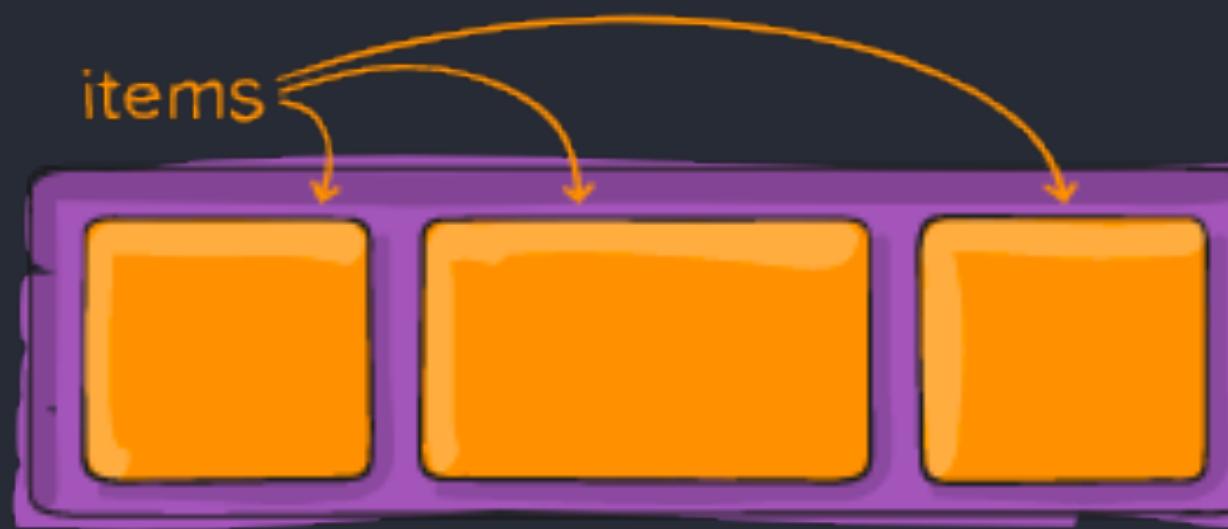
@media (min-width: 992px) {
  .flexbox article {
    flex-basis: 27%;
  }
}
```

CSS Flexbox

... makes it easier to design flexible responsive layout
structure without using float or positioning.



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



```
.item {  
  flex: 50%;  
}
```

FLEX-DIRECTION

CSS Demo: flex-direction

RESET

`flex-direction: row;`

`flex-direction: row-reverse;`

`flex-direction: column;`

`flex-direction: column-reverse;`

Item One Item Two Item Three

FLEX - WRAP

CSS Demo: flex-wrap

RESET

`flex-wrap: nowrap;`

`flex-wrap: wrap;`

`flex-wrap: wrap-reverse;`

The screenshot shows a dark-themed CSS demo interface. On the left, there are three large rectangular boxes containing CSS code snippets. From top to bottom, they are: `flex-wrap: nowrap;`, `flex-wrap: wrap;`, and `flex-wrap: wrap-reverse;`. To the right of these boxes is a larger area containing six blue-outlined rectangular items labeled "Item One" through "Item Six". The items are arranged horizontally, illustrating the effect of each flex-wrap value. A "RESET" button is located in the top right corner of the demo area.

Item One Item Two Item Three Item Four Item Five Item Six

<https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap>

Flexbox

The screenshot shows a cookie recipe card with the following structure:

- Header:** Cookies
- Image:** Amerikanske chocolate chip cookies (a stack of cookies)
- Description:** Amerikanske chocolate chip cookies - sprøde udenpå og bløde inden... Og hvis der er noget der virkelig kan, derover på den anden side af atlanten, så er det at bage cookies. Om vi kalder dem Chocolate chip cookies eller bare Cookies gør nok ikke den store forskel. Der er chokoladestykker i - og de smager pragtfuldt. Prøv denne opskrift på den perfekte Chocolate chip cookie, næste gang lækkersulten banker på døren!
- Section: Ingredienser**
 - Cookies
 - Blødt smør, 150 g
 - Sukker, (ca. 1½ dl) 125 g
 - Brun farin, (ca. 1½ dl) 125 g
 - Vanillesukker, 2 tsk
 - Æg, 1
 - Hvedemel (ca. 4½ dl), 250 g
 - Natron, 1 tsk
 - Groft salt, ¾ tsk
 - Bagepulver, ½ tsk
 - Grofthakket mørk chokolade (ca. 55% kakao) eller chokoladeknapper, 200 g
- Section: Sådan gør du**
 - Cookies
 - 1. Pisk smør, sukker, brun farin, vanilje og æg godt sammen.
 - 2. Bland hvedemel, natron, salt og bagepulver og rør det i smørblandingen sammen med chokolade.
 - 3. Del dejen i 20 stykker a ca. 45 g og form dem til kugler.
 - 4. Sæt kuglerne på plader med bagepapir - tryk dem let flade. Kagerne flyder en del ud, så sørge for, at der er god plads imellem dem.
 - 5. Bag kagerne til de er gyldne, men bløde i midten.
 - 6. Lad kagerne afkøle ca. 5 min. på bagepladen og flyt dem så over på en rist.
- Text:** Bagetid
10-12 min. ved 175° - varmluft.

```
/* ----- flexbox styling ----- */
.flex-container {
  display: flex;
  flex-wrap: wrap; /*The flex items break into multiple lines.*/
}

.description {
  flex: 60%;
  align-self: center;
}

figure {
  flex: 40%;
}

.steps {
  flex: 50%;
}

.ingredients {
  flex: 50%;
```

<https://cederdorff.github.io/web-diplom-frontend/html-semantics-recipe-flexbox/>

Flexbox Responsive

A screenshot of a web browser displaying a cookie recipe card. The card is divided into sections: 'Amerikanske chocolate chip cookies' (title), 'Klassiske amerikanske cookies - sprøde udenpå og bløde indenpå... Og hvis der er noget du virkelig kan, derover på den anden side af atlanten, så er det at bage cookies. Om vi kalder dem Chocolate chip cookies eller bare Cookies gør nok ikke den store forskel. Der er chokoladestykker i - og de smager pragtfuldt. Prøv denne opskrift på den perfekte Chocolate chip cookie, næste gang lækkersulten banker på døren!', 'Ingredienser' (list of ingredients), 'Sådan gør du' (instructions), and 'Bagetid' (baking time). The card is styled with a light blue background and white text. A sidebar on the right shows the 'Arla Cookies Opskrift' logo.

<https://cederdorff.github.io/web-diplom-frontend/html-semantics-recipe-flexbox-responsive/>

```
/* ----- flexbox responsive styling ----- */
@media (min-width: 700px) {
  .flex-container {
    display: flex;
    flex-wrap: wrap; /*The flex items break into multiple lines*/
  }

  .description {
    flex: 60%;
  }

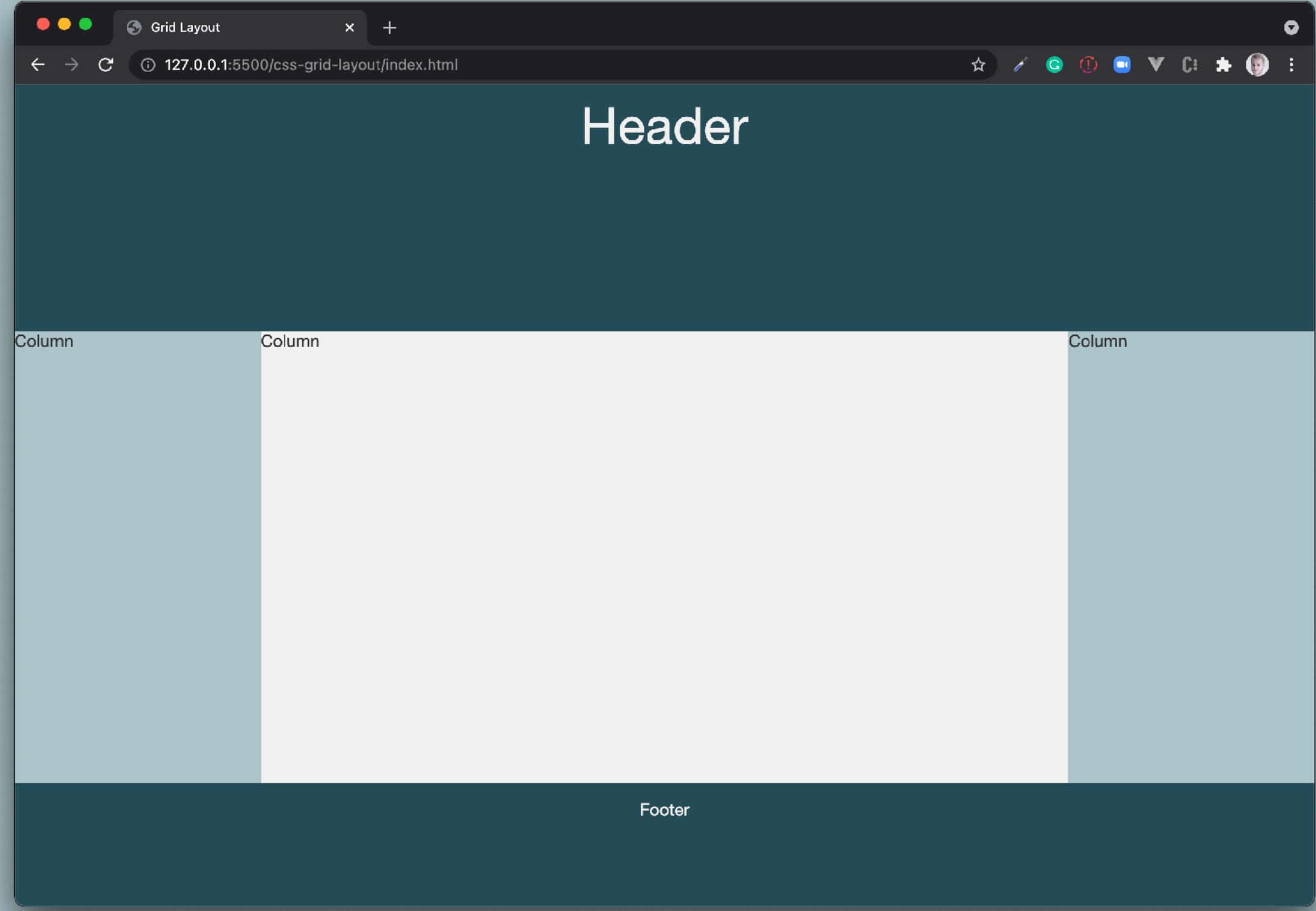
  figure {
    flex: 40%;
  }

  .steps {
    flex: 50%;
  }

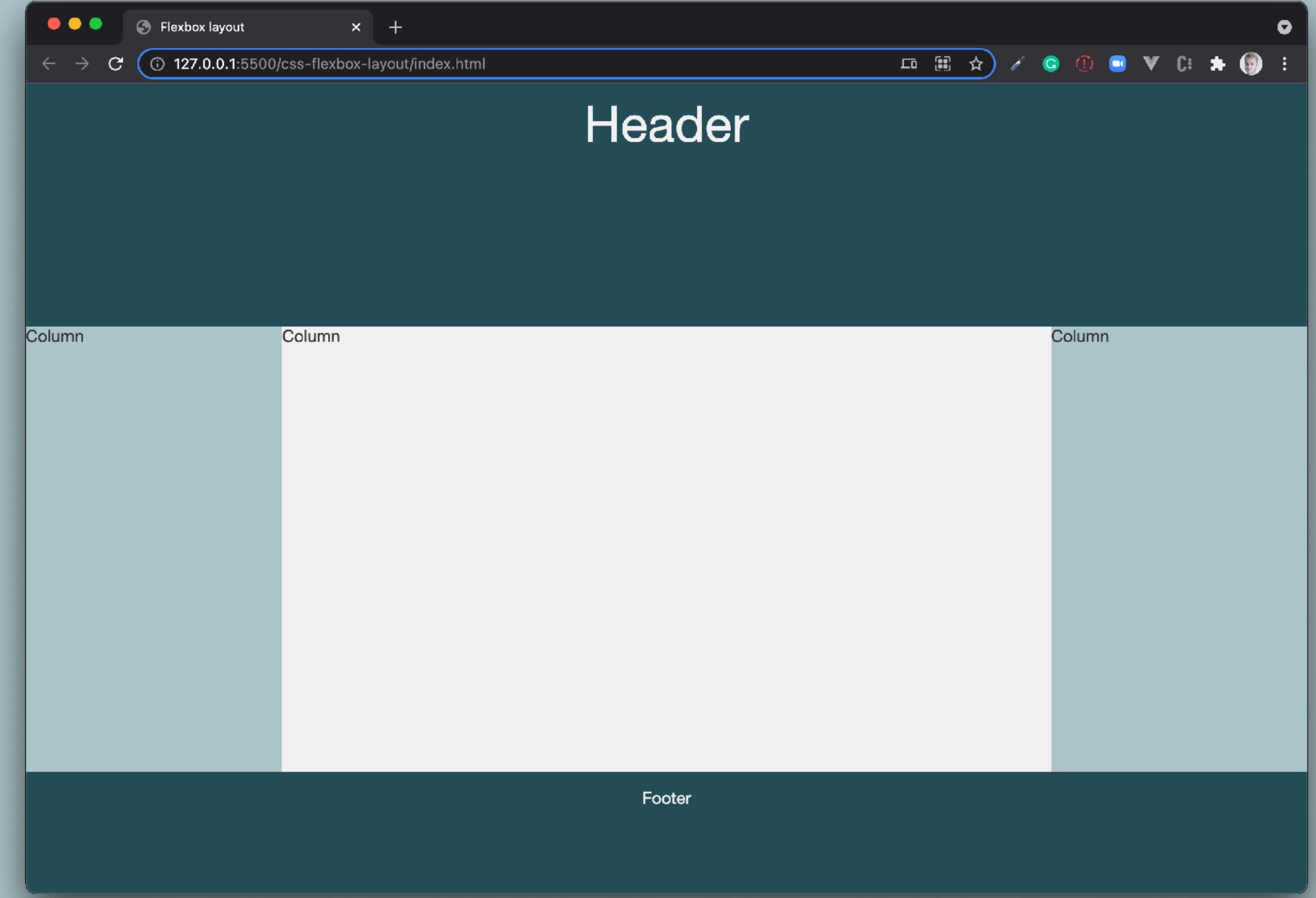
  .ingredients {
    flex: 50%;
  }
}

@media (min-width: 1200px) {
  .description {
    flex: 50%;
    align-self: center;
  }

  figure {
    flex: 50%;
  }
}
```



css-grid-layout



css-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

Flexbox Recipe

- Implementer flexbox for din opskriftsside.
- Placer det indhold du vil “flexe” i en container og tilføj display: flex og flex-wrap: wrap
- Angiv flex for dine elementer (kun elementer på øverste niveau).
- Gør løsningen responsiv så stylingen tilpasser afhængigt af skærmstørrelse. Fx ingen flex på små skærme, men flex fra en given min-width.

```
/* ----- flexbox styling ----- */
.flex-container {
  display: flex;
  flex-wrap: wrap; /*The flex items break into multiple lines if they don't fit on one line*/
}

.description {
  flex: 60%;
  align-self: center;
}

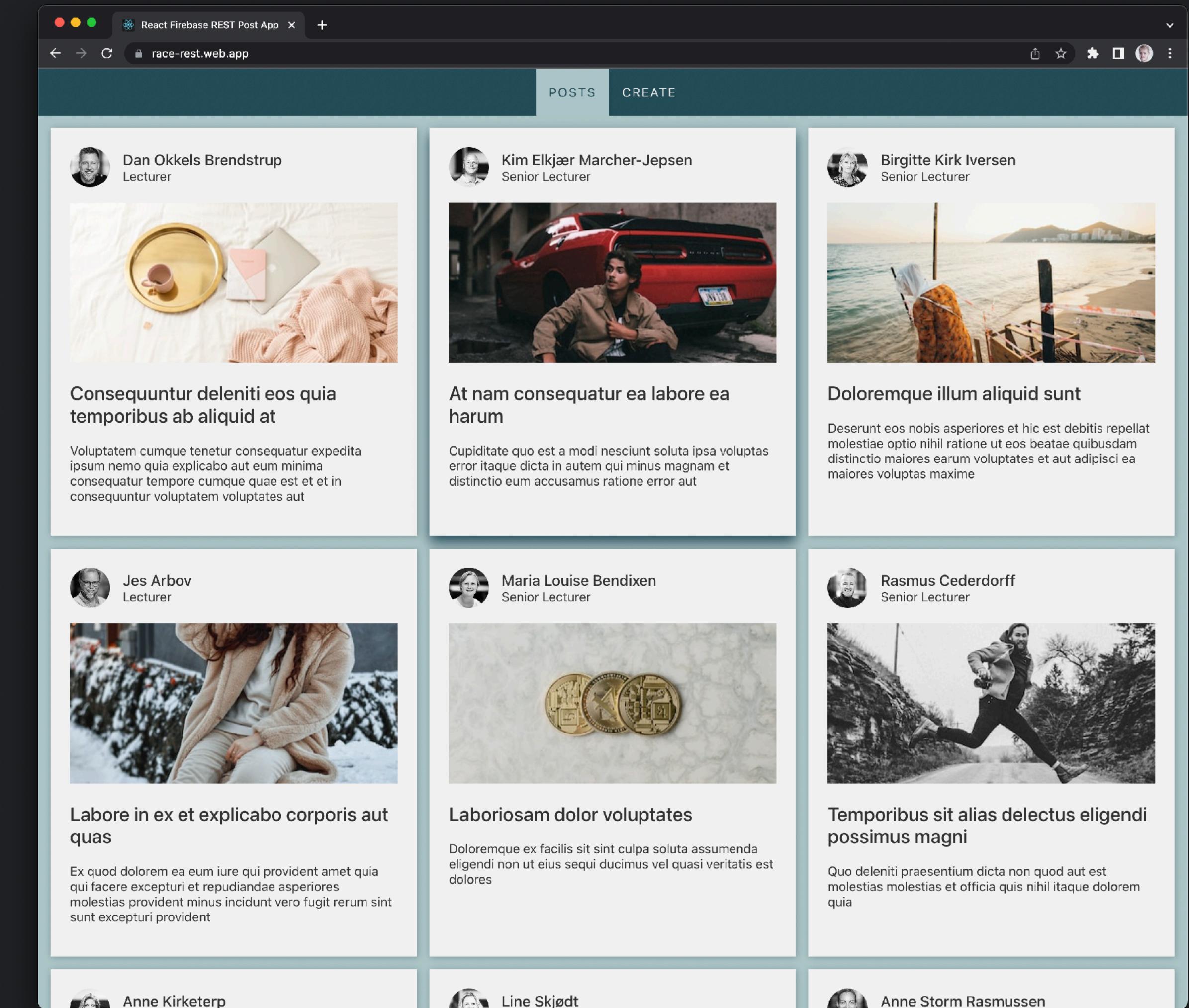
.figure {
  flex: 40%;
}

.steps {
  flex: 50%;
}

.ingredients {
  flex: 50%;
}
```

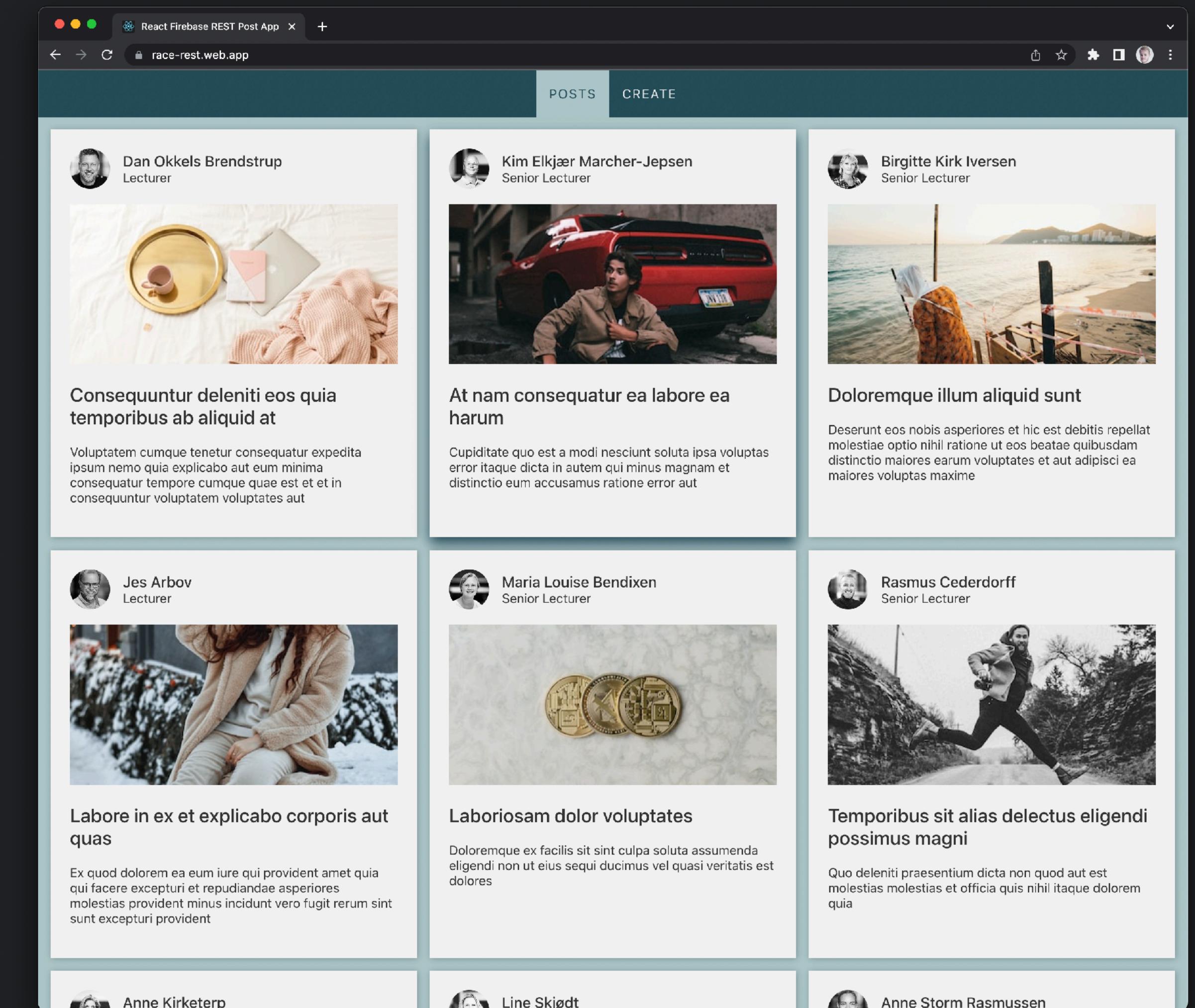
Flexbox Grid

- Brug flexbox til at implementere et grid.
- Du bestemmer selv indhold af dine elementer.
- Sørg for at layoutet er responsivt og følger Mobile First-princippet.



Flexbox Page Layout

- Brug flexbox til at implementere et layout for en hel side.
- Du bestemmer selv indhold af dine elementer.
- Sørg for at layoutet er responsivt og følger Mobile First-princippet.
- Inspiration: [CSS Flexbox](#) (scroll ned) og [CSS Templates](#)



CSS LAYOUT.IO

LAYOUTS AND PATTERNS MADE WITH CSS

CSS TEMPLATES

https://www.w3schools.com/css/css_templates.asp

Frontend stack

HTML



JS



CSS



STRUCTURE
CONTENT

FUNCTIONALITY
BEHAVIOR

LAYOUT/STYLING
PRESENTATION

Introduction to JavaScript

Browser Web Developer Tools:

Console

Variables (let & const)

Template literals

Data types

Arrays & objects

Operators and control structures

JavaScript load in HTML



Code
Every
Day