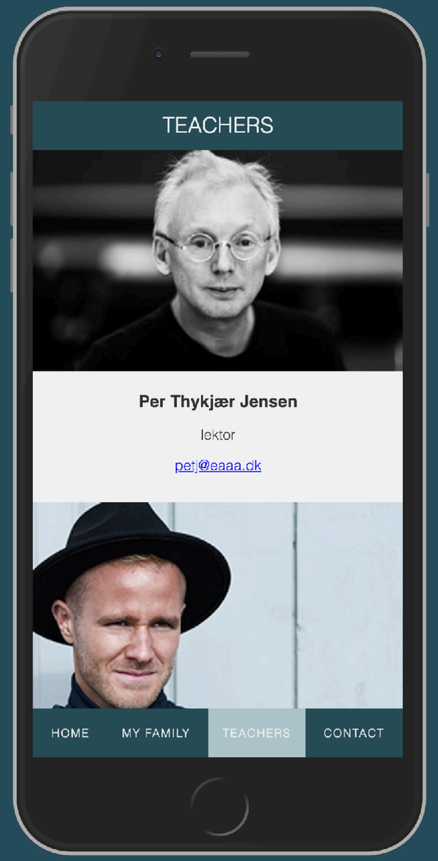
WEB DEVELOPMENT

FRONTEND

(API)

BACKEND



FETCH(...)

REQUEST

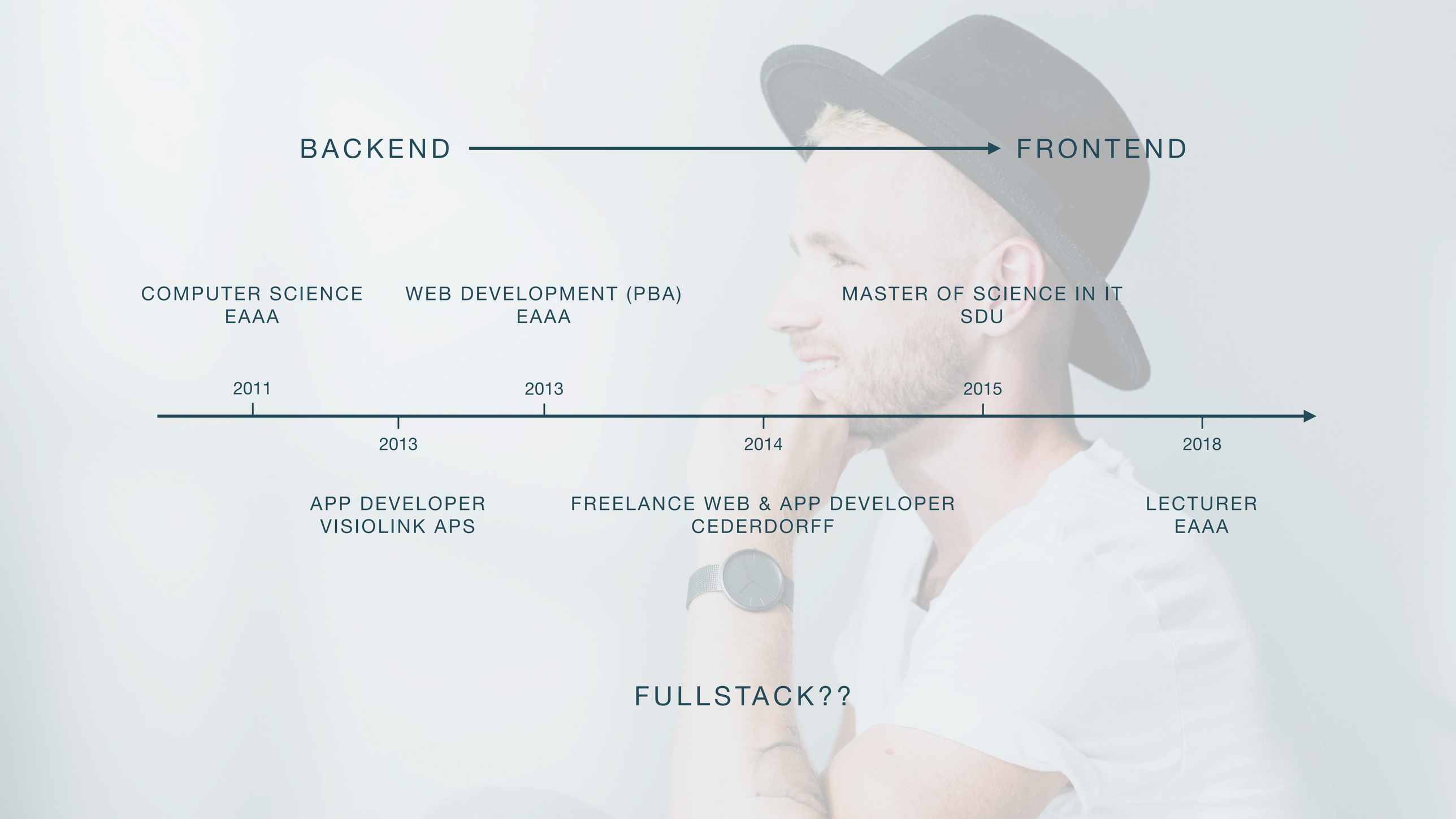
JSON DATA

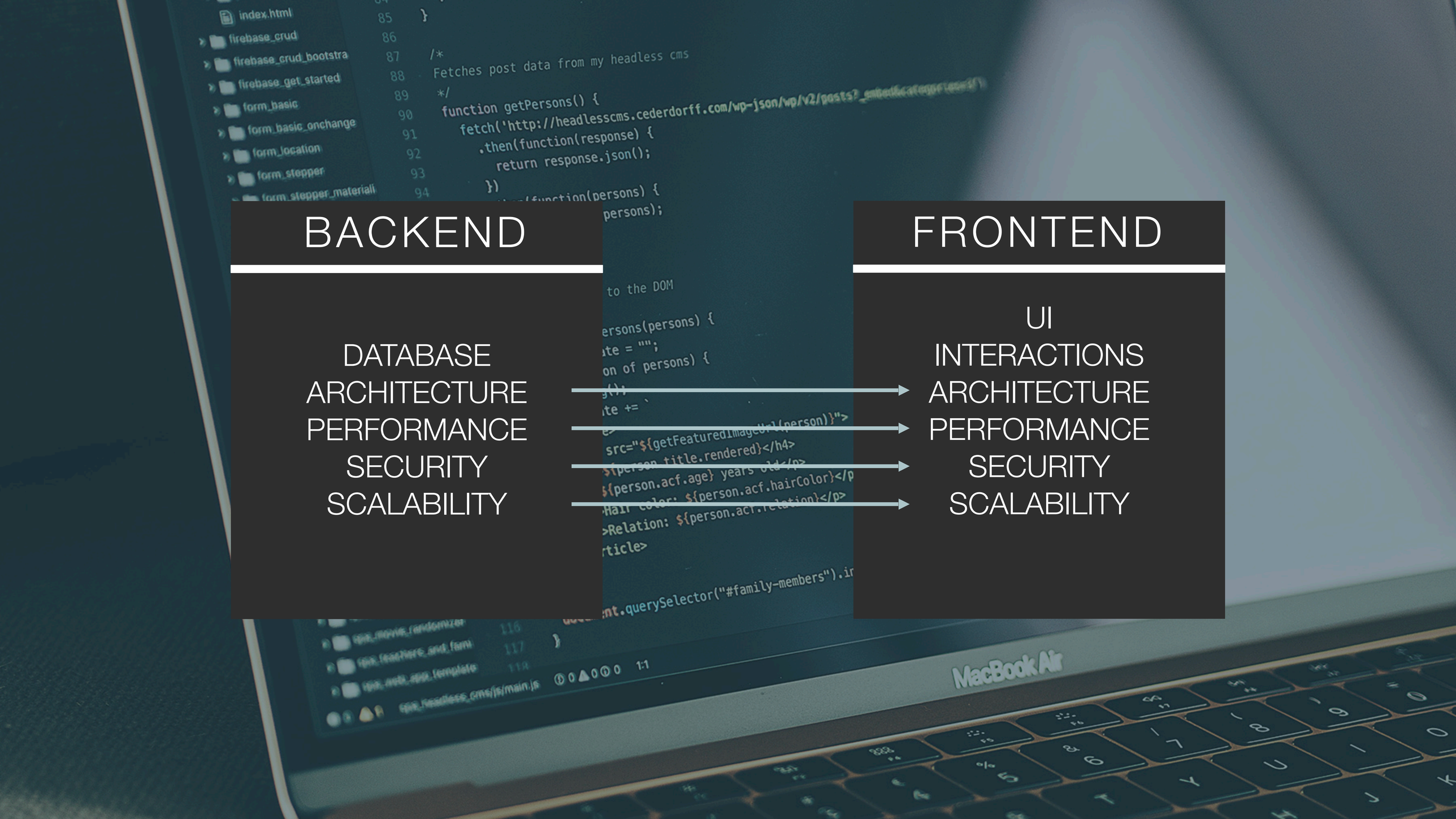
RESPONSE

CLIENT SERVER MODEL

CLIENT

SERVER


BACKEND
FRONTEND
COMPUTER SCIENCE
EAAA
WEB DEVELOPMENT (PBA)
EAAA
MASTER OF SCIENCE IN IT
SDU
2011
2013
2013
2014
2015
2018
APP DEVELOPER
VISIOLINK APS
FREELANCE WEB & APP DEVELOPER
CEDERDORFF
LECTURER
EAAA
FULLSTACK??


BACKEND
DATABASE
ARCHITECTURE
PERFORMANCE
SECURITY
SCALABILITY
FRONTEND
UI
INTERACTIONS
ARCHITECTURE
PERFORMANCE
SECURITY
SCALABILITY

Test and explore: users-match-frontend

Explore the code: users-match-backend

IN PAIRS

1. Test and explore the above projects (fetch and pull from GitHub).
2. Discuss and explain the functionality of the two projects.
3. What’s the main functionality of the projects and how are they communicating?
4. Put the projects in to perspective of what you have learn so far. Consider why it is important to know about both backend and frontend development.

INTRODUCTION TO THE COURSE:

JAVASCRIPT, HTML & CSS

FRONTEND PROGRAMMING

ERHVERVSAKADEMI AARHUS
BUSINESS ACADEMY AARHUS

AGENDA

Introduction to the course

Frontend Dev Stack

CSS layouts

JavaScript

Canvas Users Case

EXERCISES

Test Users Match

Setup Dev Environment

Grid Layout

Page Layout

Grid & Card Layout

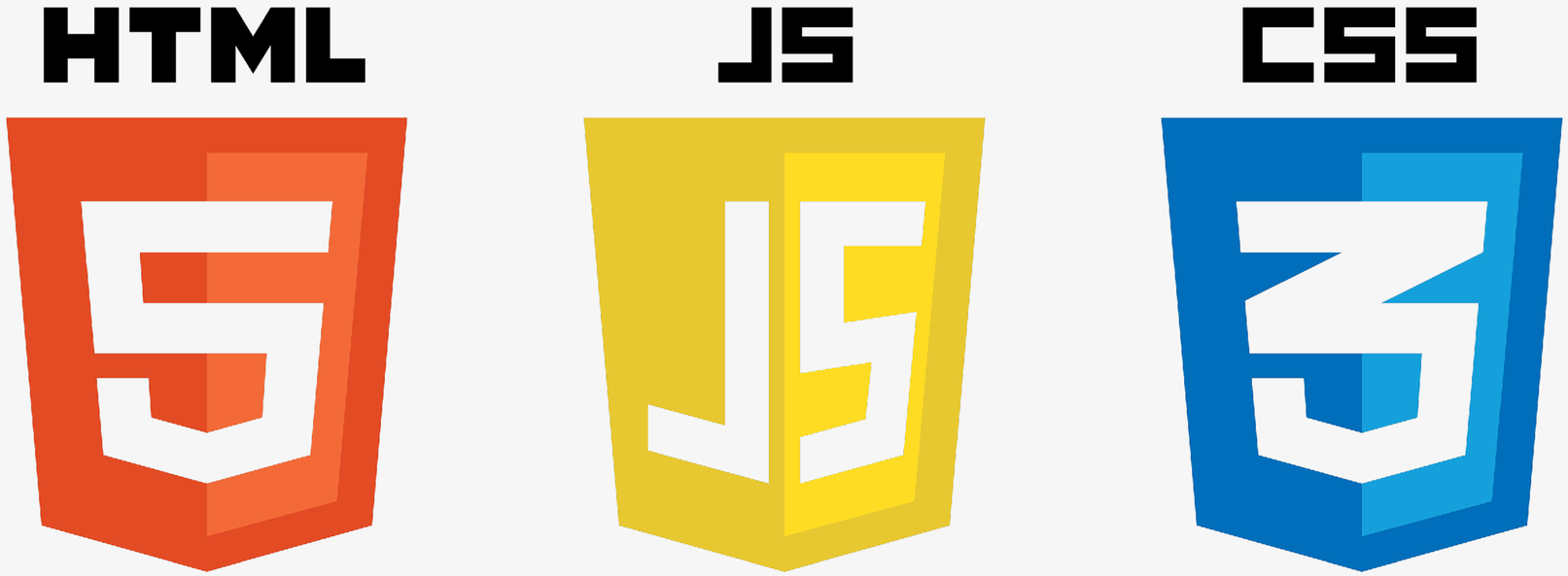
JS HTML DOM #1 & #2

Canvas Users Case #1, #2 & #3

FRONTEND PROGRAMMING (5 ECTS)

"[...] deals with the development of simple, modern web applications. The course element contains HTML, CSS, modern web architecture and HTTP protocol. The subject element includes programming and implementation of [...] simple web applications, JavaScript programming language, debugging tools to ensure the quality of the development process and data from web services."

Curriculum, Part 2: Institutional Part


HTML
JS
CSS

KNOWLEDGE, SKILLS & COMPETENCIES

“practice and applied theory and methods within Frontend development”

“[...] JavaScript programming language and will master the skills associated with employment within Frontend development and the implementation of simple web applications”

“manage complex and development-orientated situations in relation to the construction of web-based applications”

“identify their own learning needs and develop their own knowledge, skills and competencies in relation to Frontend development”

Curriculum, Part 2: Institutional Part

The Frontend Programming Course will improve your knowledge of HTML, CSS, and JavaScript, and prepare you for the Advanced Web Programming course in the 2nd semester.


PROGRAMMING KNOWLEDGE
BACKEND & FRONTEND

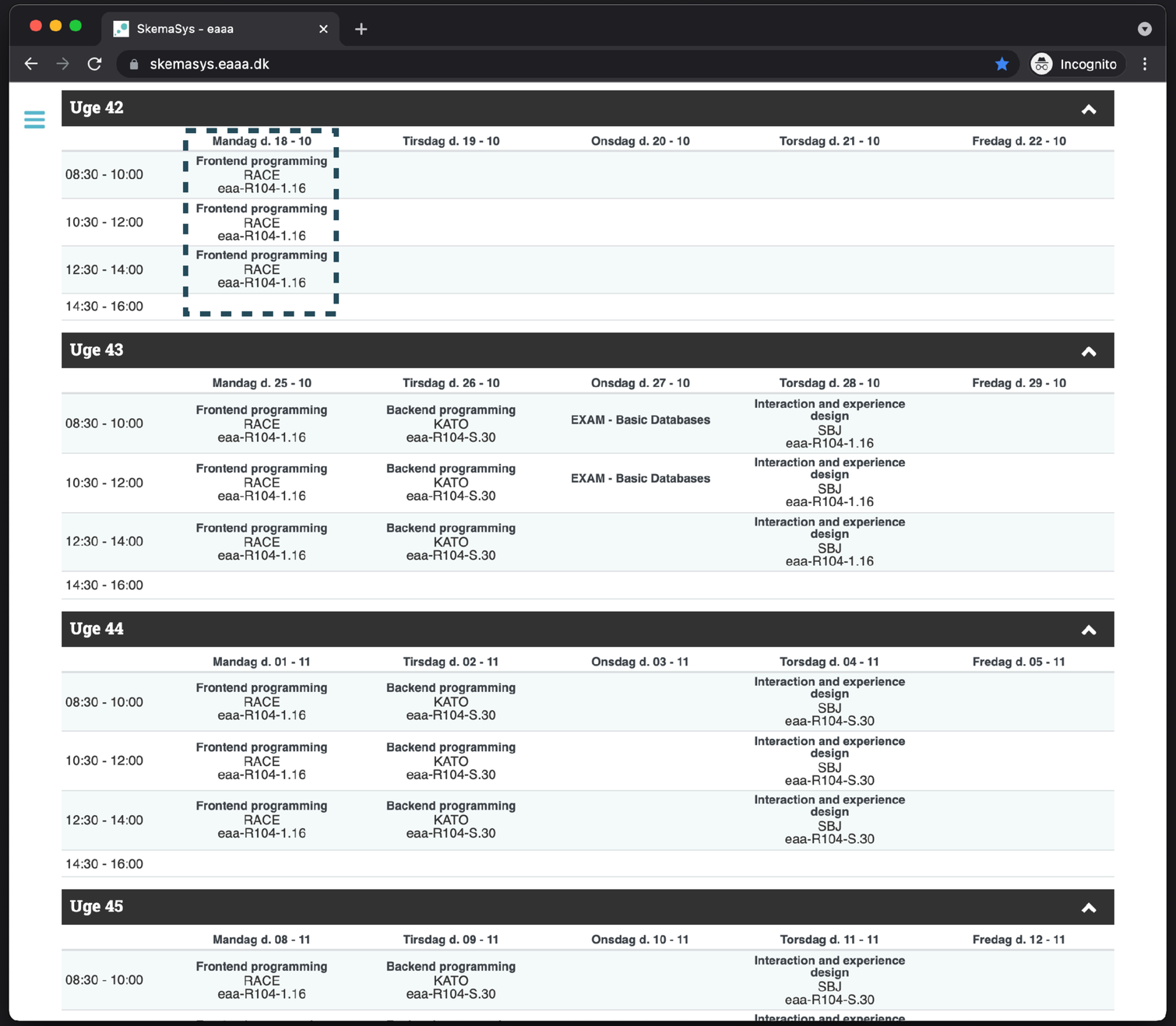
Course Overview

Date	Time	Room	Lesson
18.10.2021	08:30 - 14:00	R104-1.16	1. Introduction to the course: JavaScript, HTML & CSS
25.10.2021	08:30 - 14:00	R104-1.16	2. JavaScript & Web App Development
01.11.2021	08:30 - 14:00	R104-1.16	3. Data & ASYNC JavaScript
08.11.2021	08:30 - 14:00	R104-1.16	4. Object-oriented JS, Modules & Components
15.11.2021	08:30 - 14:00	R104-1.16	5. Advanced JavaScript
22.11.2021	08:30 - 14:00	R104-1.16	6. Recap, Exam prep & Supervision

eaaa.instructure.com/courses/13355

or

Course in Notion



Uge 42

	Mandag d. 18 - 10	Tirsdag d. 19 - 10	Onsdag d. 20 - 10	Torsdag d. 21 - 10	Fredag d. 22 - 10
08:30 - 10:00	Frontend programming RACE eaa-R104-1.16				
10:30 - 12:00	Frontend programming RACE eaa-R104-1.16				
12:30 - 14:00	Frontend programming RACE eaa-R104-1.16				
14:30 - 16:00					

Uge 43

	Mandag d. 25 - 10	Tirsdag d. 26 - 10	Onsdag d. 27 - 10	Torsdag d. 28 - 10	Fredag d. 29 - 10
08:30 - 10:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30	EXAM - Basic Databases	Interaction and experience design SBJ eaa-R104-1.16	
10:30 - 12:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30	EXAM - Basic Databases	Interaction and experience design SBJ eaa-R104-1.16	
12:30 - 14:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-1.16	
14:30 - 16:00					

Uge 44

	Mandag d. 01 - 11	Tirsdag d. 02 - 11	Onsdag d. 03 - 11	Torsdag d. 04 - 11	Fredag d. 05 - 11
08:30 - 10:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-S.30	
10:30 - 12:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-S.30	
12:30 - 14:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-S.30	
14:30 - 16:00					

Uge 45

	Mandag d. 08 - 11	Tirsdag d. 09 - 11	Onsdag d. 10 - 11	Torsdag d. 11 - 11	Fredag d. 12 - 11
08:30 - 10:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-S.30	

EXAM

Individual digitally written 1-hour exam (multiple choice)

Online in WISEFlow

Yes, you can do it remotely

EXAMS - 1ST SEMESTER

Date	Course	Type	Censor/examinator
27.10.2022	Basic Databases	1-hour online written exam, Multiple Choice in WISEflow	Internal/KATO
04.01.2022	Frontend Programming	1-hour online written exam, Multiple Choice in WISEflow	Internal/RACE
10.01.2022 - 12.01.2022	Interaction and experience design	Individual oral examination – 30 min. prep + 30 min. examination	External/SBJ
18.01.2022 - 20.01.2022	Backend Programming	Individual oral examination – 30 min. prep + 30 min. examination	External/KATO

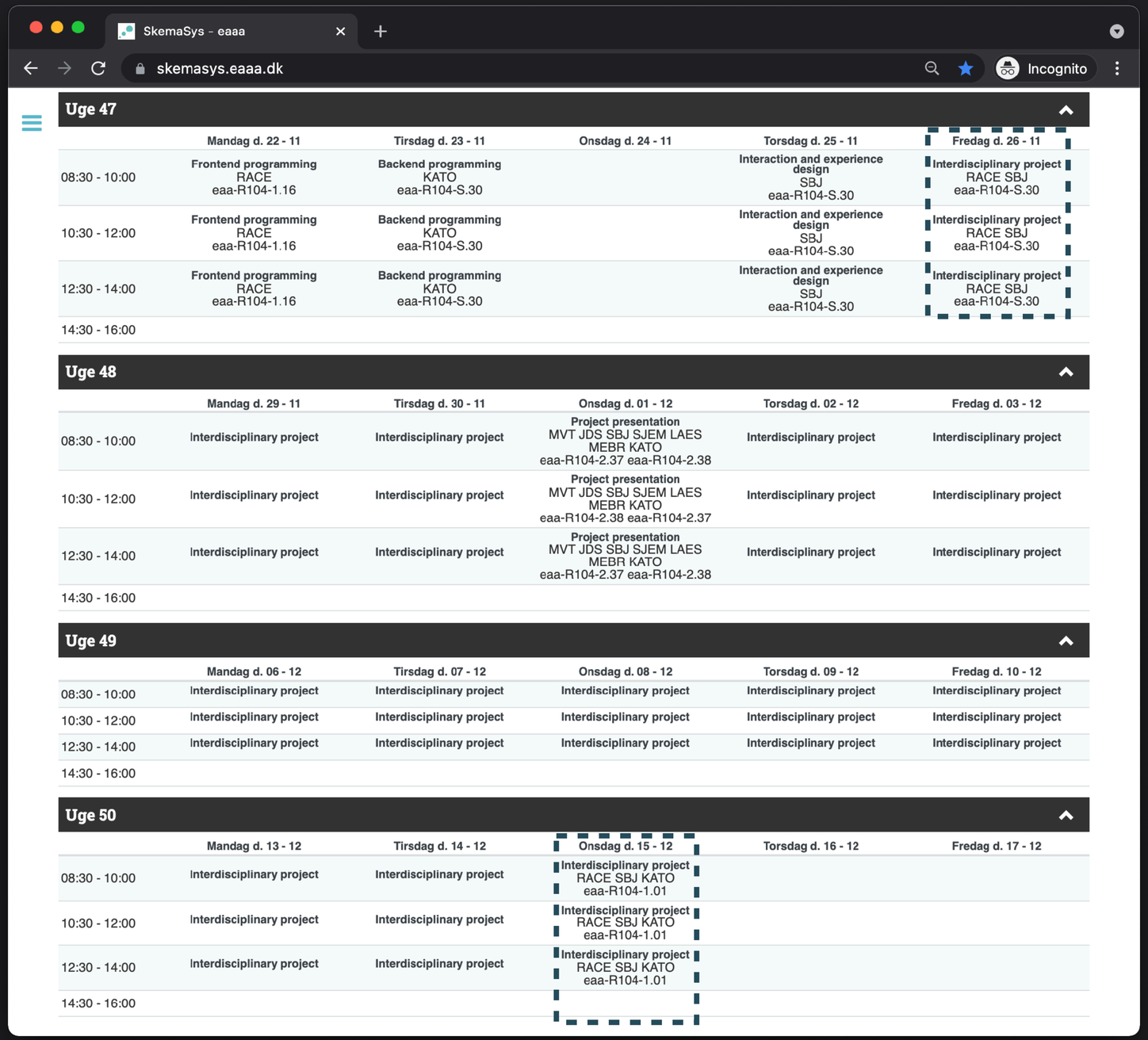
https://eaaa.instructure.com/courses/13351/pages/important-dates-interdisciplinary-project-and-exams

INTERDISCIPLINARY PROJECT

In groups, you have to develop a full-stack web app demonstrating the main parts of your learned skills and competencies from the 1st semester (Interaction and experience design, Databases, Backend & Frontend).

The project is finalized by a group presentation of your web app. The project will be used in addition to your exams in January.

Format: Full-stack Web App developed in groups of 2-3 students and a group presentation.
Project Kick-Off: November 26
Hand in: December 15
Group presentations: December 15



Uge 47

	Mandag d. 22 - 11	Tirsdag d. 23 - 11	Onsdag d. 24 - 11	Torsdag d. 25 - 11	Fredag d. 26 - 11
08:30 - 10:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-S.30	Interdisciplinary project RACE SBJ eaa-R104-S.30
10:30 - 12:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-S.30	Interdisciplinary project RACE SBJ eaa-R104-S.30
12:30 - 14:00	Frontend programming RACE eaa-R104-1.16	Backend programming KATO eaa-R104-S.30		Interaction and experience design SBJ eaa-R104-S.30	Interdisciplinary project RACE SBJ eaa-R104-S.30
14:30 - 16:00					

Uge 48

	Mandag d. 29 - 11	Tirsdag d. 30 - 11	Onsdag d. 01 - 12	Torsdag d. 02 - 12	Fredag d. 03 - 12
08:30 - 10:00	Interdisciplinary project	Interdisciplinary project	Project presentation MVT JDS SBJ SJEM LAES MEBR KATO eaa-R104-2.37 eaa-R104-2.38	Interdisciplinary project	Interdisciplinary project
10:30 - 12:00	Interdisciplinary project	Interdisciplinary project	Project presentation MVT JDS SBJ SJEM LAES MEBR KATO eaa-R104-2.38 eaa-R104-2.37	Interdisciplinary project	Interdisciplinary project
12:30 - 14:00	Interdisciplinary project	Interdisciplinary project	Project presentation MVT JDS SBJ SJEM LAES MEBR KATO eaa-R104-2.37 eaa-R104-2.38	Interdisciplinary project	Interdisciplinary project
14:30 - 16:00					

Uge 49

	Mandag d. 06 - 12	Tirsdag d. 07 - 12	Onsdag d. 08 - 12	Torsdag d. 09 - 12	Fredag d. 10 - 12
08:30 - 10:00	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project
10:30 - 12:00	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project
12:30 - 14:00	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project
14:30 - 16:00					

Uge 50

	Mandag d. 13 - 12	Tirsdag d. 14 - 12	Onsdag d. 15 - 12	Torsdag d. 16 - 12	Fredag d. 17 - 12
08:30 - 10:00	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project RACE SBJ KATO eaa-R104-1.01		
10:30 - 12:00	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project RACE SBJ KATO eaa-R104-1.01		
12:30 - 14:00	Interdisciplinary project	Interdisciplinary project	Interdisciplinary project RACE SBJ KATO eaa-R104-1.01		
14:30 - 16:00					

INTERDISCIPLINARY PROJECT

Case?
Clever Coffee? Dating App? Or something completely different?

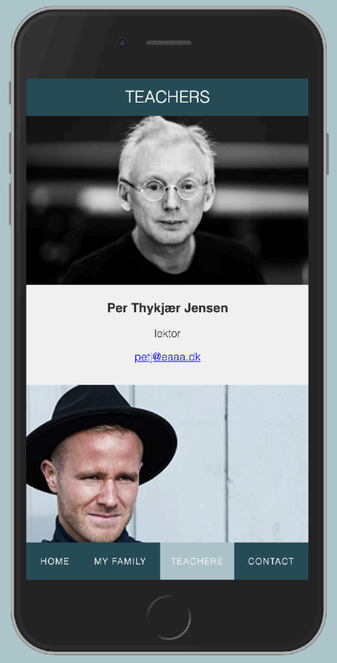
User Research will be a part of Interaction & Experience Design (before project kick-off).

INTERDISCIPLINARY PROJECT

FRONTEND

(API)

BACKEND



HTML JS CSS

FETCH(...)

REQUEST

JSON DATA

RESPONSE

JSON

php

SQL

MySQL

TOOLS

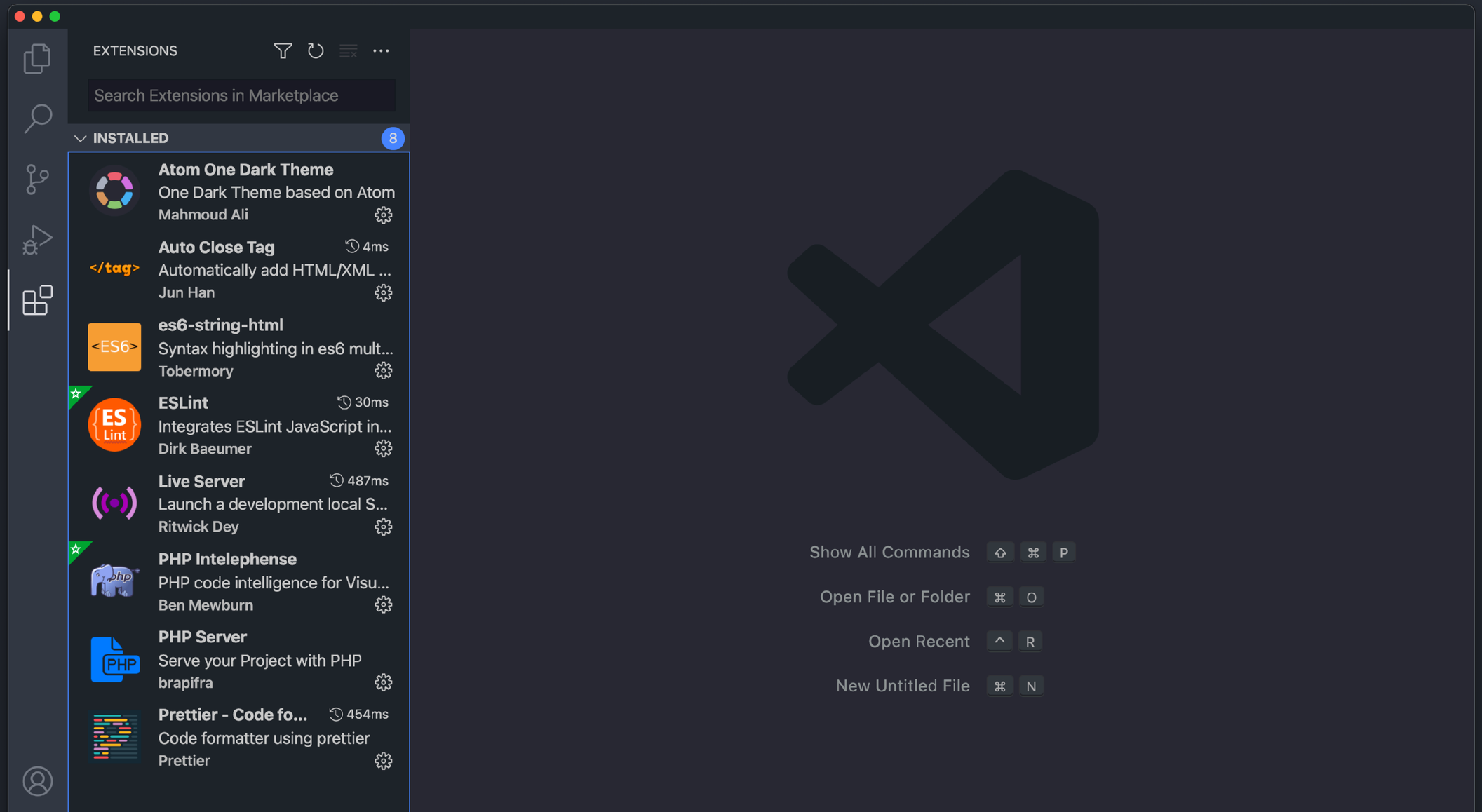
VISUAL STUDIO CODE

EXTENSIONS: ESLINT & LIVE SERVER

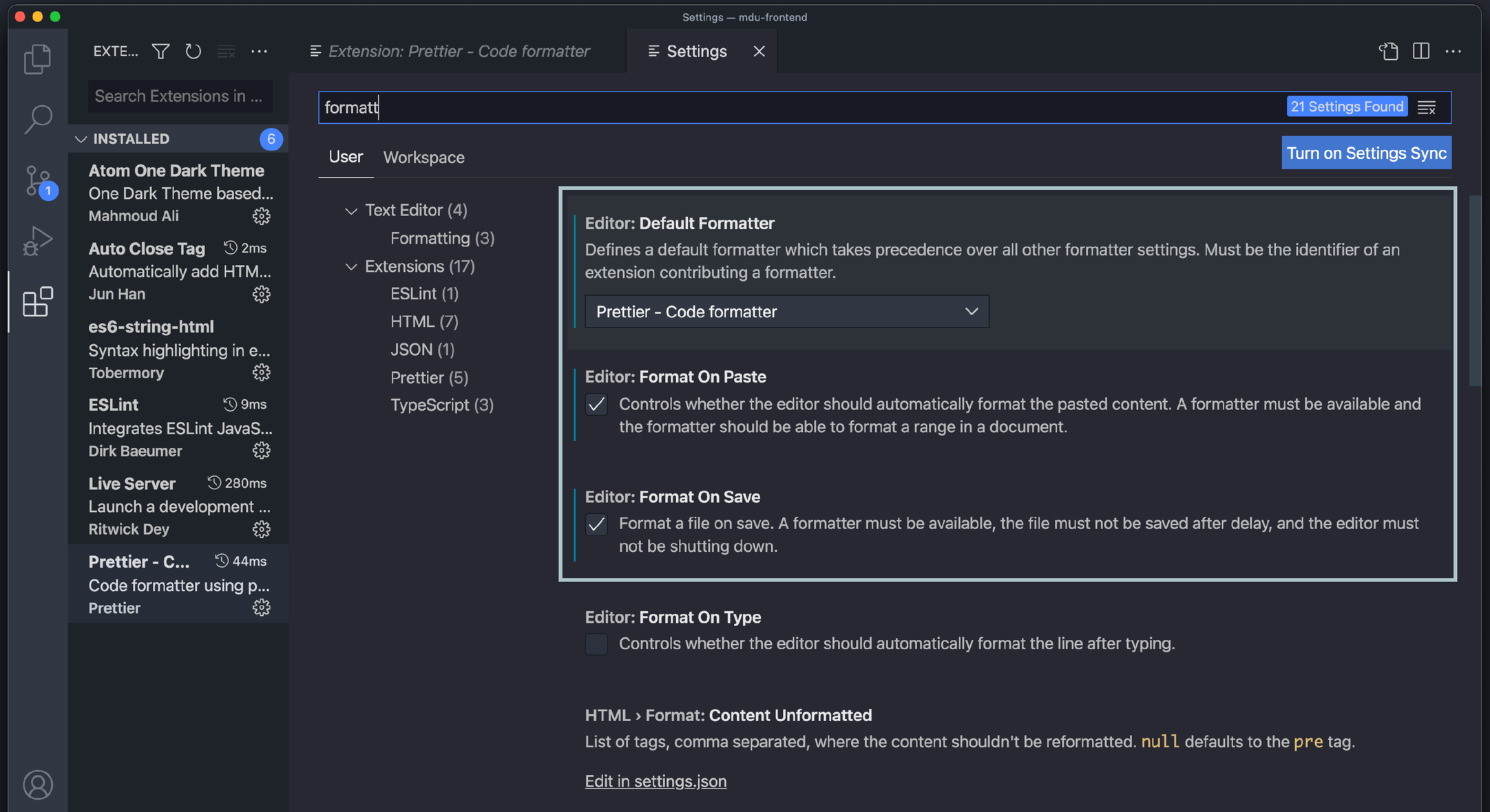
GITHUB DESKTOP

OPTIONAL: FTP CLIENT

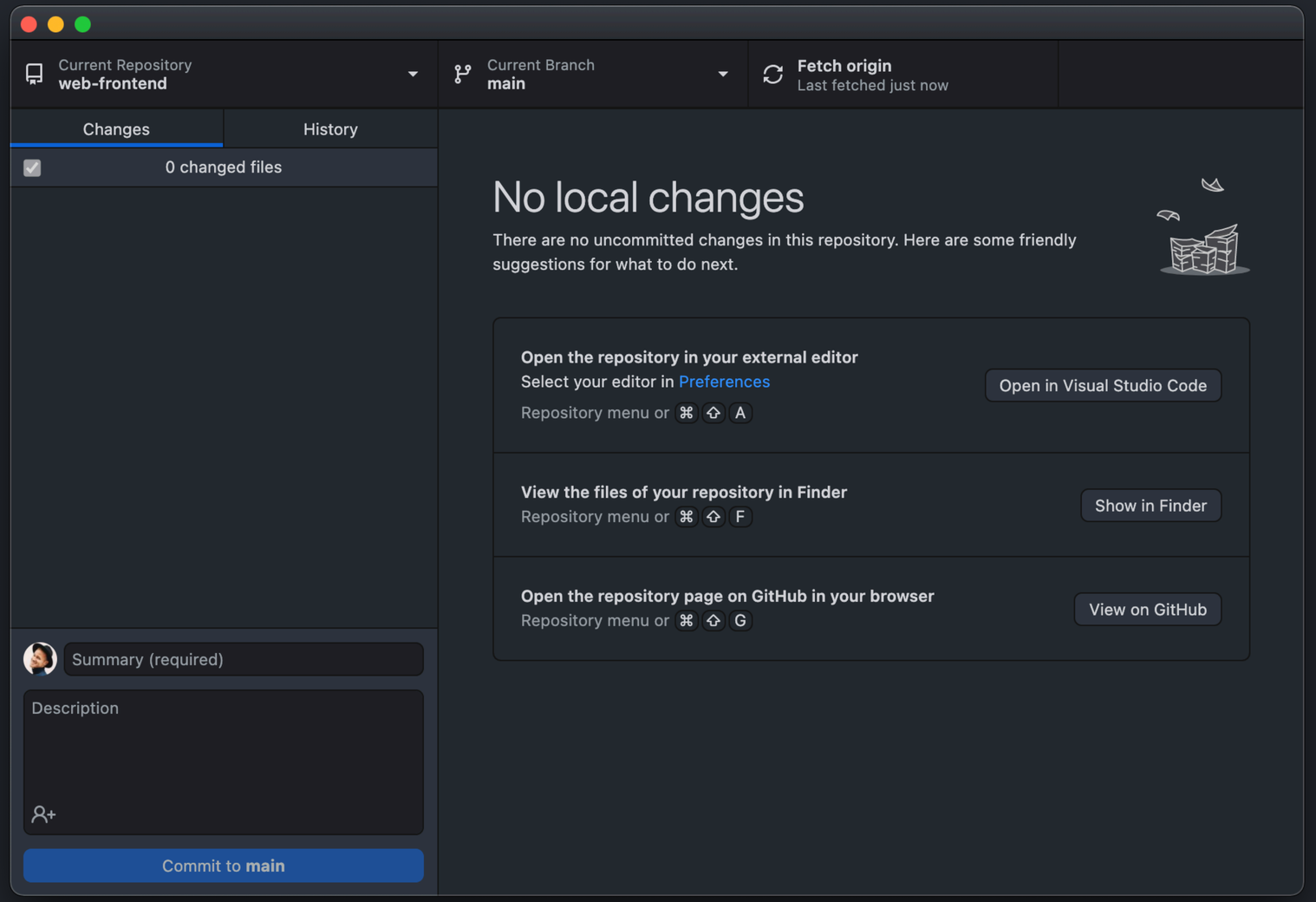
EXTENSIONS


EXTENSIONS
Search Extensions in Marketplace
INSTALLED
8
Atom One Dark Theme
One Dark Theme based on Atom
Mahmoud Ali
Auto Close Tag
4ms
Automatically add HTML/XML ...
Jun Han
es6-string-html
Syntax highlighting in es6 mult...
Tobermory
ESLint
30ms
Integrates ESLint JavaScript in...
Dirk Baeumer
Live Server
487ms
Launch a development local S...
Ritwick Dey
PHP Intelephense
PHP code intelligence for Visu...
Ben Mewburn
PHP Server
Serve your Project with PHP
brapifra
Prettier - Code fo...
454ms
Code formatter using prettier
Prettier
Show All Commands
Open File or Folder
Open Recent
New Untitled File

CHOOSE DEFAULT FORMMATTER IN SETTINGS



FETCH & PULL LATEST



WORKFLOW

1. FETCH & PULL LATEST FROM GITHUB REPO WEB-FRONTEND

2. CREATE YOUR OWN LOCAL DEV FOLDER

3. COPY PROJECTS FROM WEB-FRONTEND INTO YOUR OWN DEV FOLDER.

4. OPEN YOUR OWN LOCAL DEV FOLDER IN VS CODE.

5. USE LIVE SERVER OR PHP SERVER EXTENSION TO RUN PROJECTS (PHP SERVER WHEN RUNNING PHP PROJECTS 🤷‍♂️).

6. START WORKING IN YOUR OWN LOCAL DEV FOLDER.

OPTIONAL: SETUP GIT FOR YOUR OWN DEV FOLDER

SETUP DEV ENVIRONMENT

1. INSTALL EXTENSIONS.

2. CHOOSE DEFAULT FORMATTER.

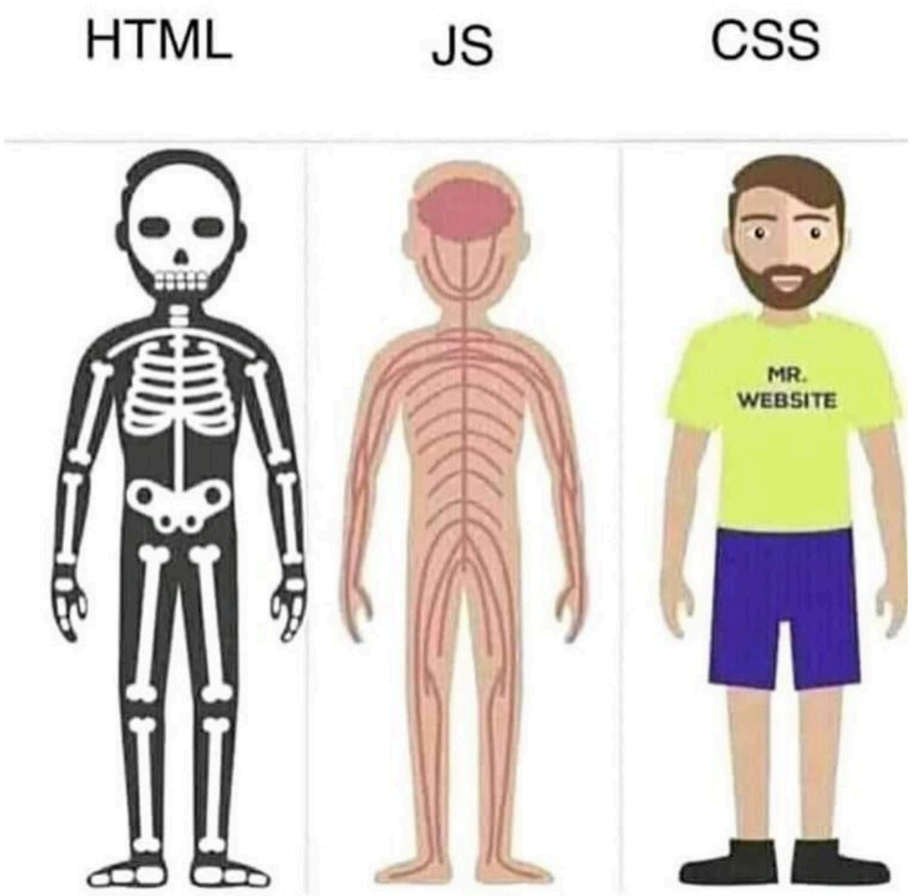
3. FOLLOW BULLETS ON THE WORKFLOW SLIDE.

FRONTEND STACK

STRUCTURE
CONTENT

FUNCTIONALITY
BEHAVIOR

LAYOUT/STYLING
PRESENTATION



https://www.instagram.com/p/Bw90x2Ojo4g/

JAVASCRIPT, HTML & CSS

YOU CAN LITERALLY BUILD ANYTHING WITH IT

SEPARATION OF CONCERNS

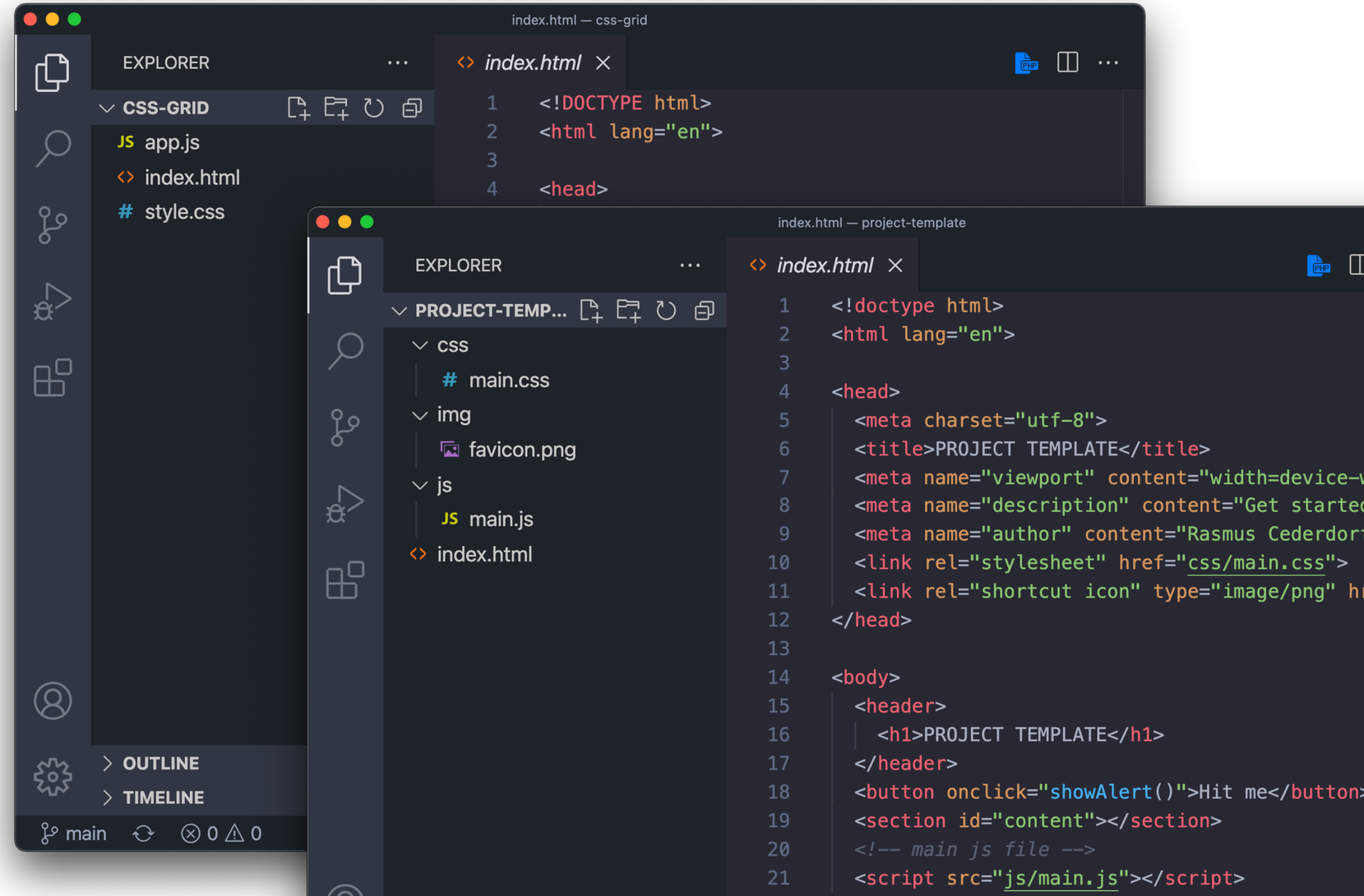
STRUCTURE
CONTENT

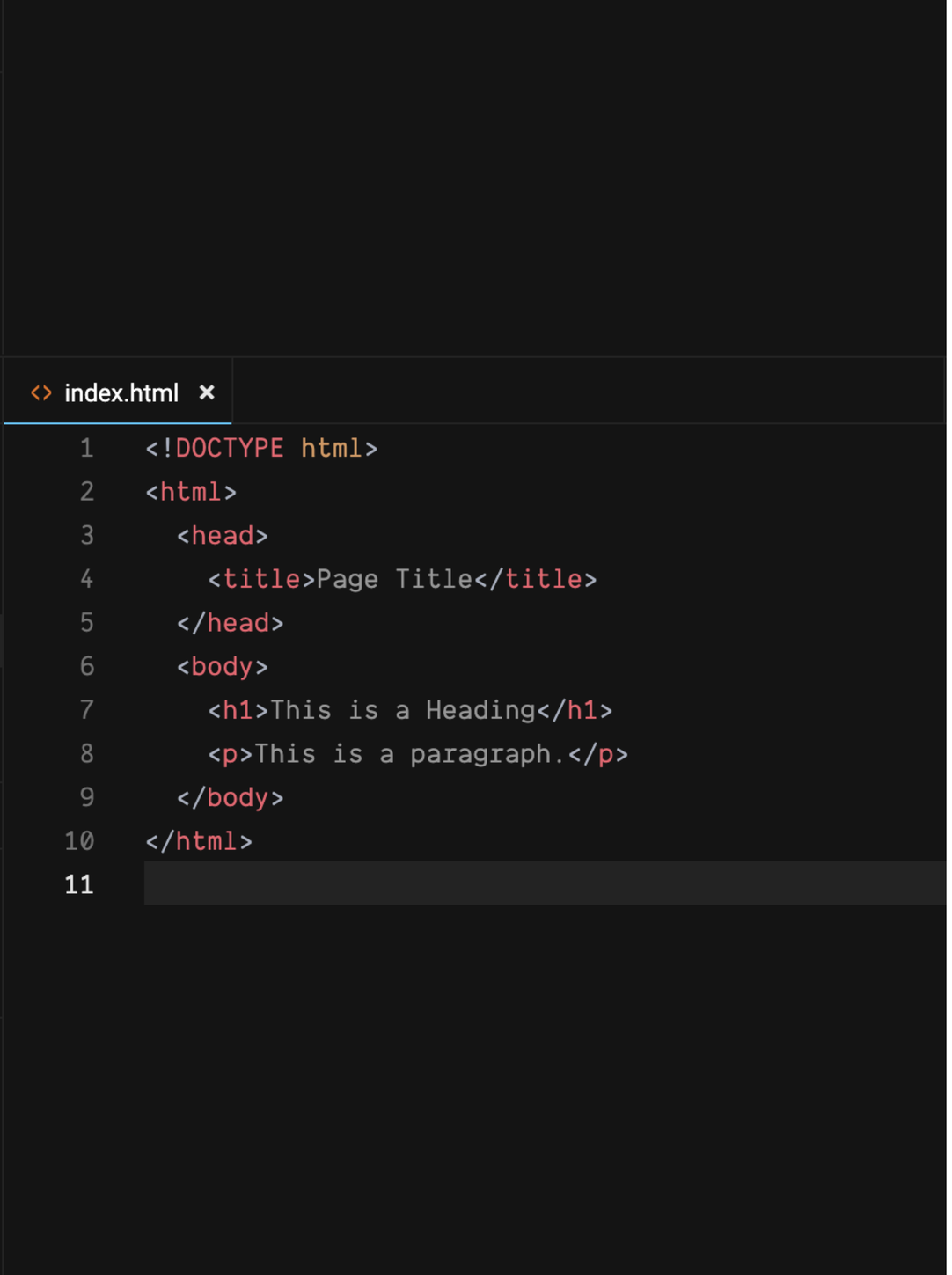
FUNCTIONALITY
BEHAVIOR

LAYOUT/STYLING
PRESENTATION

FRONTEND PROJECT STRUCTURE

- Project structure:
  - HTML
  - CSS
  - JavaScript
- Keep a good structure
- Separation of concerns
- Naming & conventions





WHAT IS HTML?

Hyper Text Markup Language

Standard markup language for creating Web pages

Describes the structure of a Web page

Consists of a series of elements

HTML elements tell the browser how to display the content

https://www.w3schools.com/html/html_intro.asp

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

Browser Tests

https://9nypo.csb.app/

This is a Heading

This is a paragraph.

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document

The `<html>` element is the root element of an HTML page

The `<head>` element contains meta information about the HTML page

The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

The `<h1>` element defines a large heading

The `<p>` element defines a paragraph

https://www.w3schools.com/html/html_intro.asp

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

https://www.w3schools.com/html/html_intro.asp

HTML SEMANTIC ELEMENTS

... clearly describes its meaning to both the browser and the developer.

https://www.w3schools.com/html/html5_semantic_elements.asp

WHY USE SEMANTIC ELEMENTS?

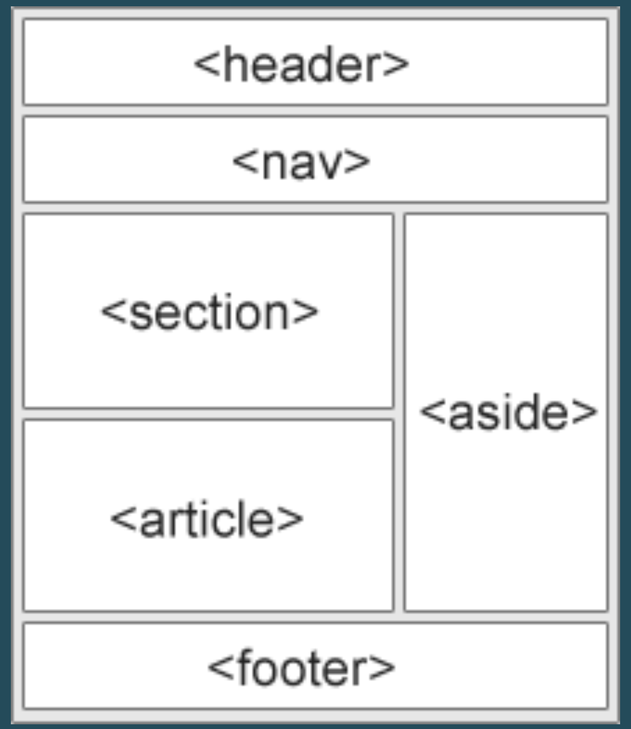
```
<body>
  <div id="header">
    <h1>PROJECT TEMPLATE</h1>
  </div>
  <div class="sections">
    <div class="article">
      <div class="figure">
        <img>
        <div class="figcaption"></div>
      </div>
    </div>
  </div>
  <div id="footer"></div>
  <!-- main js file -->
  <script src="js/main.js"></script>
</body>
```

```
<body>
  <header>
    <h1>PROJECT TEMPLATE</h1>
  </header>
  <section>
    <article>
      <figure>
        <img>
        <figcaption></figcaption>
      </figure>
    </article>
  </section>
  <footer></footer>
  <!-- main js file -->
  <script src="js/main.js"></script>
</body>
```

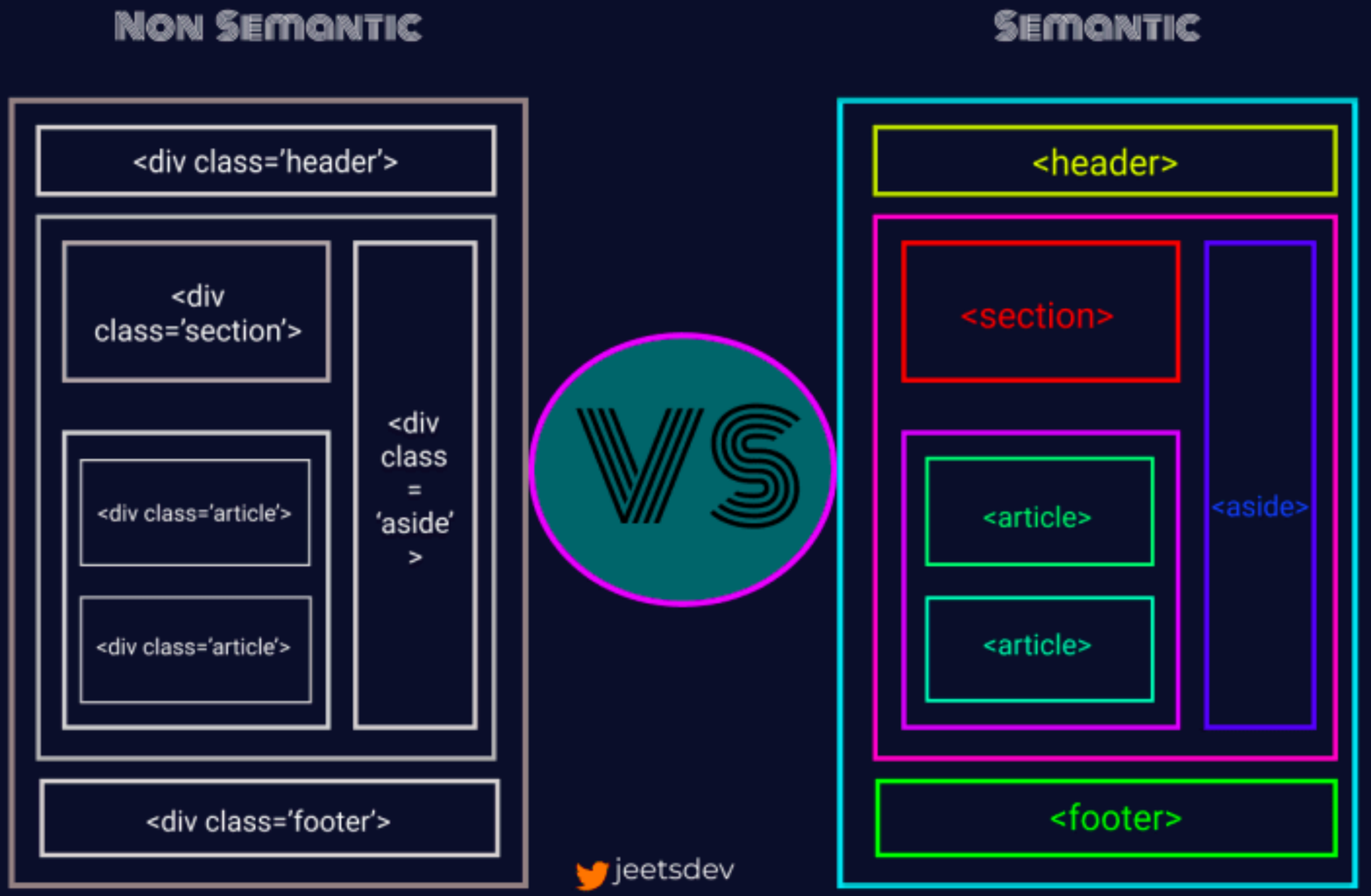
HTML SEMANTIC ELEMENTS

Semantic elements that can be used to define different parts of a website

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



https://www.w3schools.com/html/html5_semantic_elements.asp


Non Semantic

VS
jeetsdev
Semantic
<header>
<section>
<article>
<article>
<aside>
<footer>

DIV TAGS?

Many web sites contain HTML code like:

```
<div id=“nav">
<div class=“header">
<div id="footer">
```

to indicate navigation, header, and footer.

Can we still use `div` tags?

https://www.w3schools.com/html/html5_semantic_elements.asp

Apple

apple.com

Incognito

div 971 × 64

Store Mac iPad iPhone Watch TV Music Support

We look forward to welcoming you to our stores. Whether you shop in a store or shop online, our Specialists can help you buy the products you love.

iPhone 13 Pro

Oh. So. Pro.

Learn more › Buy ›

iPhone 13

Your new superpower.

Learn more › Buy ›

Elements Console Sources Network 1

```
<aside id="ac-localeswitcher" data-analytics-region="locale switcher" data-analytics-activitymap-region-id="locale switcher" lang="da-DK" dir="ltr">…</aside>
<h1 class="visuallyhidden">Apple</h1>
<meta name="ac-gn-store-key" content="SFX9YPYY9PPXCU9KH">
<aside id="ac-gn-segmentbar" class="ac-gn-segmentbar" lang="en-US" dir="ltr" data-strings="{ 'exit': 'Exit', 'view': '{%STOREFRONT%} Store Home', 'segments': { 'smb': 'Business Store Home', 'eduInd': 'Education Store Home', 'other': 'Store Home' } }"></aside>
<input type="checkbox" id="ac-gn-menustate" class="ac-gn-menustate">
<nav id="ac-globalnav" class="js no-touch no-windows" role="navigation" aria-label="Global" data-hires="false" data-analytics-region="global nav" lang="en-US" dir="ltr" data-www-domain="www.apple.com" data-store-locale="us" data-store-root-path="/us" data-store-api="/[storefront]/shop/bag/status" data-search-locale="en_US" data-search-suggestions-api="/search-services/suggestions/" data-search-defaultlinks-api="/search-services/suggestions/defaultlinks/">…</nav>
<div class="ac-gn-blur"></div> == $0
<div id="ac-gn-curtain" class="ac-gn-curtain"></div>
<div id="ac-gn-placeholder" class="ac-nav-placeholder"></div>
<script type="text/javascript" src="/ac/globalnav/6/en_US/scripts/ac-globalnav.built.js"></script>
<div id="ac-gn-viewport-emitter">…</div>
<script src="/metrics/ac-analytics/2.13.1/scripts/ac-analytics.js" type="text/javascript" charset="utf-8"></script>
<main class="main" role="main">
<section class="homepage-section collection-module" data-module-template="ribbon"> grid
<div data-unit-id="shop-online" data-analytics-activitymap-region-id="ribbon-shop-online">…</div>
</section>
<section class="homepage-section collection-module" data-module-template="heroes" data-analytics-region="hero"> grid
<div data-unit-id="iphone-13-pro" data-analytics-section-engagement="name:hero-iphone-13-pro">…</div>
<div data-unit-id="iphone-13" data-analytics-section-engagement="nam
```

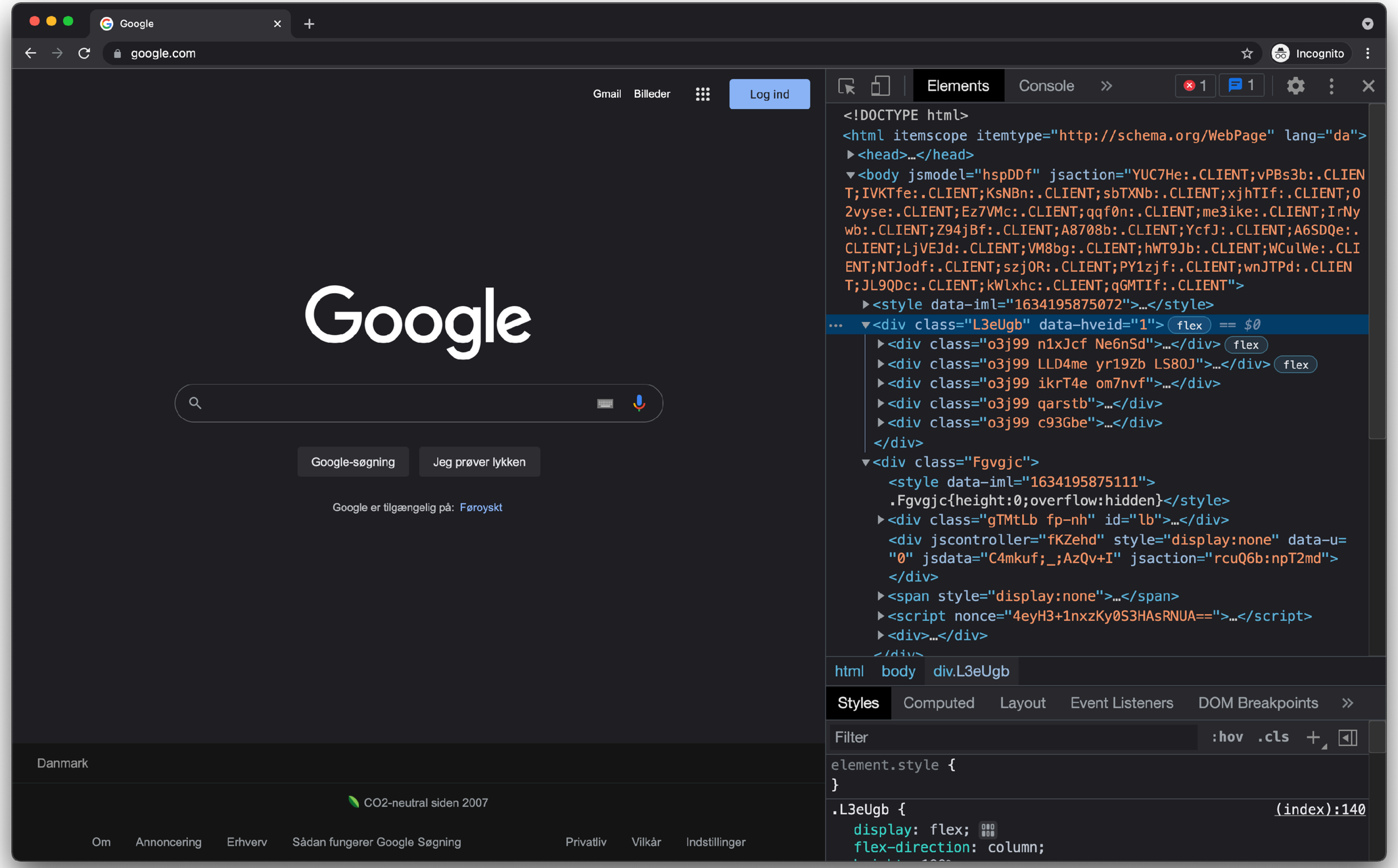
… .enhanced-layout body.page-home.ac-nav-overlap.body-with-ribbon div.ac-gn-blur …

Styles Computed Layout Event Listeners DOM Breakpoints Properties

Filter :hov .cls

```
element.style {
}

html #ac-globalnav, html #ac-globalnav ~ .ac-gn-blur, html #ac-gn-segmentbar, html #ac-localeswitcher, html div.adv-wrapper {
  position: fixed;
}
```

main.built.css:1


Google
google.com
Incognito
Gmail
Billeder
Log ind
Google
Google-søgning
Jeg prøver lykken
Google er tilgængelig på: Føroyskt
Danmark
CO2-neutral siden 2007
Om
Annoncering
Erhverv
Sådan fungerer Google Søgning
Privatliv
Vilkår
Indstillinger
Elements
Console
<!DOCTYPE html>
<html itemscope itemtype="http://schema.org/WebPage" lang="da">
<head>…</head>
<body jsmodel="hspDDf" jsaction="YUC7He:.CLIENT;vPBs3b:.CLIENT;IVKTfe:.CLIENT;KsNBn:.CLIENT;sbTXNb:.CLIENT;xjhTIf:.CLIENT;O2vyse:.CLIENT;Ez7VMc:.CLIENT;qqf0n:.CLIENT;me3ike:.CLIENT;IrNywb:.CLIENT;Z94jBf:.CLIENT;A8708b:.CLIENT;YcfJ:.CLIENT;A6SDQe:.CLIENT;LjVEJd:.CLIENT;VM8bg:.CLIENT;hWT9Jb:.CLIENT;WCulWe:.CLIENT;NTJodf:.CLIENT;szjOR:.CLIENT;PY1zjf:.CLIENT;wnJTPd:.CLIENT;JL9QDc:.CLIENT;kWlxhc:.CLIENT;qGMTIf:.CLIENT">
<style data-iml="1634195875072">…</style>

…
…
…
…
…

<style data-iml="1634195875111">
.Fgvgjc{height:0;overflow:hidden}</style>
…

…
<script nonce="4eyH3+1nxzKy0S3HAsRNUA==">…</script>
…
html
body
div.L3eUgb
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Filter
:hov
.cls
element.style {
}
.L3eUgb {
(index):140
display: flex;
flex-direction: column;

HTML Semantic Elements

w3schools.com/html/html5_semantic_elements.asp

Incognito

Tutorials References Exercises Website Paid Courses Log in

HTML CSS JAVASCRIPT SQL PYTHON PHP BOOTSTRAP

HTML Semantic Elements

❮ Previous Next ❯

Semantic elements = elements with a meaning.

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

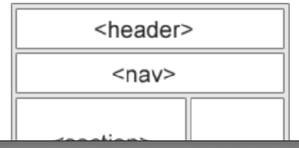
Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

Semantic Elements in HTML

Many web sites contain HTML code like: to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>



Elements Console 2 2 28

```
<body> == $0
  <div class="w3-bar w3-card-2 notranslate" style="position: relative;z-index: 4;font-size: 18px;background-color: white;color:#282A35;padding-left:12px;padding-right:16px;font-family: 'Source Sans Pro', sans-serif;">…</div>
  <div style="display:none;position:absolute;z-index:4;right:52px;height:44px;background-color:#282A35;letter-spacing:normal;" id="googleSearch">…</div>
  <div style="display:none;position:absolute;z-index:3;right:111px;height:44px;background-color:#282A35;text-align:right;padding-top:9px;" id="google_translate_element"></div>
  <div class="w3-card-2 topnav notranslate" id="topnav" style="position: relative;">…</div>
  <div id="myAccordion" class="w3-card-2 w3-center w3-hide-large w3-hide-medium" style="width: 100%; position: absolute; display: none; background-color: rgb(231, 233, 235); padding-top: 0px;">…</div>
  <script> </script>
  <div class="w3-sidebar w3-collapse" id="sidenav" style="top: 118px; display: none;">…</div>
  <div class="w3-main w3-light-grey" id="belowtopnav" style="margin-left: 220px; padding-top: 0px;">
    <div class="w3-row w3-white">
      ::before
      <div class="w3-col l10 m12" id="main">
        <div id="mainLeaderboard" style="overflow:hidden;">…</div>
        <h1>…</h1>
        <div class="w3-clear nextprev"> </div>
```

html body

Styles Computed Layout Event Listeners DOM Breakpoints

Filter :hov .cls

```
element.style {
}
html, body {                                    w3schools28.css:27
  font-family: Verdana,sans-serif;
  font-size: 15px;
```

DIV TAGS?

Use semantic elements as much as possible (!)

... but let it not stop you from building something amazing 🎉

- RACE.js

https://www.w3schools.com/html/html5_semantic_elements.asp

WHAT IS CSS?

CSS stands for **C**ascading **S**tyle **S**heets

CSS describes how HTML elements are to be displayed on screen, paper, or in other media

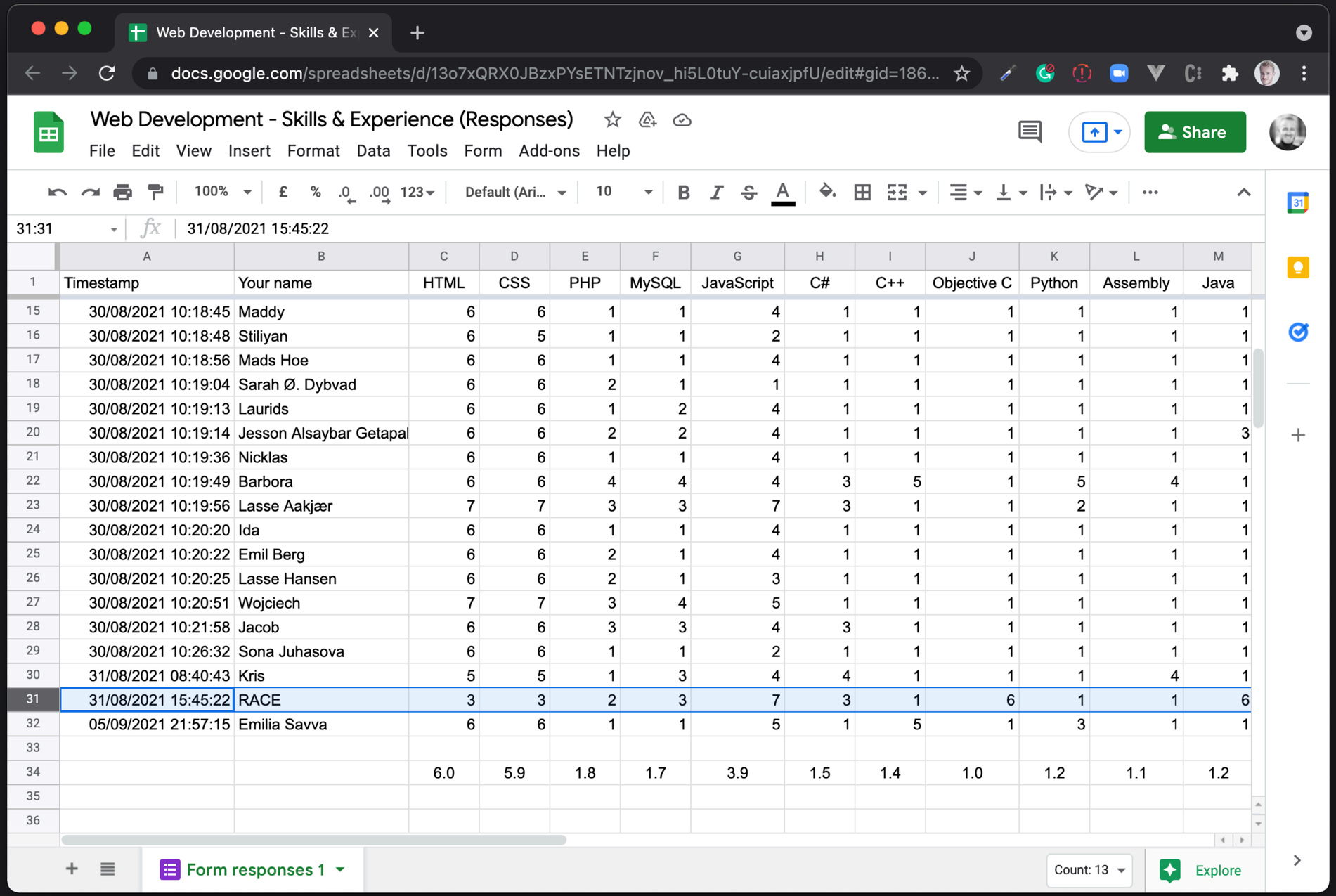
CSS **saves a lot of work**. It can control the layout of multiple web pages all at once

External stylesheets are stored in **CSS files**

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

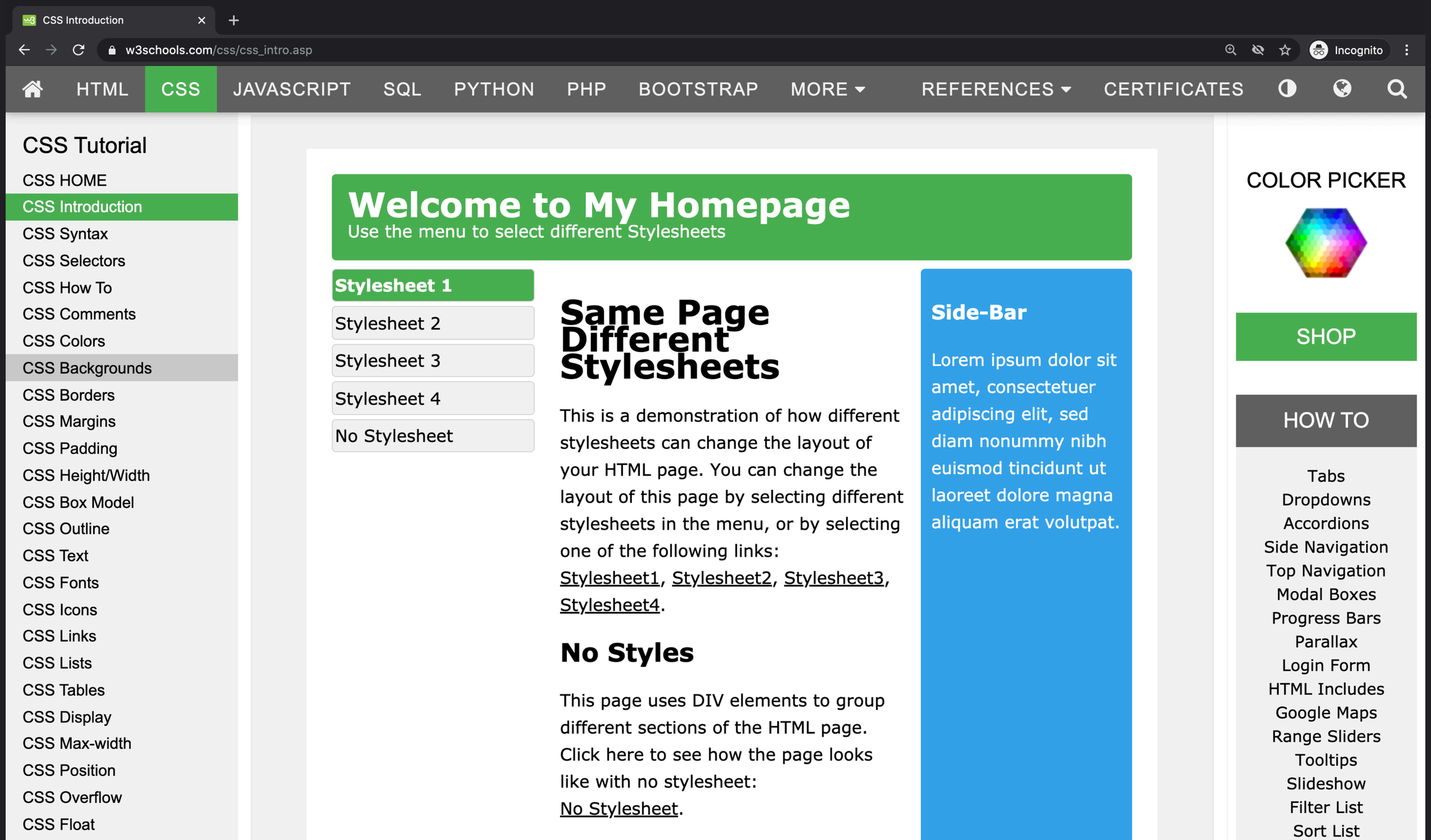
styles.css

```
1   body {
2     font-family: Helvetica, Arial, sans-serif;
3     text-align: center;
4   }
5
6   h1 {
7     font-weight: lighter;
8   }
9
10  #my-paragraph {
11    color: blue;
12  }
13
14  .big {
15    font-size: 25px;
16  }
17
```



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Timestamp	Your name	HTML	CSS	PHP	MySQL	JavaScript	C#	C++	Objective C	Python	Assembly	Java
15	30/08/2021 10:18:45	Maddy	6	6	1	1	4	1	1	1	1	1	1
16	30/08/2021 10:18:48	Stiliyan	6	5	1	1	2	1	1	1	1	1	1
17	30/08/2021 10:18:56	Mads Hoe	6	6	1	1	4	1	1	1	1	1	1
18	30/08/2021 10:19:04	Sarah Ø. Dybvad	6	6	2	1	1	1	1	1	1	1	1
19	30/08/2021 10:19:13	Laurids	6	6	1	2	4	1	1	1	1	1	1
20	30/08/2021 10:19:14	Jesson Alsaybar Getapal	6	6	2	2	4	1	1	1	1	1	3
21	30/08/2021 10:19:36	Nicklas	6	6	1	1	4	1	1	1	1	1	1
22	30/08/2021 10:19:49	Barbora	6	6	4	4	4	3	5	1	5	4	1
23	30/08/2021 10:19:56	Lasse Aakjær	7	7	3	3	7	3	1	1	2	1	1
24	30/08/2021 10:20:20	Ida	6	6	1	1	4	1	1	1	1	1	1
25	30/08/2021 10:20:22	Emil Berg	6	6	2	1	4	1	1	1	1	1	1
26	30/08/2021 10:20:25	Lasse Hansen	6	6	2	1	3	1	1	1	1	1	1
27	30/08/2021 10:20:51	Wojciech	7	7	3	4	5	1	1	1	1	1	1
28	30/08/2021 10:21:58	Jacob	6	6	3	3	4	3	1	1	1	1	1
29	30/08/2021 10:26:32	Sona Juhasova	6	6	1	1	2	1	1	1	1	1	1
30	31/08/2021 08:40:43	Kris	5	5	1	3	4	4	1	1	1	4	1
31	31/08/2021 15:45:22	RACE	3	3	2	3	7	3	1	6	1	1	6
32	05/09/2021 21:57:15	Emilia Savva	6	6	1	1	5	1	5	1	3	1	1
33													
34			6.0	5.9	1.8	1.7	3.9	1.5	1.4	1.0	1.2	1.1	1.2
35													
36													

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



index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

styles.css

```css
body {
  font-family: Helvetica, Arial, sans-serif;
  text-align: center;
}

h1 {
  font-weight: lighter;
}
```

Browser Tests

https://9nypo.csb.app/

This is a Heading

This is a paragraph.

CSS SELECTORS

CSS selectors are used to "find" (or select) the HTML elements you want to style.

Simple selectors (name, id, class)

Combinator selectors (a specific relationship between them)

Pseudo-class selectors (a certain state)

Pseudo-elements selectors (a part of an element)

Attribute selectors (attribute or attribute value)

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

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <p id="my-paragraph">This is another paragraph.</p>
    <p class="big">This is another paragraph.</p>
  </body>
</html>
```

styles.css

```css
body {
  font-family: Helvetica, Arial, sans-serif;
  text-align: center;
}

h1 {
  font-weight: lighter;
}

#my-paragraph {
  color: blue;
}

.big {
  font-size: 25px;
}
```

Browser Tests

https://9nypo.csb.app/

This is a Heading

This is a paragraph.

This is another paragraph.

This is another paragraph.

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <p id="my-paragraph">This is another paragraph.</p>
    <p class="big">This is another paragraph.</p>
    <img src="https://cederdorff.com/img/karolineshus_web.jpg" />
  </body>
</html>
```

styles.css

```css

#my-paragraph {
  color: blue;
}

.big {
  font-size: 25px;
}

img {
  width: 100%;
}

```

Browser Tests

https://9nypo.csb.app/

This is a Heading

This is a paragraph.

This is another paragraph.

This is another paragraph.

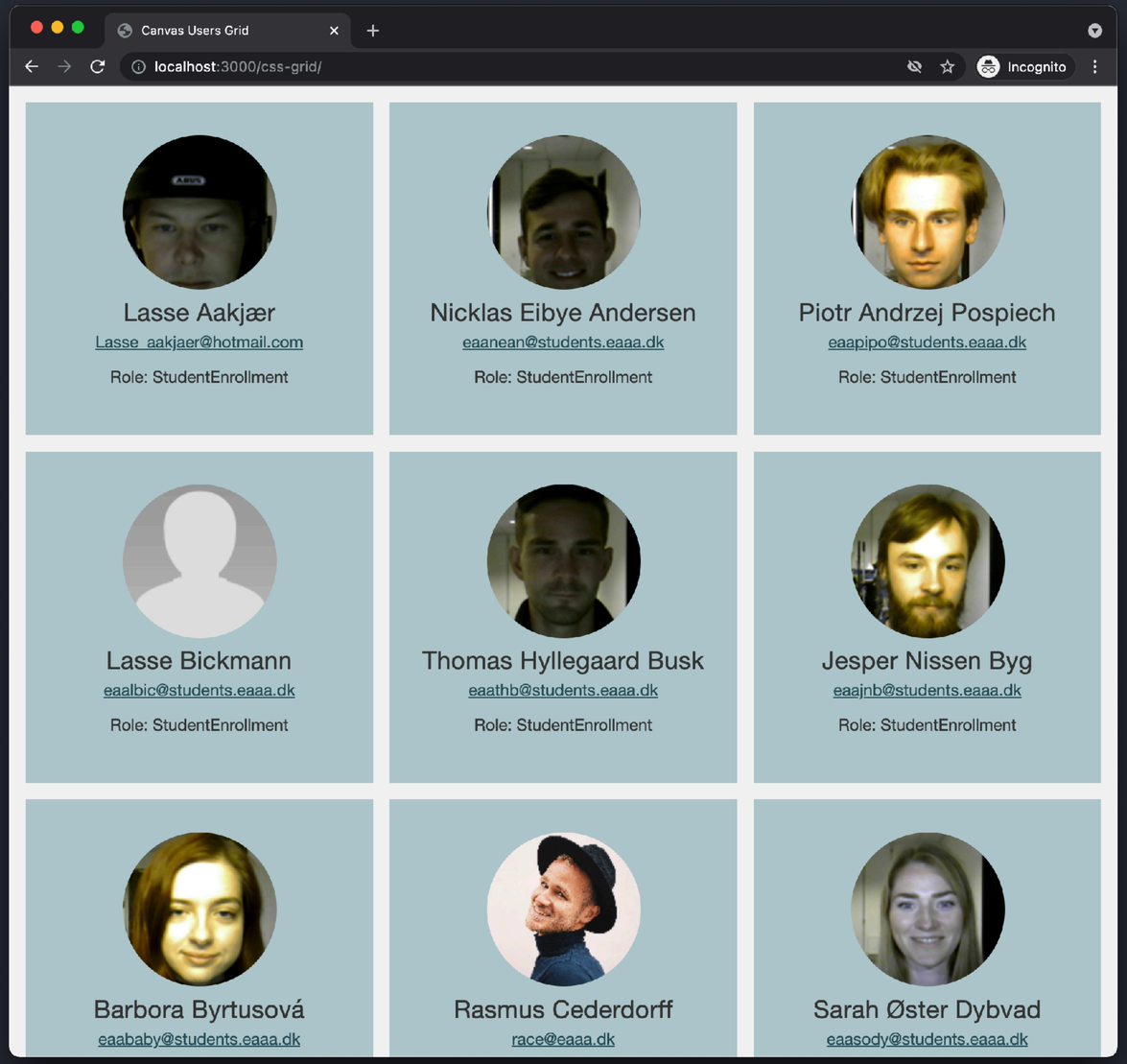
CSS LAYOUTS

GRID & FLEXBOX

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

... responsive and flexible layouts.

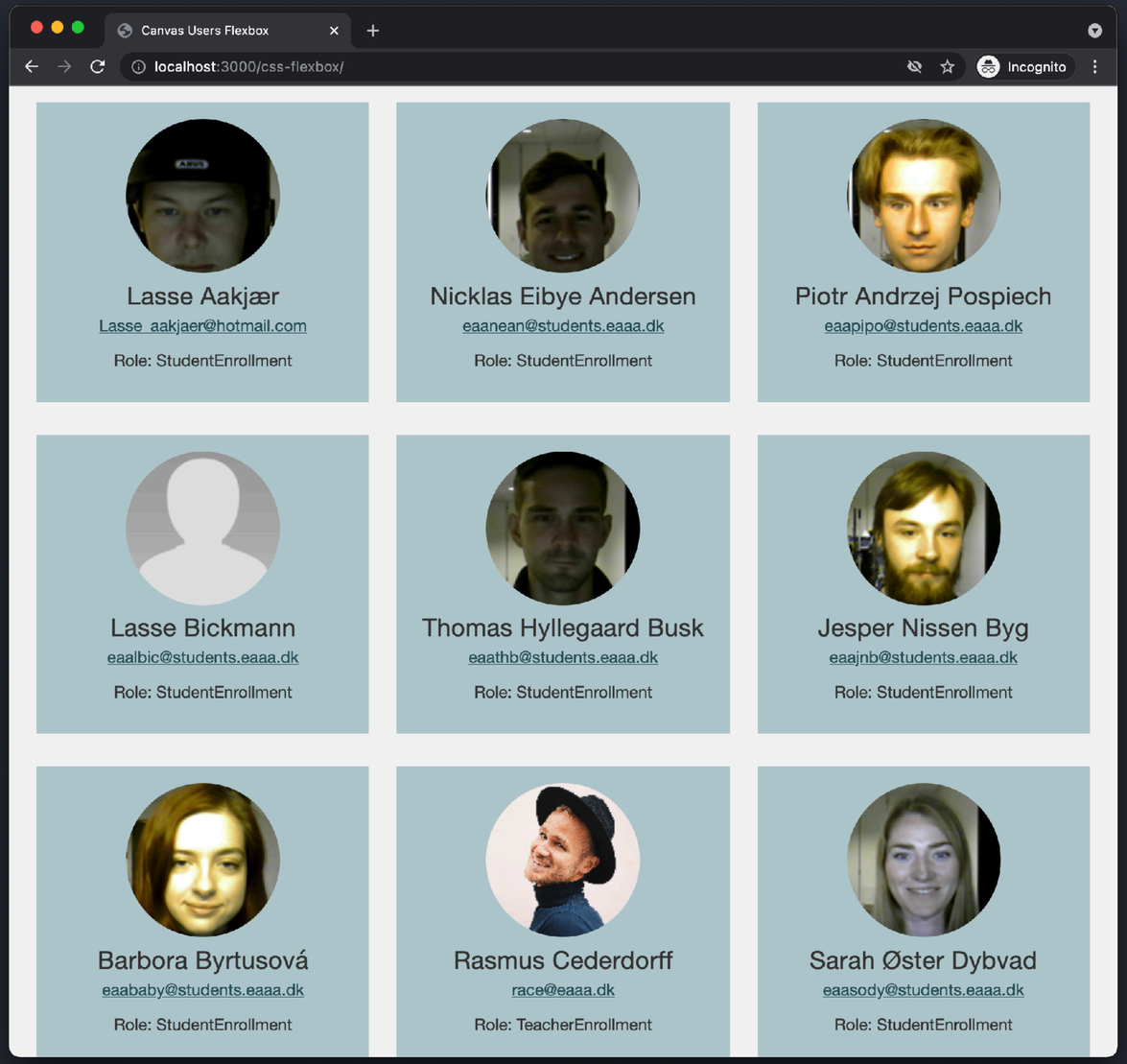
CSS GRID




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

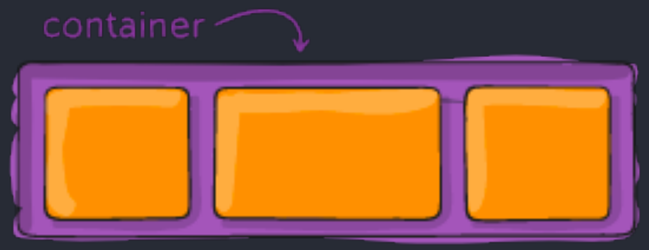
```css
/* ---------- grid styling ---------- */

.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;
  }
}
```

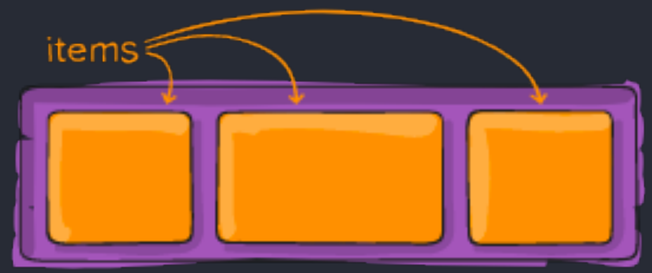
CSS FLEXBOX




```
<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 spa
  flex-grow: 1; /*ability for a flex item to grow. 1 = all items equally*
}

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

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

CSS GRID



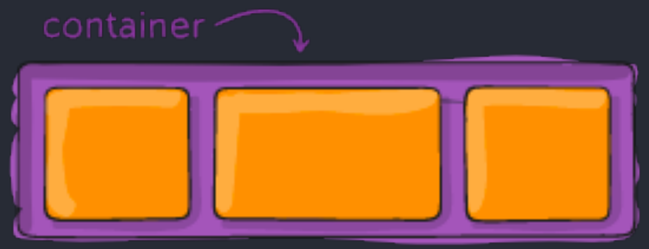


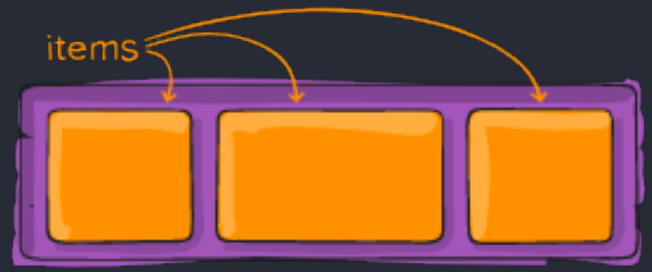

```css
.grid {
  display: grid;
  grid-template-columns: 1fr;
  padding: 1em;
  gap: 1em;
}

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

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

```css
.grid > article {
  text-align: center;
  padding: 2em 1em;
  background-color: var(--light-green);
  transition: 0.3s;
}

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

.grid > article img {
  width: 100%;
  max-width: 150px;
  height: auto;
  border-radius: 50%;
}
```

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

CSS FLEXBOX

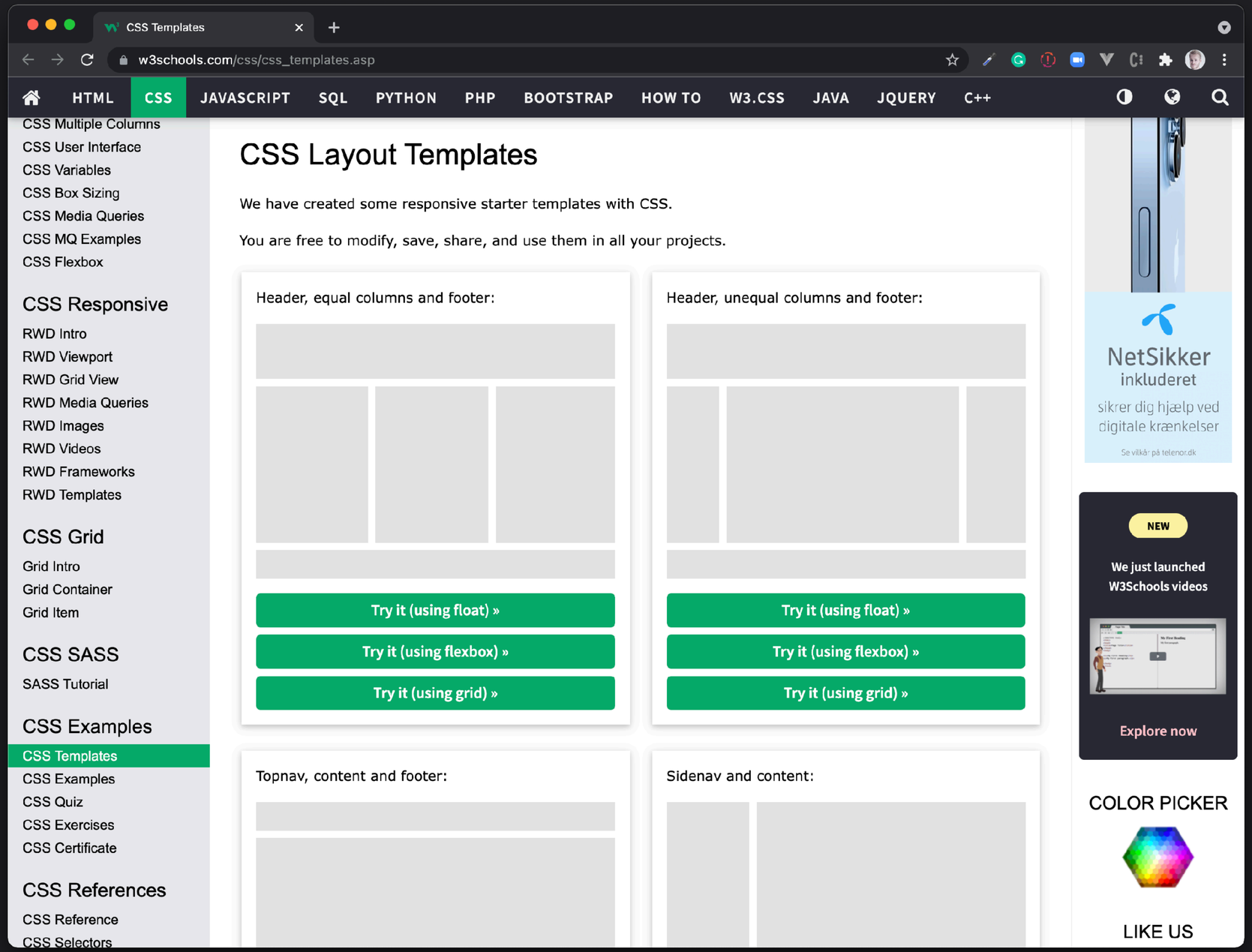




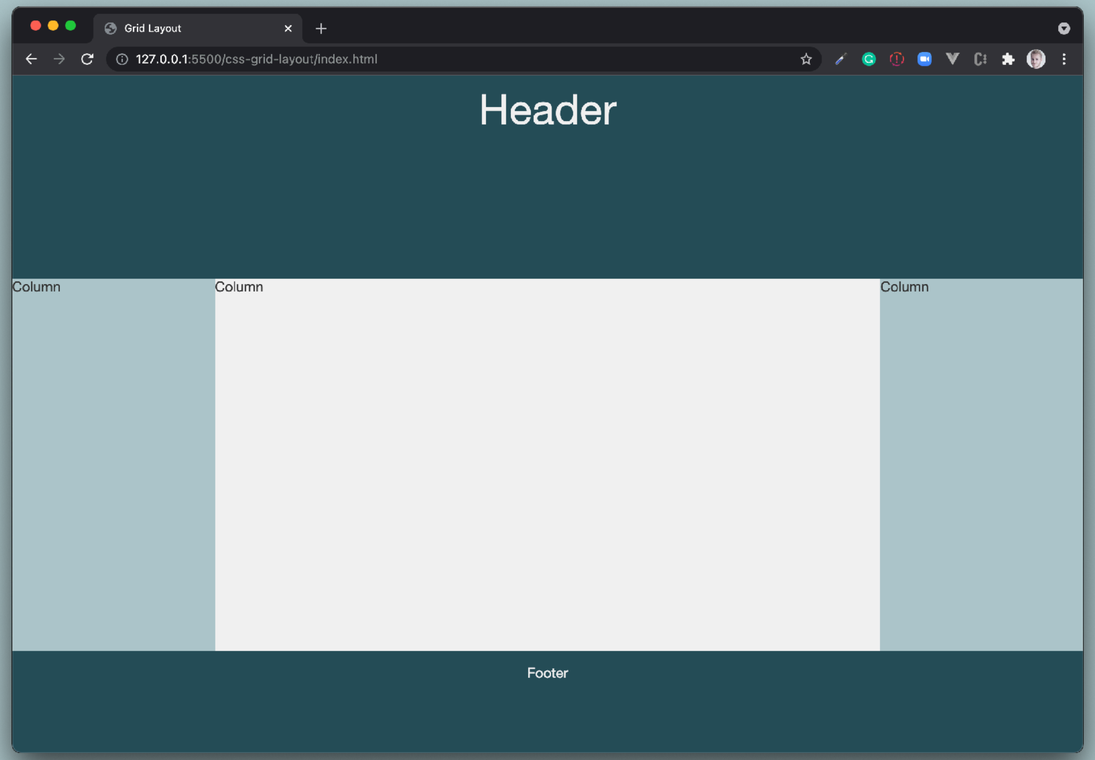

```css
.flexbox {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-evenly;
}
```

```css
.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 distrib
  flex-grow: 1; /*ability for a flex item to grow. 1 = all items equally*/
}

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

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

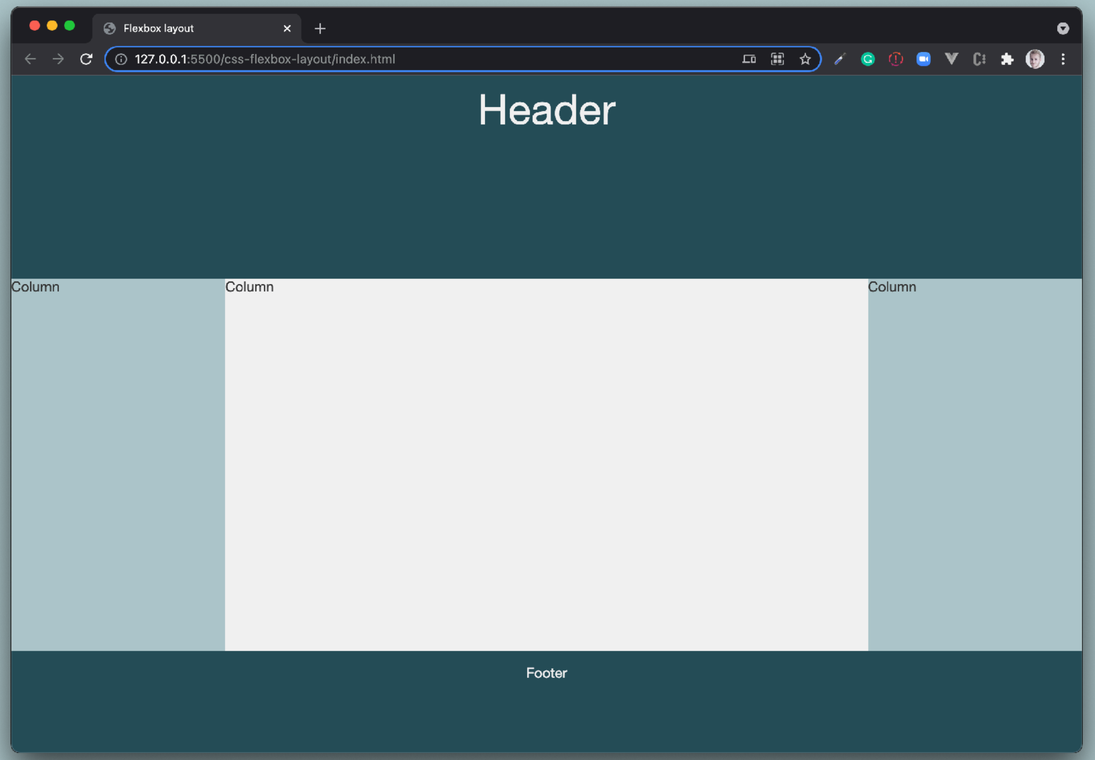
https://css-tricks.com/snippets/css/a-guide-to-flexbox/



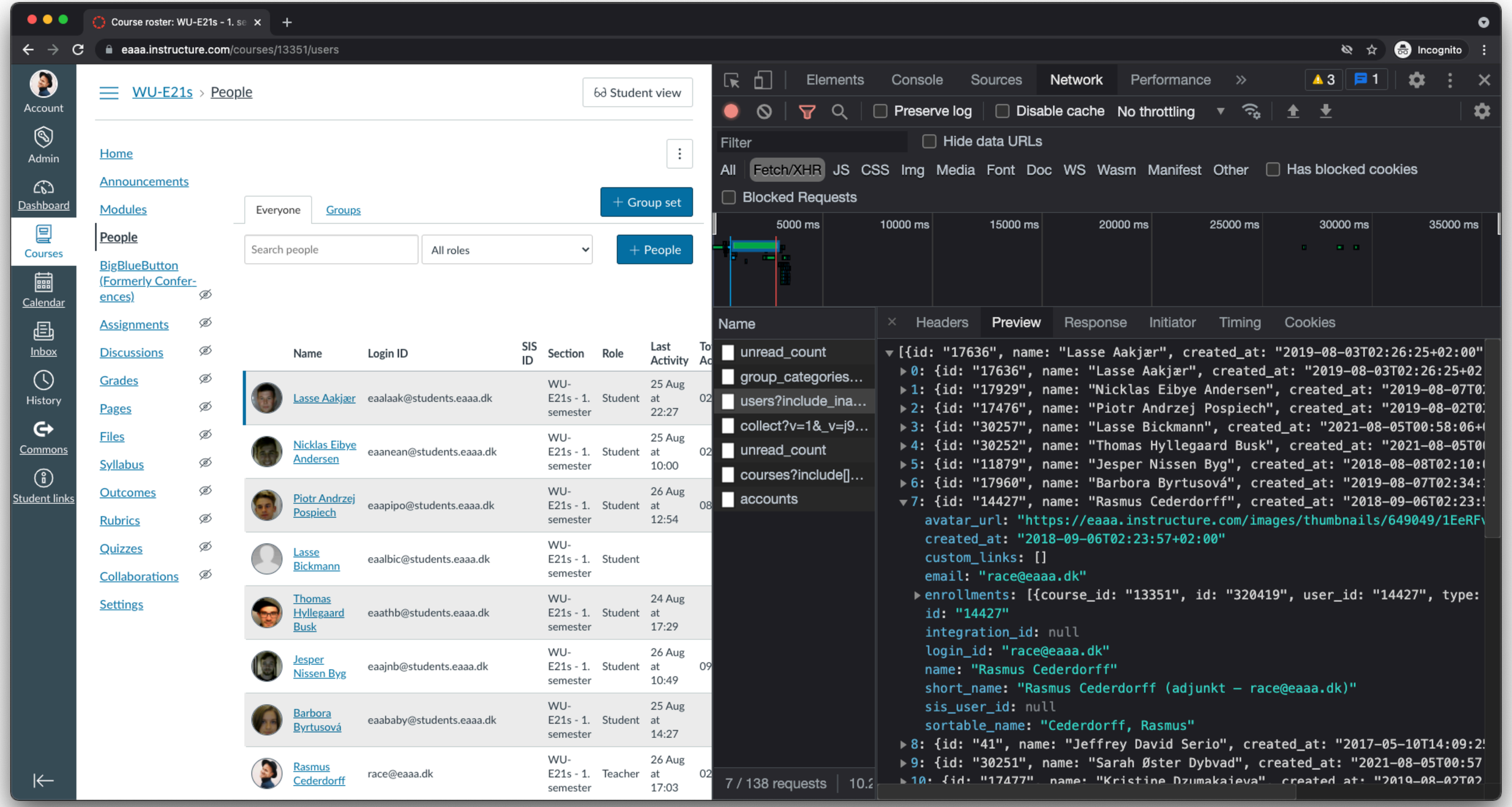
https://www.w3schools.com/css/css_templates.asp



css-grid-layout



css-flexbox-layout


Course roster: WU-E21s - 1. se
eaaa.instructure.com/courses/13351/users
Incognito
Account
Admin
Dashboard
Courses
Calendar
Inbox
History
Commons
Student links
WU-E21s › People
Student view
Home
Announcements
Modules
People
BigBlueButton (Formerly Confer-ences)
Assignments
Discussions
Grades
Pages
Files
Syllabus
Outcomes
Rubrics
Quizzes
Collaborations
Settings
Everyone
Groups
+ Group set
Search people
All roles
+ People
Name
Login ID
SIS ID
Section
Role
Last Activity
Lasse Aakjær
eaalaak@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 22:27
Nicklas Eibye Andersen
eaanean@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 10:00
Piotr Andrzej Pospiech
eaapipo@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 12:54
Lasse Bickmann
eaalbic@students.eaaa.dk
WU-E21s - 1. semester
Student
Thomas Hyllegaard Busk
eaathb@students.eaaa.dk
WU-E21s - 1. semester
Student
24 Aug at 17:29
Jesper Nissen Byg
eaajnb@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 10:49
Barbora Byrtusová
eaababy@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 14:27
Rasmus Cederdorff
race@eaaa.dk
WU-E21s - 1. semester
Teacher
26 Aug at 17:03
Elements
Console
Sources
Network
Performance
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
Name
unread_count
group_categories...
users?include_ina...
collect?v=1&_v=j9...
unread_count
courses?include[]...
accounts
7 / 138 requests
Headers
Preview
Response
Initiator
Timing
Cookies
[{id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02:00"
0: {id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02
1: {id: "17929", name: "Nicklas Eibye Andersen", created_at: "2019-08-07T0
2: {id: "17476", name: "Piotr Andrzej Pospiech", created_at: "2019-08-02T0
3: {id: "30257", name: "Lasse Bickmann", created_at: "2021-08-05T00:58:06+
4: {id: "30252", name: "Thomas Hyllegaard Busk", created_at: "2021-08-05T0
5: {id: "11879", name: "Jesper Nissen Byg", created_at: "2018-08-08T02:10:
6: {id: "17960", name: "Barbora Byrtusová", created_at: "2019-08-07T02:34:
7: {id: "14427", name: "Rasmus Cederdorff", created_at: "2018-09-06T02:23:
avatar_url: "https://eaaa.instructure.com/images/thumbnails/649049/1EeRF
created_at: "2018-09-06T02:23:57+02:00"
custom_links: []
email: "race@eaaa.dk"
enrollments: [{course_id: "13351", id: "320419", user_id: "14427", type:
id: "14427"
integration_id: null
login_id: "race@eaaa.dk"
name: "Rasmus Cederdorff"
short_name: "Rasmus Cederdorff (adjunkt – race@eaaa.dk)"
sis_user_id: null
sortable_name: "Cederdorff, Rasmus"
8: {id: "41", name: "Jeffrey David Serio", created_at: "2017-05-10T14:09:2
9: {id: "30251", name: "Sarah Øster Dybvad", created_at: "2021-08-05T00:57

GRID LAYOUT

BACK

1. Create a new project folder in your own local dev folder.
2. Inside the project folder, create a HTML file (`index.html`) and define the basic structure of an HTML page with `html`, `head` & `body`. Get a basic HTML5 Template here.
3. Add a `section` tag as your grid or flex box container.
4. Add some `article` tags with content inside of your `section`. The content could be user data (name, email, image, etc. Inspiration: https://cederdorff.github.io/web-frontend/canvas-users/data.json)
5. Add and link to an external style sheet named `main.css`
6. Make use of a CSS grid or flexbox to create a layout to display users. Inspiration: CSS Grid Layout Module & CSS Flexbox

Learn CSS Grid for free: https://scrimba.com/learn/cssgrid
Learn Flexbox for free: https://scrimba.com/learn/flexbox

PAGE LAYOUT

BACK

1. Create a new project folder in your own local dev folder.
2. Inside the project folder, create a HTML file (`index.html`) and define the basic structure of an HTML page with `html`, `head` & `body`. Get a basic HTML5 Template here.
3. Add the following tags with content: `header`, `nav`, `main`, `section`, `article`, `aside`, `footer`, some `headings` & `paragrafs`.
4. Add and link to an external style sheet named `main.css`
5. Make use of a CSS grid or flexbox to create a full page layout with navbar, header, main and footer. Inspiration: CSS Grid Layout Module & CSS Layout Templates

Learn CSS Grid for free: https://scrimba.com/learn/cssgrid
Learn Flexbox for free: https://scrimba.com/learn/flexbox

CSS.LAYOUT.IO

LAYOUTS AND PATTERNS MADE WITH CSS

CSS TEMPLATES

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

CSS VARIABLES

`var()` - refer to a defined CSS variable.

CSS variables are declared inside the `:root` selector

CSS variables can be accessed through the entire CSS document.

Makes the code easier to read (more understandable)

Makes it much easier to change the color values

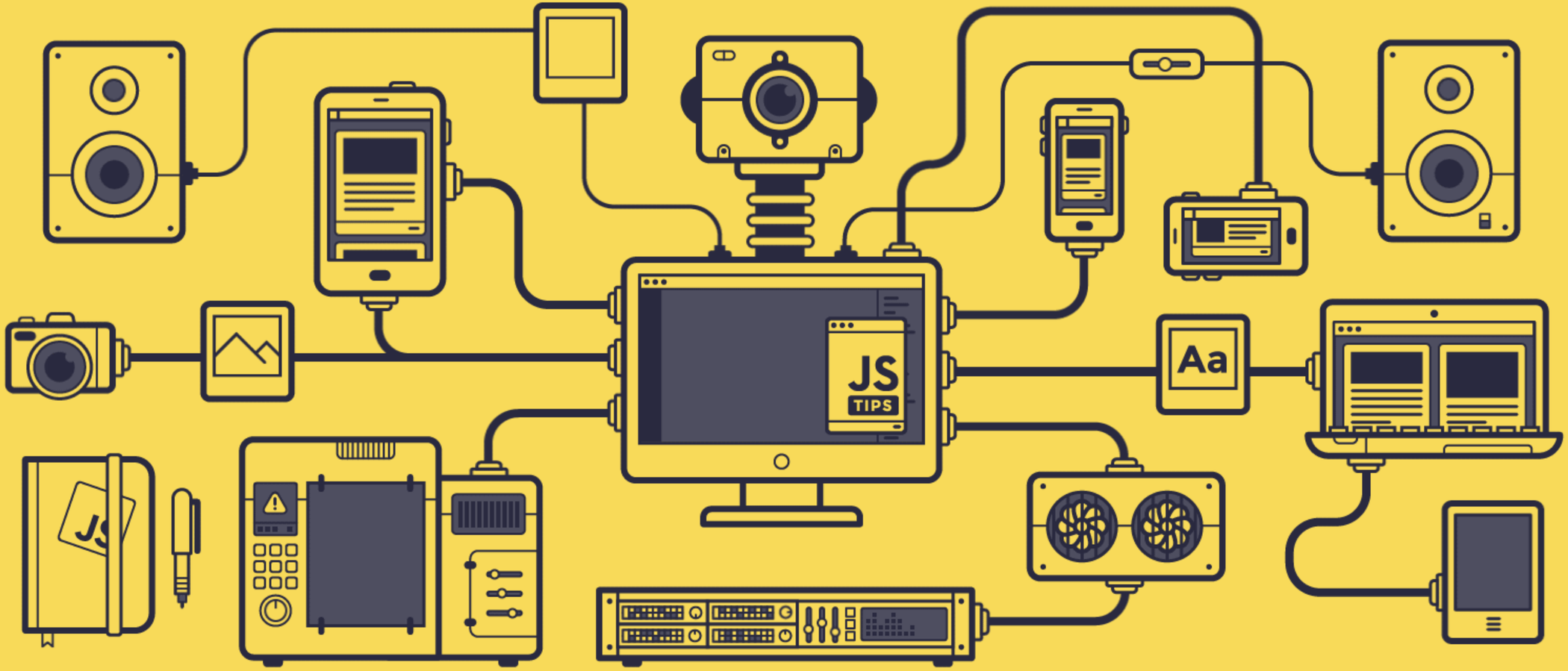
https://www.w3schools.com/css/css3_variables.asp

```css
/* ---------- root variables ---------- */
:root {
  --green: rgb(38, 76, 89);
  --green-opacity: rgba(38, 76, 89, 0.2);
  --light-green: rgb(172, 198, 201);
  --light-grey: #f1f1f4;
  --text-color-light: #f1f1f1;
  --text-color-dark: #333;
  --white: #fff;
  --font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
}

/* ---------- styling ---------- */
html,
body {
  color: var(--text-color-dark);
  font-family: var(--font-family);
  margin: 0;
  padding: 0;
  border: 0;
  outline: 0;
  background-color: var(--light-grey);
}

a {
  color: var(--green);
}

.grid > article {
  text-align: center;
  padding: 2em 1em;
  background-color: var(--light-green);
  transition: 0.3s;
}
```

GRID & CARD LAYOUT

BACK

1. Create a grid using a grid container and grid items. You can use the layout from the previous exercise and place the new grid and card layout inside of your `main`.
2. Make sure the grid adjusts to the size of the screen (adjust the number of columns with media queries).
3. Add static content to your grid items. The content must be based on Canvas user objects with values from the properties: https://cederdorff.github.io/web-frontend/canvas-users/data.json. Add at least 6 items (users).
4. Display the following for each user: `name`, `image`, `mail` & `enrollment_type`
5. Style each item (user) as a card. Inspiration: https://csslayout.io/card/, Cards, Profile Card & Flip Card
6. Add root variables at the top of your CSS script.
7. Make sure to use the variables through out your CSS script instead of keep typing the same color, font, sizes, etc.


JS
TIPS
Aa

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <button onclick="tryMe()">Try me</button>
    <script src="app.js"></script>
  </body>
</html>
```

app.js

```js
function tryMe() {
  document.body.style.backgroundColor = "red";
  document.body.style.color = "white";
  document.querySelector("h1").innerHTML = "My new header";
  alert("Hi, Welcome to my web app!");
}
```

Browser Tests

https://brg12.csb.app/

This is a Heading

This is a paragraph.

Try me

https://codesandbox.io/s/javascript-starter-brg12

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <button onclick="tryMe()">Try me</button>
    <script src="app.js"></script>
  </body>
</html>
```

app.js

```js
function tryMe() {
  document.body.style.backgroundColor = "red";
  document.body.style.color = "white";
}
```

WHAT IS JAVASCRIPT?

.. is the world's most popular programming language.

... is the programming language of the Web.

... is easy to learn.

... can change content of a webpage (HTML content).

... can change styling of HTML.

https://www.w3schools.com/js/default.asp

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <input type="text" placeholder="Type your name" id="name" />
    <button onclick="tryMe()">Try me</button>

    <script src="app.js"></script>
  </body>
</html>
```

app.js

```js
function tryMe() {
  let name = document.getElementById("name").value;
  alert("Hi " + name + ", Welcome to my web app!");
}
```

Browser Tests

https://brg12.csb.app/

This is a Heading

This is a paragraph.

Peter Try me

https://codesandbox.io/s/javascript-input-okm7e

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <input type="text" placeholder="Type your name" id="name" />
    <button onclick="tryMe()">Try me</button>
    <section id="content"></section>
    <script src="app.js"></script>
  </body>
</html>
```

app.js

```js
function tryMe() {
  let name = document.getElementById("name").value;
  document.getElementById("content").innerHTML += name;
}
```

Browser Tests

https://brg12.csb.app/

This is a Heading

This is a paragraph.

Type your name Try me

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
    <button onclick="tryMe()">Try me</button>
    <script src="app.js"></script>
  </body>
</html>
```

app.js

```js
function tryMe() {
  document.body.style.backgroundColor = "red";
  document.body.style.color = "white";
}
```

WITH JAVASCRIPT WE ARE ABLE TO

... build dynamic web pages and web apps.

... fetch content/ data from a backend (web service, data source, etc.) through an API.

... do DOM-manipulation.

... build and develop anything 🤷

https://www.w3schools.com/js/default.asp

DOM MANIPULATION

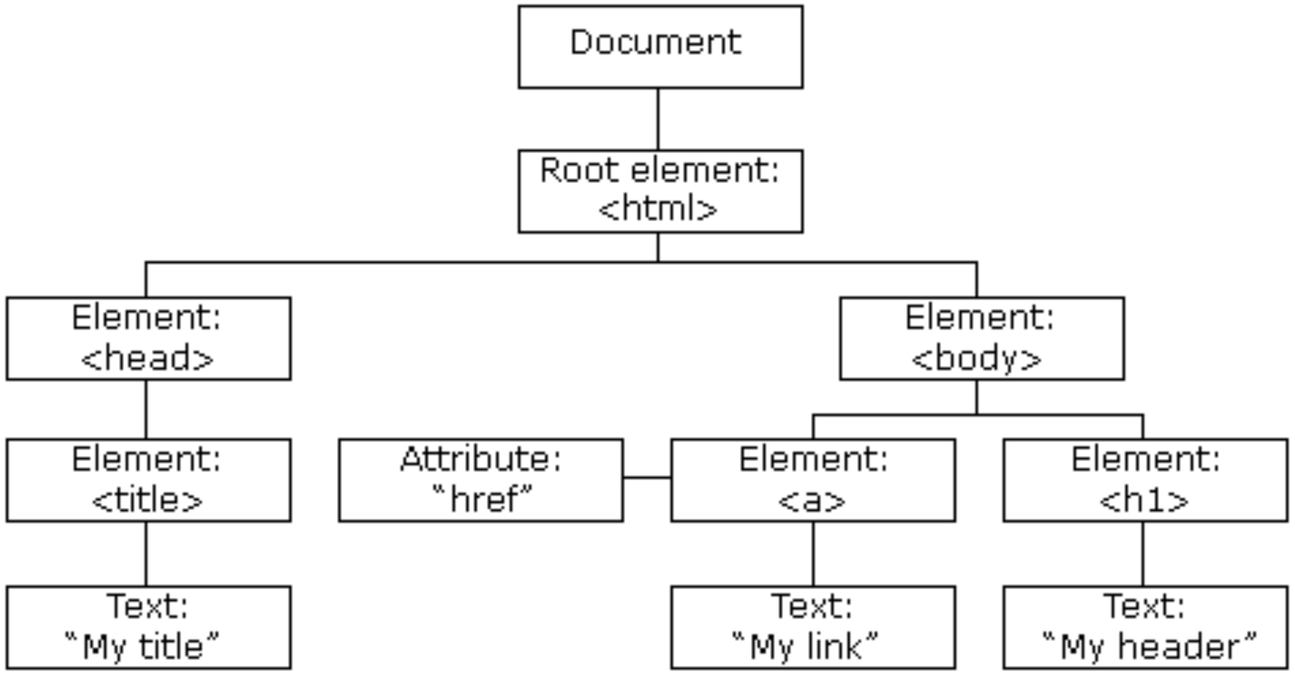
```
// declaring a variable with a value
let message = "Hi Frontenders!"

//accessing the variable and logging it to the console
console.log(message);

// appending the variable (the string) to the DOM element #content
document.querySelector("#content").innerHTML = message;
```

```
<body>
  <header>
    <h1>PROJECT TEMPLATE</h1>
  </header>
  <section id="content"></section>

  <!-- main js file -->
  <script src="js/main.js"></script>
</body>
```

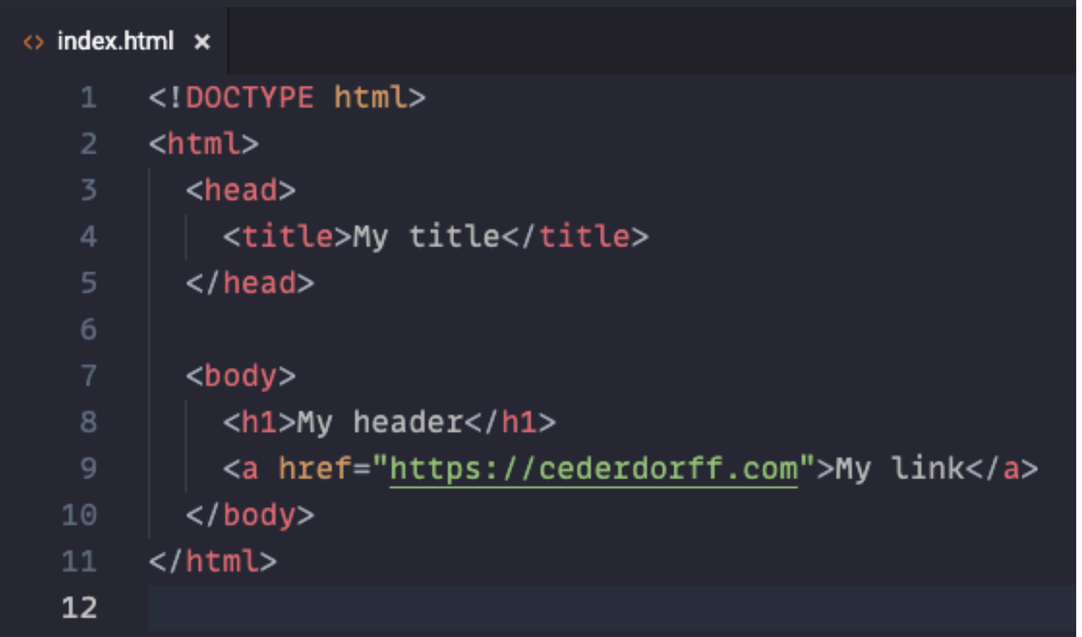
JAVASCRIPT HTML DOM

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My title</title>
  </head>

  <body>
    <h1>My header</h1>
    <a href="https://cederdorff.com">My link</a>
  </body>
</html>
```



https://www.w3schools.com/js/js_htmldom.asp
https://javascript.info/dom-nodes
https://javascript.info/dom-navigation



JAVASCRIPT HTML DOM

Document Object Model

The HTML document as an object

Gives us the power to create dynamic HTML and manipulate with the HTML (the DOM).

JavaScript can:

… change all the HTML elements in the page
… change all the HTML attributes in the page
… change all the CSS styles in the page
… remove existing HTML elements and attributes
… add new HTML elements and attributes
… react to all existing HTML events in the page
… create new HTML events in the page

https://www.w3schools.com/js/js_htmldom.asp
https://javascript.info/dom-nodes
https://javascript.info/dom-navigation

SEARCHING THE DOM: getElement* & querySelector*

```
<section id="elem">
    <article id="elem-content">Element</article>
</section>

<script>
    // get the element
    const element = document.getElementById('elem');
    // make its background red
    element.style.background = 'red';
    // get the elementContent
    const elementContent = document.querySelector('#elem-content');
    // change inner HTML
    elementContent.innerHTML = "<h2>Hi Web Developers!</h2>"
</script>
```

https://javascript.info/searching-elements-dom

SEARCHING THE DOM: getElementsBy*

```
<section id="elem">
    <article class="elem-content">Element</article>
    <article class="elem-content">Element</article>
    <article class="elem-content">Element</article>
</section>

<script>
    // get all elements matching the selector - returns an array
    const elements = document.getElementsByClassName('elem-content');
    // loop through all elements
    for (const element of elements) {
        element.innerHTML = "<h2>Hi Web Developers!</h2>";
    }
</script>
```

https://javascript.info/searching-elements-dom

SEARCHING THE DOM: querySelectorAll

```
<section id="elem">
    <article class="elem-content">Element</article>
    <article class="elem-content">Element</article>
    <article class="elem-content">Element</article>
</section>

<script>
    // get all elements matching the selector - returns an array
    const elements = document.querySelectorAll('.elem-content');
    // loop through all elements
    for (const element of elements) {
        element.innerHTML = "<h2>Hi Web Developers!</h2>";
    }
</script>
```

https://javascript.info/searching-elements-dom

JS HTML DOM

`getElement*` or `querySelector*`?

JS HTML DOM #1

BACK

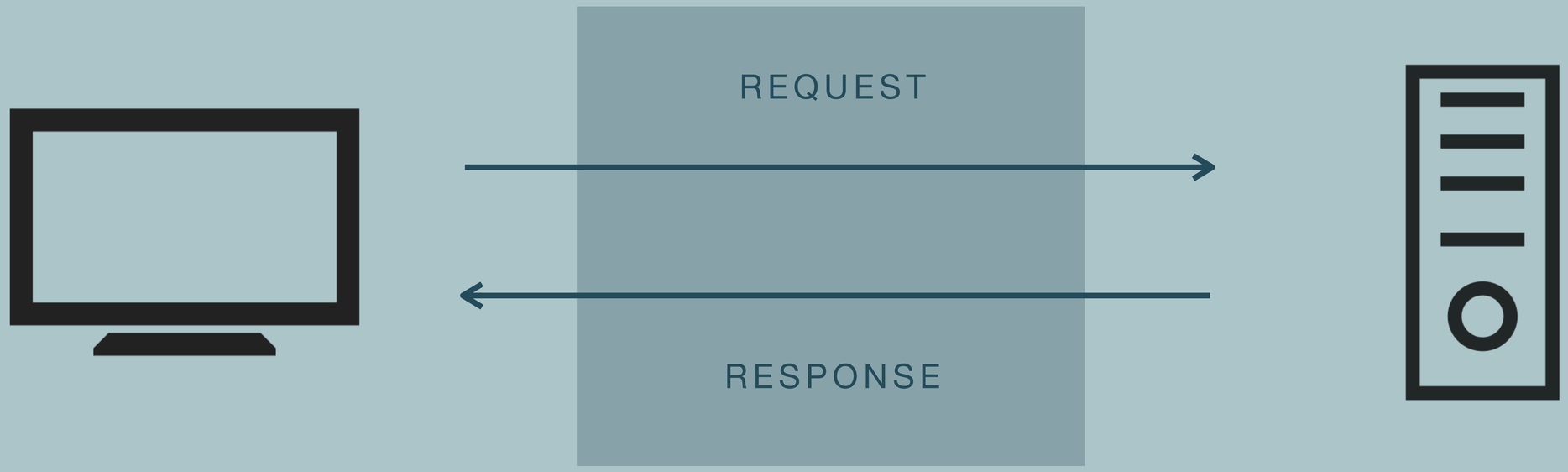
1. Use your previous codebase.
2. Create and link to an external JavaScript script called `app.js`
3. Consider where to place the link to your external JavaScript script.
4. Make sure the script is connected by creating a `console.log(…)` at the top of the script, logging “`app.js is ready` 🎉”. Test in browser and developer tool.
5. Create a function called `changeBackgroundColor()` changing the background color of the `body`. Use `document.querySelector(…)` to refer to `body` in the DOM.
6. Create a `button` in your html file and add a `onclick` event calling `changeBackgroundColor()`. You can use the `onclick` attribute on the button tag or add an `eventlistener` in your JS. Test the button and functionality.
7. Extra: Customise `changeBackgroundColor()` or create a new function called `turnOnDarkMode()`. The function must change the styling of your page to dark mode.

JS HTML DOM #2

BACK

1. Define an input field and a new button in your HTML file.
2. Declare a new function called `alertName()` in your JS file.
3. Add an `onclick` event to the button executing `alertName()`.
4. `alertName()` should grab the value of the input field and `alert(…)` the value.
5. Test in browser.

CLIENT SERVER MODEL

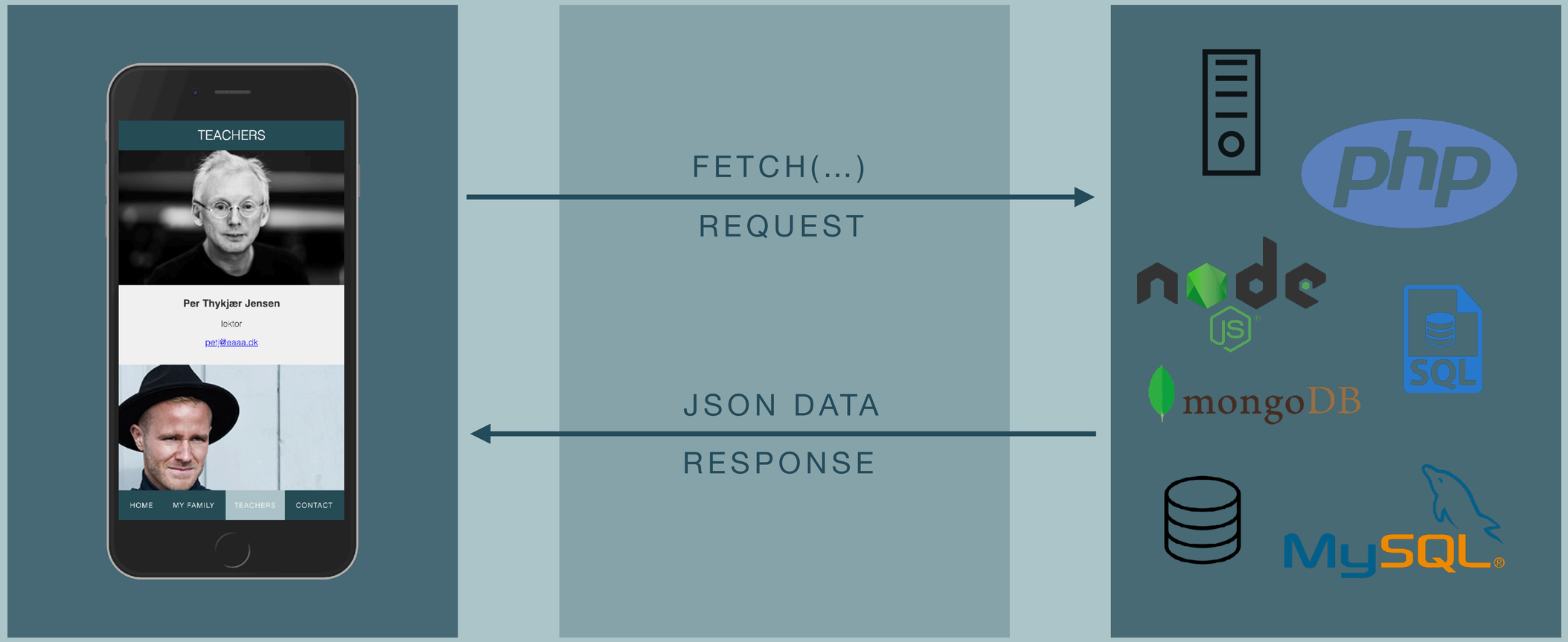


FETCH DATA

FRONTEND

API

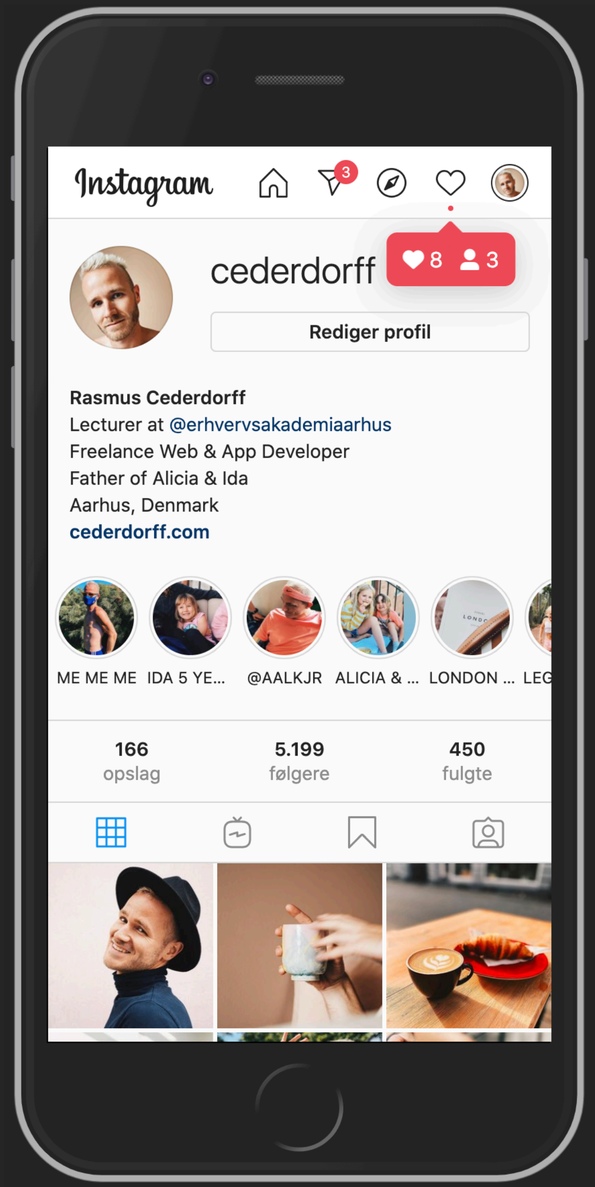
BACKEND



WEB APP

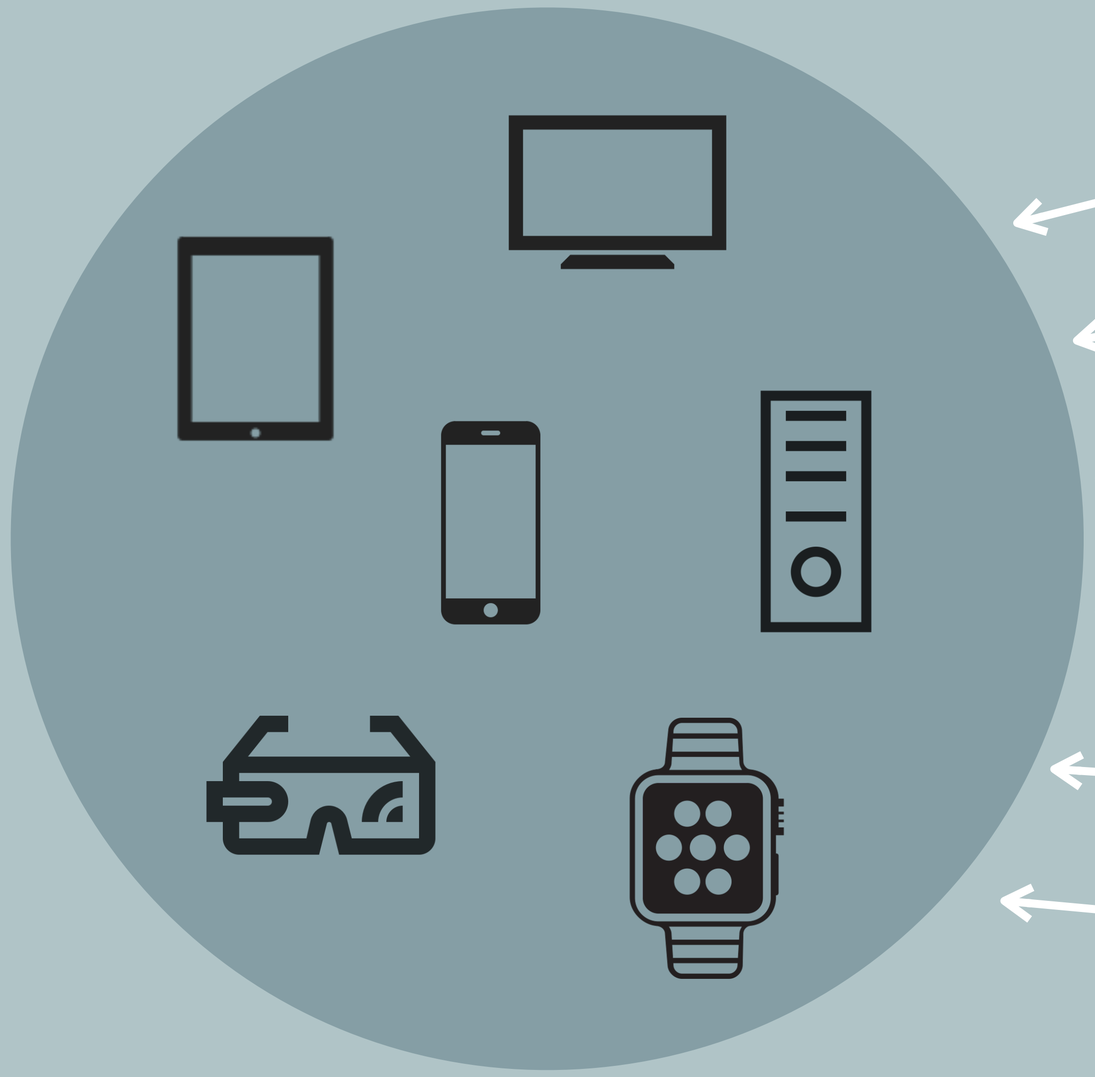
REQUEST

RESPONSE



API

WEB APP



DATA SOURCES

API — WIKIPEDIA

API — GOOGLE MAPS

API — CVR API

API — INSTAGRAM

API — WEATHER API

API — HEADLESS CMS

JAVASCRIPT CONCEPTS

FRONTEND DEVELOPMENT

ES6+

MODERN JAVASCRIPT

MODERN JAVASCRIPT

LET & CONST
TEMPLATE STRING
ARROW FUNCTIONS
FETCH
PROMISES
ASYNC & AWAIT
FOR OF LOOP
ARRAY.FIND()
ARRAY.MAP()
ARRAY.REDUCE()
ARRAY.FILTER()
ARRAY.SORT()
ARRAY.CONCAT()
DEFAULT PARAMS
DESTRUCTURING OBJECTS
DESTRUCTURING ARRAYS
OBJECT LITERAL
SPREAD OPERATOR
CLASSES
MODULES
IMPORT & EXPORT

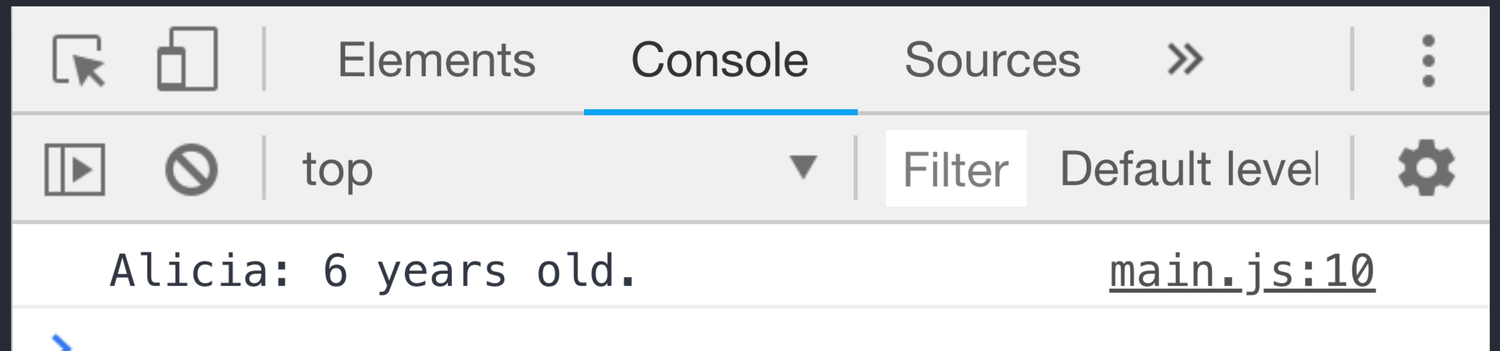
VARIABLES

... ARE USED TO STORE DATA IN THE MEMORY

VARIABLES

... STORE DATA IN THE MEMORY

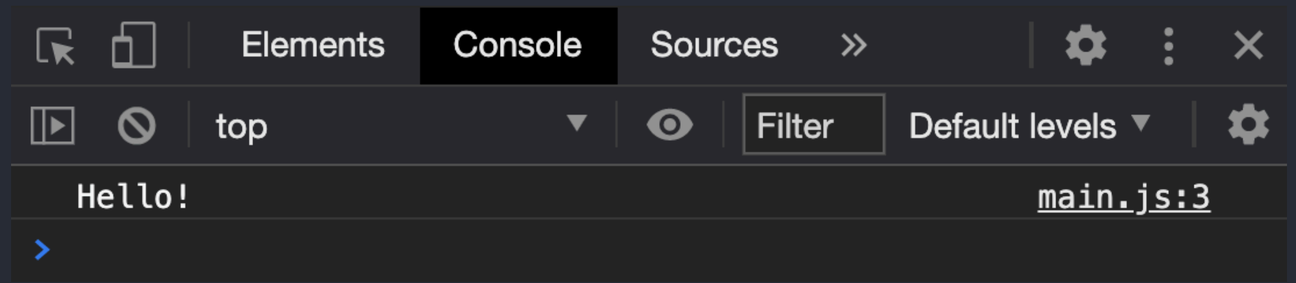
```
let name = "Alicia";
let age = 6;

console.log(name + ": " + age + " years old.");
```



VARIABLES

... STORE DATA IN THE MEMORY

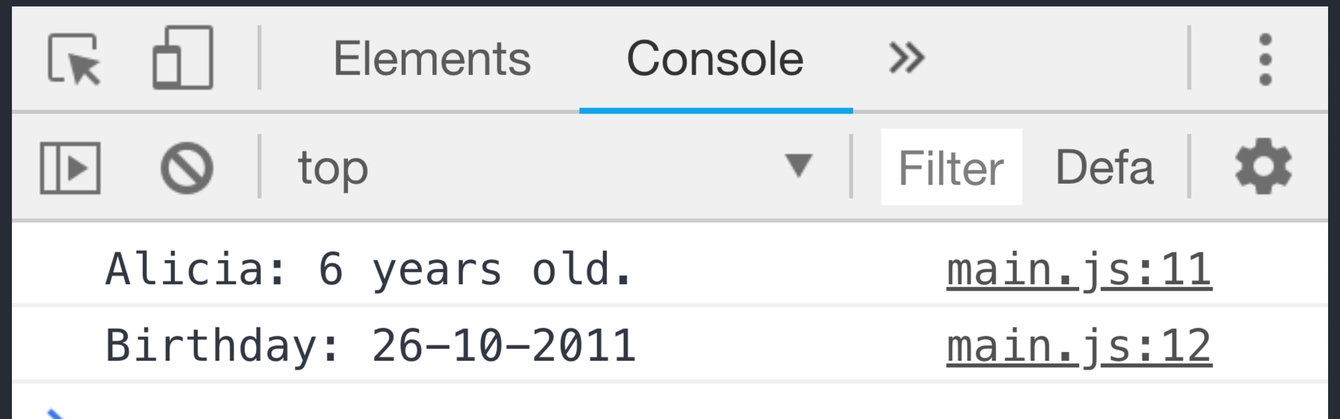
```
let message = "Hello!";

console.log(message);
```



VARIABLES

... STORE DATA IN THE MEMORY

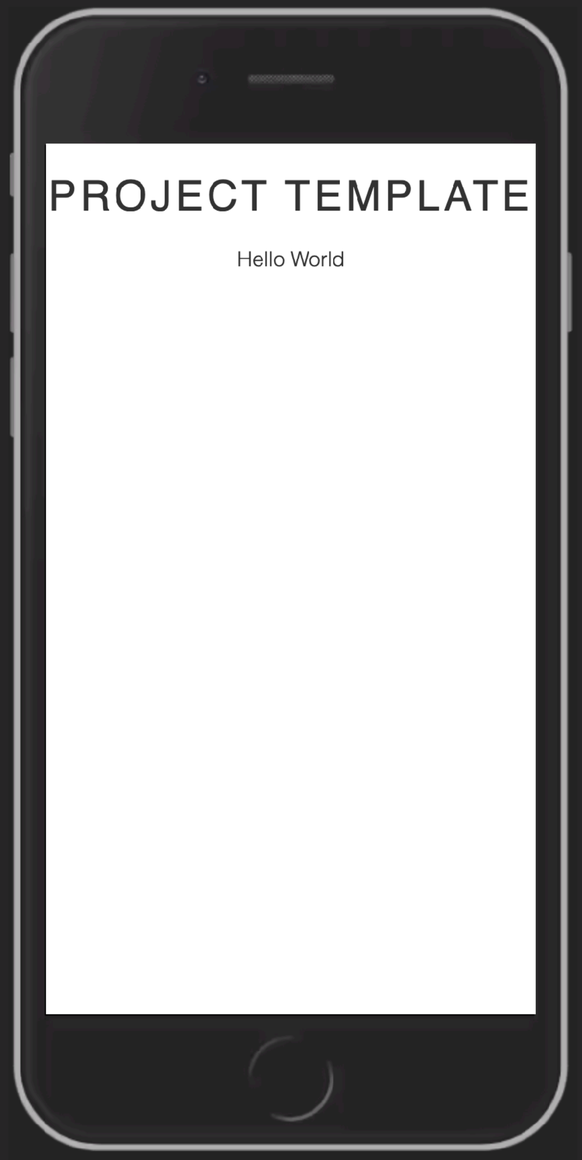
```
let name = "Alicia";
let age = 6;
const birthday = '26-10-2011';

console.log(name + ": " + age + " years old.");
console.log("Birthday: " + birthday);
```

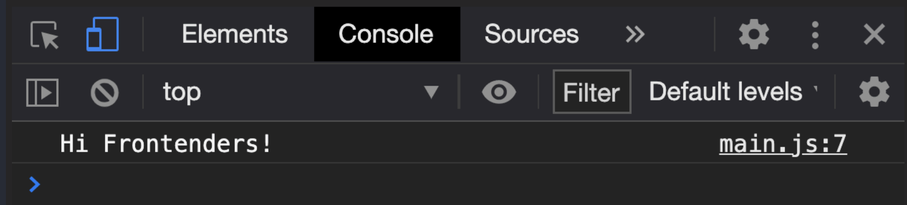


```js
// declaring a variable with a value
let message = "Hi Frontenders!"

//accessing the variable and logging it to the console
console.log(message);

// appending the variable (the string) to the DOM element #content
document.querySelector("#content").innerHTML = message;
```

```html
<body>
  <header>
    <h1>PROJECT TEMPLATE</h1>
  </header>
  <section id="content"></section>

  <!-- main js file -->
  <script src="js/main.js"></script>
</body>
```

```
// declaring a variable with a value
let message = "Hi Frontenders!"

//accessing the variable and logging it to the console
console.log(message);

// changing the value of the variable
message = "Hello World";

// appending the variable (the string) to the DOM element #content
document.querySelector("#content").innerHTML = message;
```





VAR VS LET

THE DIFFERENCE IS THE SCOPING

VAR IS FUNCTION-WIDE OR GLOBAL SCOPE

LET IS BLOCK SCOPED

VAR TOLERATES REDECLARATION

https://javascript.info/variables
https://javascript.info/var

```
// Example 1
// “var” has no block scope
if (true) {
  var test1 = true; // use "var" instead of "let"
}
console.log(test1); // true, the variable lives after if

// Example 2
if (true) {
  let test2 = true; // use "let"
}
console.log(test2); // Error: test is not defined

// Example 3
for (var i = 0; i < 10; i++) {
  // ...
}
console.log(i); // 10, "i" is visible after loop, it's a global variable
```

var-vs-let

```
// “var” tolerates redeclarations
var user1 = "Pete";
var user1 = "John"; // this "var" does nothing (already declared)
// ...it doesn't trigger an error
console.log(user1); // John

let user2;
let user2; // SyntaxError: 'user' has already been declared
```

var-vs-let

PROGRAMMING KNOWLEDGE

IT'S ALL OBJECTS & ARRAYS

OBJECTS

... A SET OF NAMED VALUES

OBJECTS

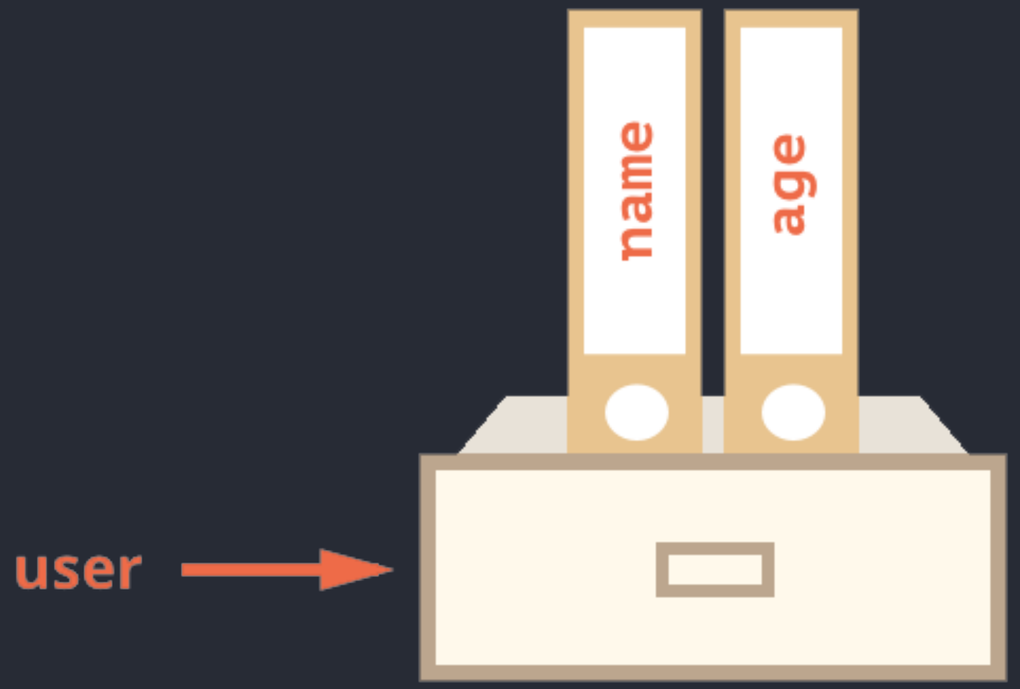
Objects are used to store keyed collections of various data

Containers for named values called properties. A property is a “`key: value`” pair

OBJECTS

A SET OF NAMED VALUES

```
let user = {
  name: 'Alicia',
  age: 6
};

console.log(user.name +
  " is " + user.age +
  " years old.");
```




```
Alicia is 6 years old.          main.js:11
>
```

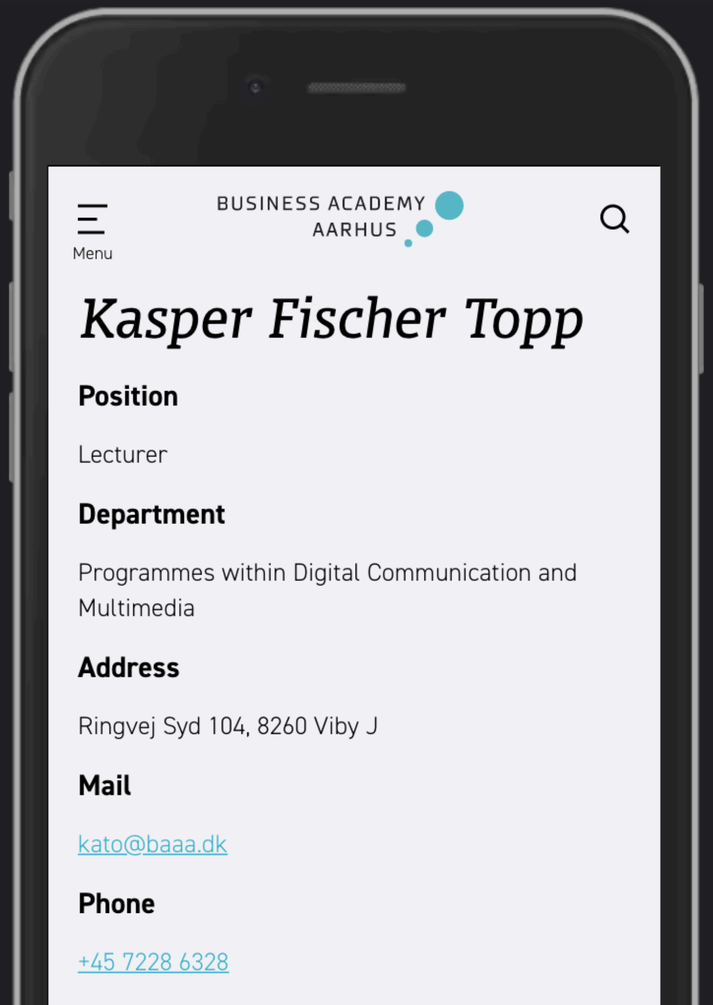
OBJECTS

A SET OF NAMED VALUES

property

value

```
const mrBackend = {
  name: "Kasper Fischer Topp",
  mail: "kato@eaaa.dk",
  phone: "72286328",
  position: "Lecturer",
  favTechnologies: ["PHP", "SQL"]
};
```



OBJECTS

property

value

```
let person = {
  name: "Birgitte Kirk Iversen",
  initials: "bki",
  mail: "bki@baaa.dk",
  phone: "72286316",
  address: "Sønderhøj 30, 8260 Viby J",
  position: "Senior Lecturer",
  department: "Multimedia Design"
};
```

Birgitte Kirk Iversen

Position

Senior Lecturer

Department

Programmes within Digital Communication and Multimedia

Address

Sønderhøj 30, 8260 Viby J

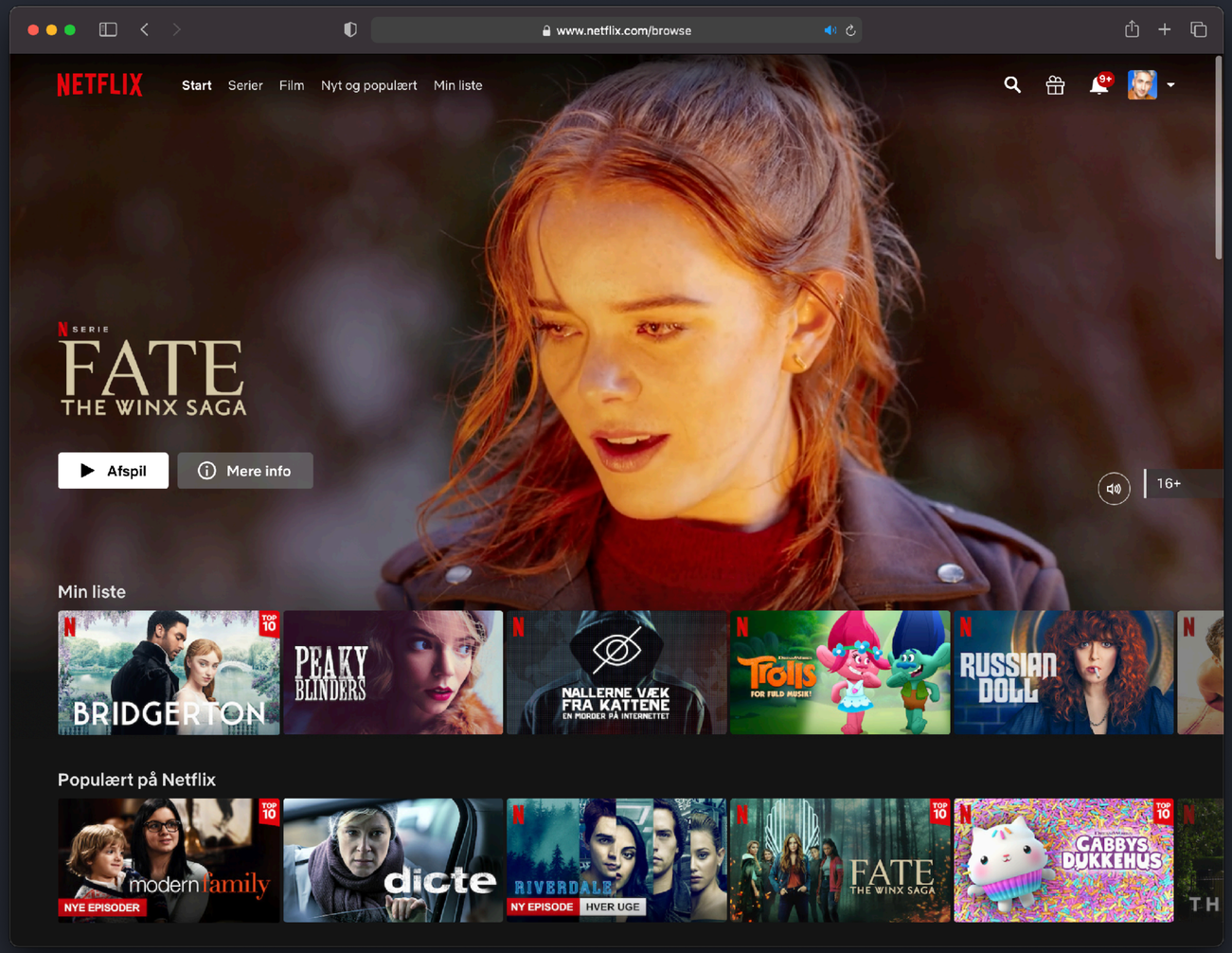
Mail

bki@baaa.dk

Phone

+45 7228 6316

https://www.baaa.dk/contact/find-employee/employee/birgitte-kirk-iversen



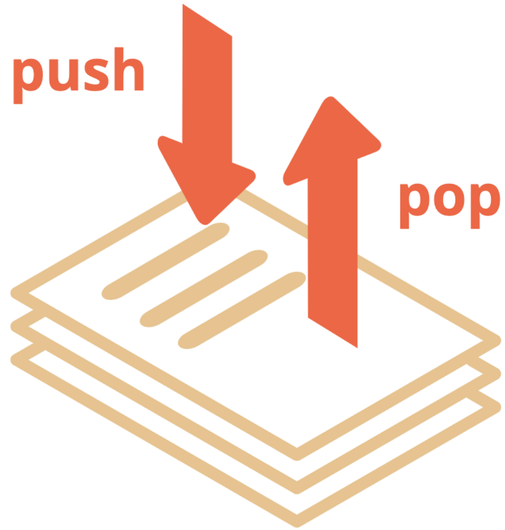

```
let movie = {
  title: "Frozen 2",
  description: "Elsa the Snow Queen has an extr
  trailer: "https://www.youtube.com/embed/bwzLi
  length: "1h 43m",
  year: "2019"
}
```

ARRAYS

... A COLLECTION OF VALUES OR OBJECTS

ARRAYS

Arrays are ordered collections



An array is a way to hold more than one value at a time we have a 1st, a 2nd, a 3rd, a 4th element and so on.

```
let todaysLecturers = [
  {
    name: "Kasper Fischer Topp",
    mail: "kato@eaaa.dk",
    phone: "72286328",
    position: "Lecturer",
    favTechnologies: ["PHP", "SQL"],
    nickname: "Mr. Backend"
  },
  {
    name: "Rasmus Cederdorff",
    mail: "race@eaaa.dk",
    phone: "72286318",
    position: "Lecturer",
    favTechnologies: ["JavaScript"],
    nickname: "Mr. Frontend"
  }
];
```

First element

Second element

ARRAYS

Rasmus Cederdorff

Position:
Lecturer

Department/
Multimedia D
Digital Concep

Address:
Ringvej Syd 1(

Mail:
race@baaa.dk

Phone:
7228 6318

Michael Hvidtfeldt

Position:
Lecturer

Department/
Multimedia D

Address:
Ringvej Syd 1(

Mail:
mhv@baaa.dk

Phone:
7228 6328

Birgitte Kirk Iversen

Position:
Senior Lecturer

Department/programme:
Multimedia Design

Address:
Sønderhøj 30, 8260 Viby J

Mail:
bki@baaa.dk

Phone:
7228 6316

```
let teachers = [{
    name: "Birgitte Kirk Iversen",
    mail: "bki@baaa.dk"
  },
  {
    name: "Michael Hvidtfeldt",
    mail: "mhv@baaa.dk"
  },
  {
    name: "Rasmus Cederdorff",
    mail: "race@baaa.dk"
  }
];
```

16.56

Instagram

drnyheder

224 likes

drnyheder 🥇 Lysten skal drive værket for vores børn 🥇
Ifølge den tidligere verdensmester og OL-sølvvinder... more

View all 2 comments

Add a comment...

1 HOUR AGO · SEE TRANSLATION

bmwdanmark
Sponsored

16.56

Instagram

bestmacsetups

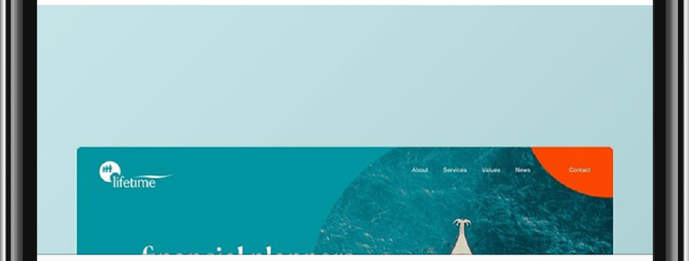
52 likes

bestmacsetups I love the Light/Airy feel of this setup! ☕
Would you work here?... more

View all 4 comments

30 MINUTES AGO

welovewebdesign



16.56

cederdorff

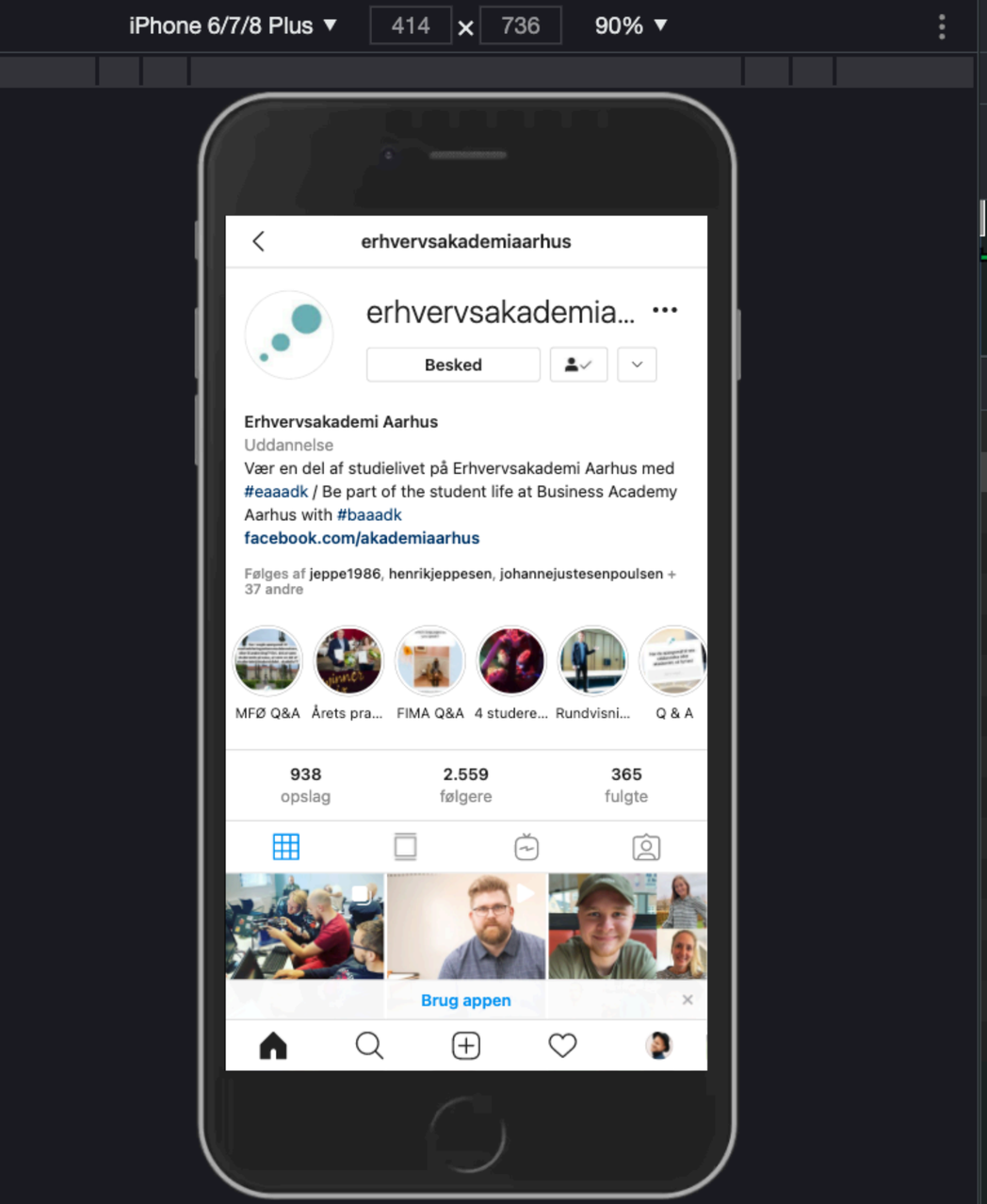
1287 profile visits in the last 7 days

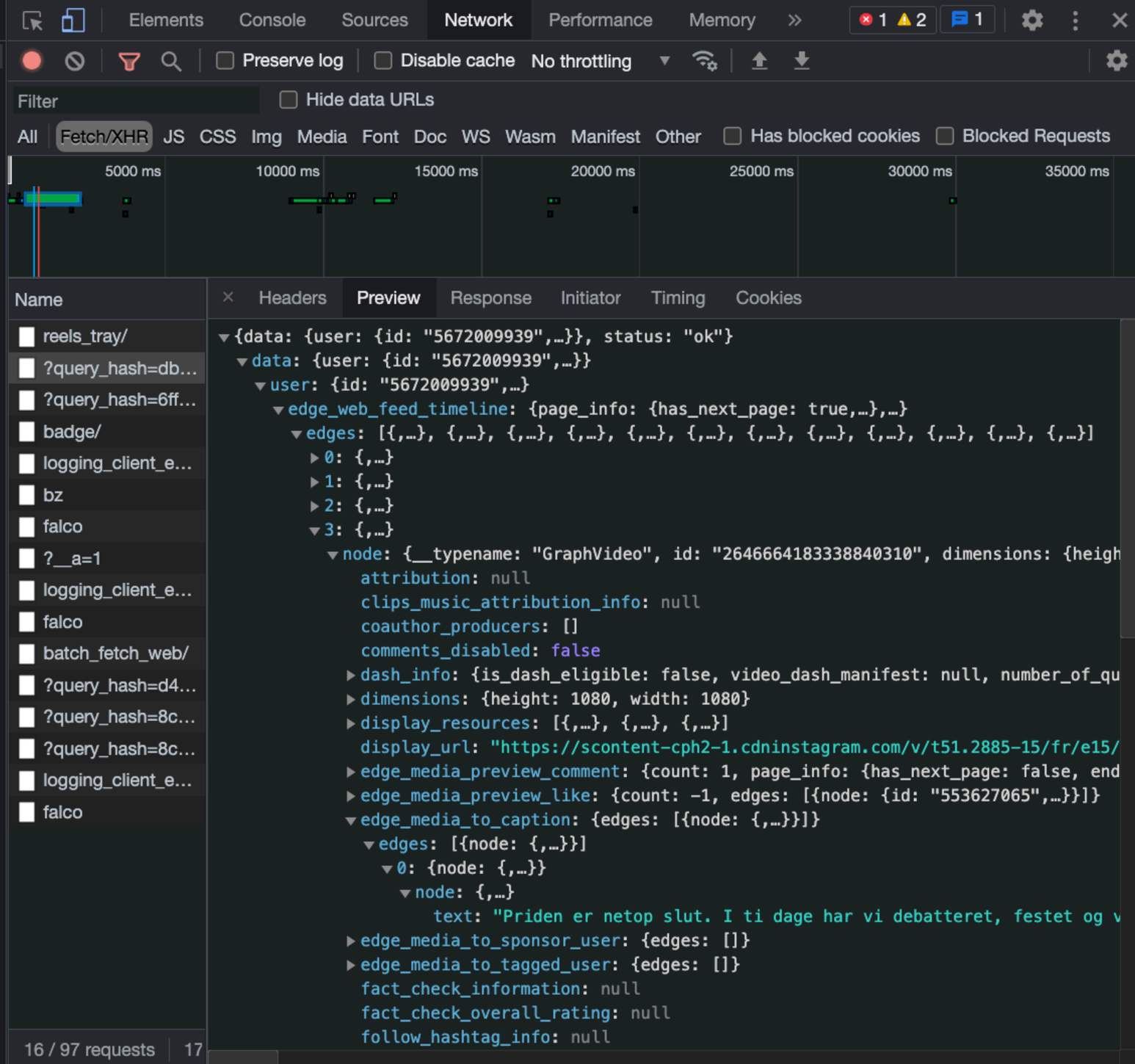
92 posts
3.967 followers
366 following

Promotions | Edit Profile

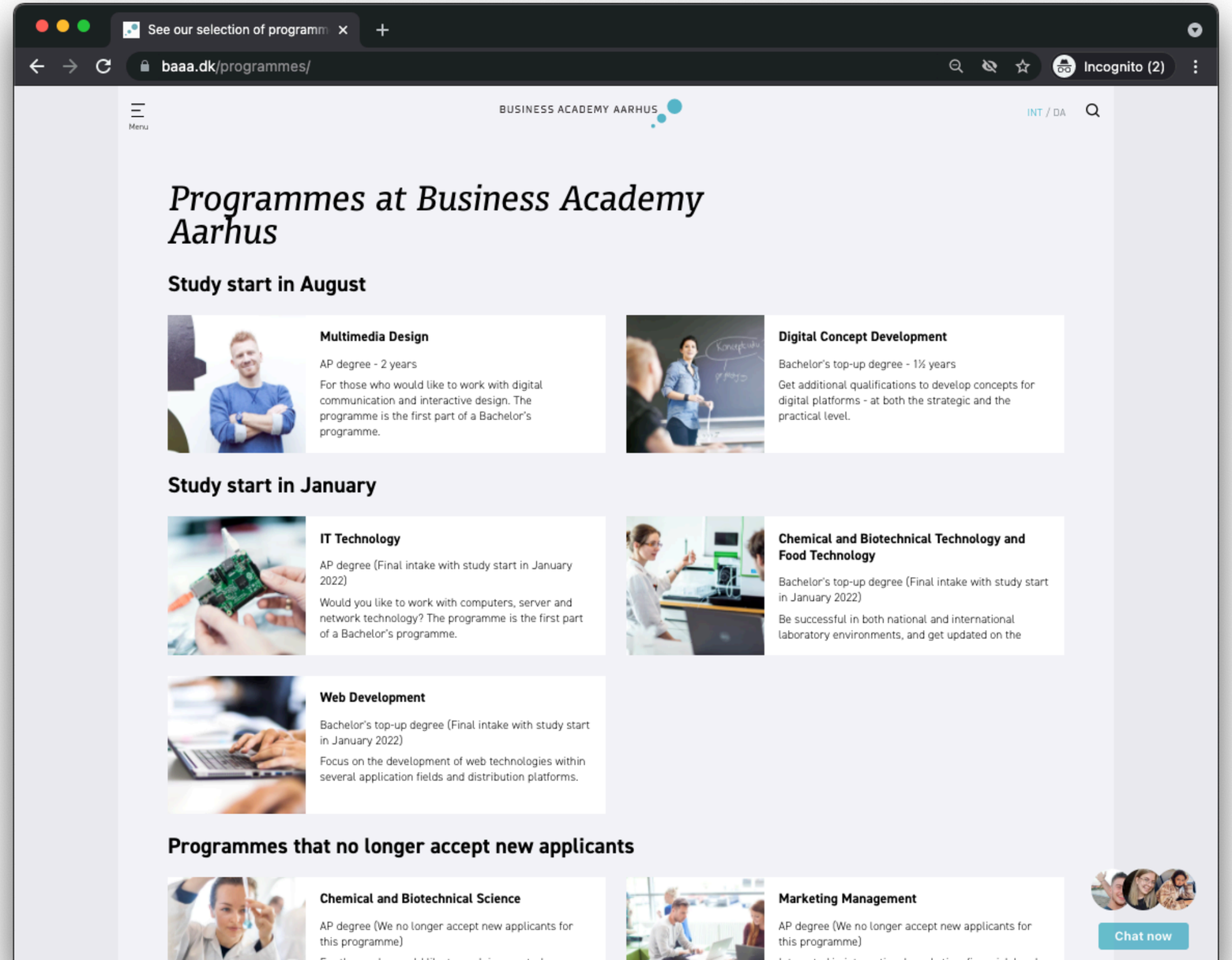
CEDERDORFF
Personal Blog
RASMUS CEDERDORFF
Freelance Web & App Developer
Lecturer at @erhvervsakademiaarhus
Father of Alicia & Ida
Aarhus, Denmark
cederdorff.com/

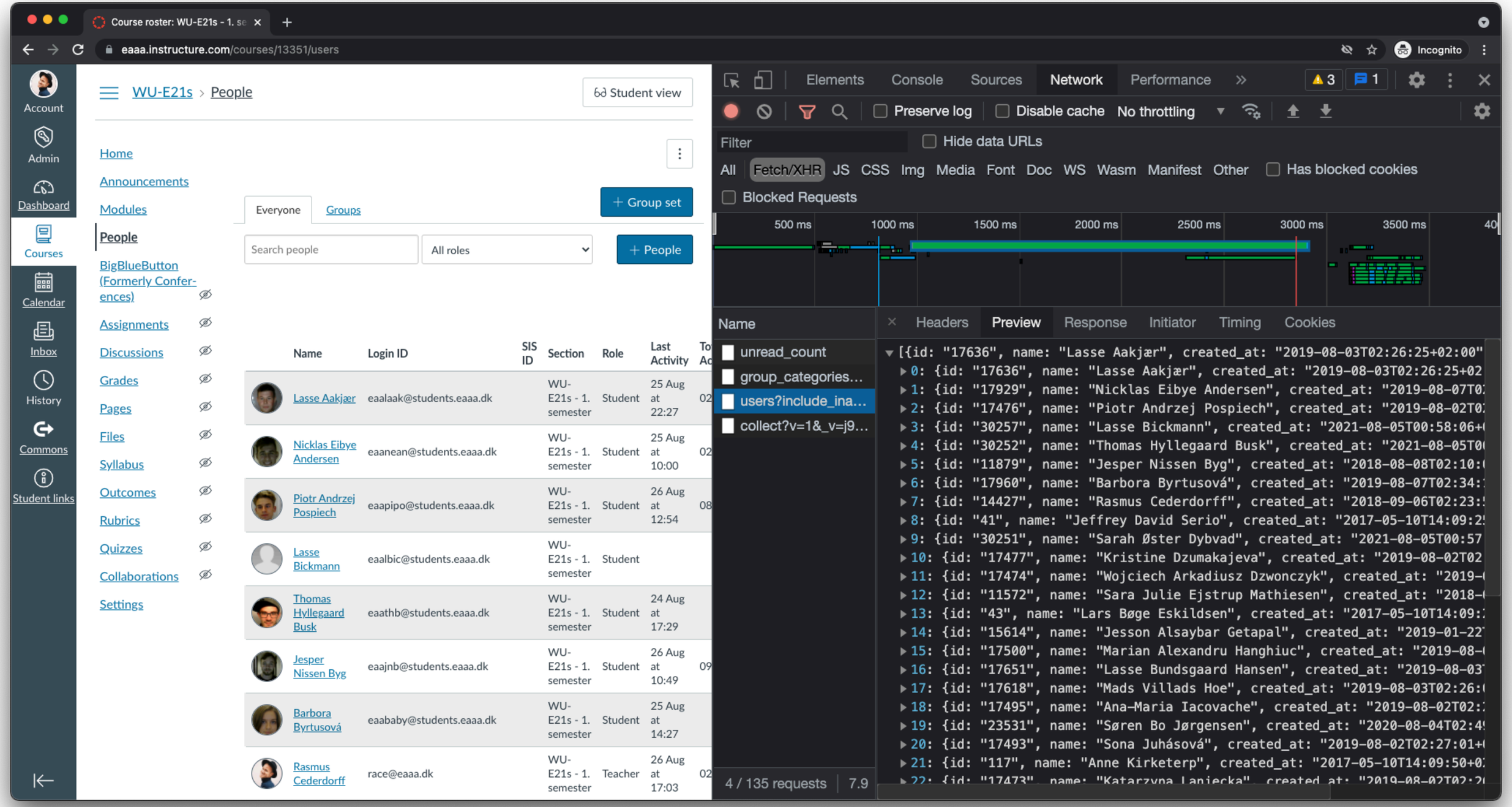
Email

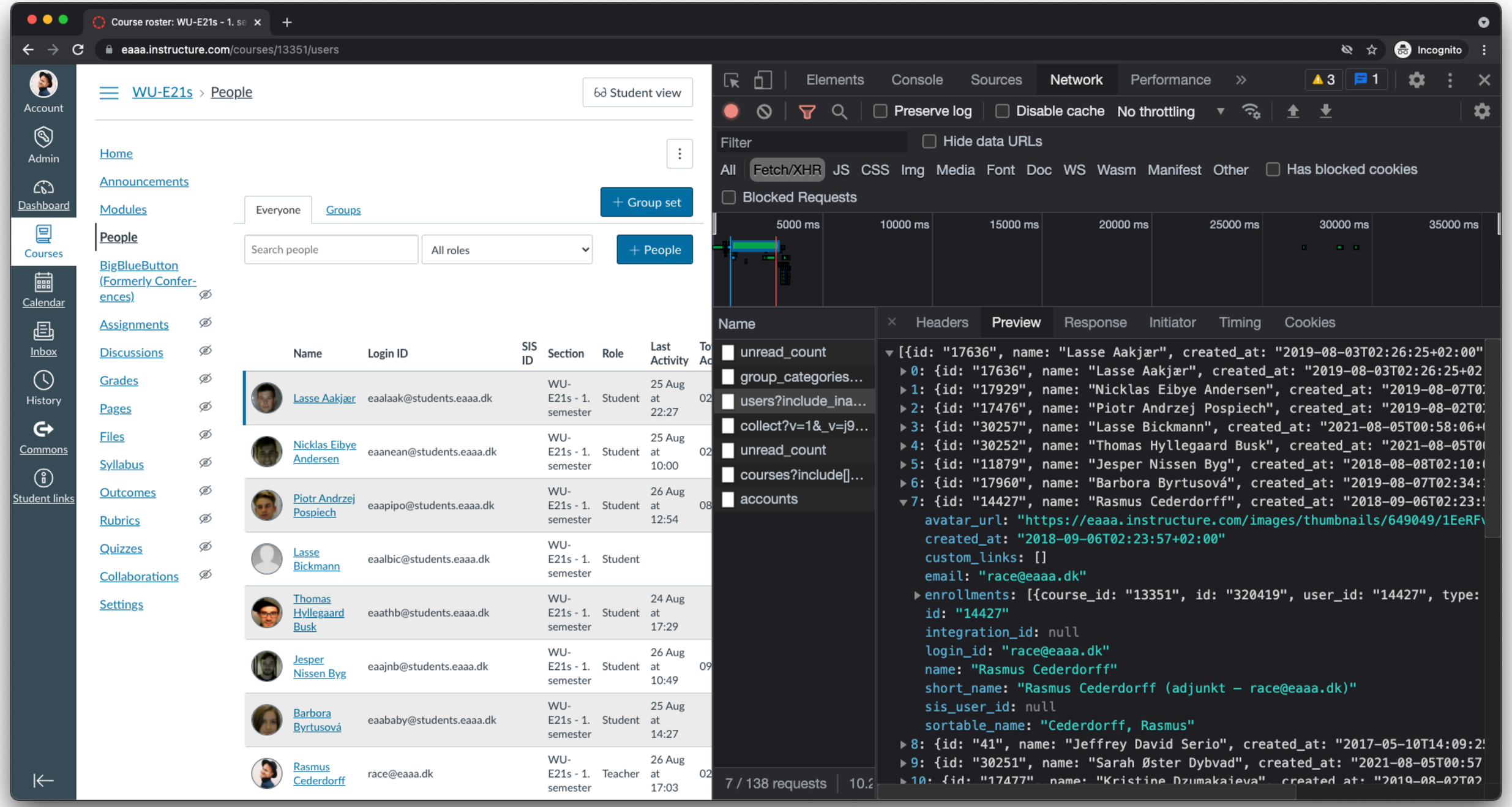

iPhone 6/7/8 Plus
414
736
90%
erhvervsakademiaarhus
erhvervsakademia...
Besked
Erhvervsakademi Aarhus
Uddannelse
Vær en del af studielivet på Erhvervsakademi Aarhus med #eaaadk / Be part of the student life at Business Academy Aarhus with #baaadk
facebook.com/akademiaarhus
Følges af jeppe1986, henrikjeppesen, johannejustesenpoulsen + 37 andre
MFØ Q&A
Årets pra...
FIMA Q&A
4 studere...
Rundvisni...
Q & A
938
opslag
2.559
følgere
365
fulgte
Brug appen


Elements
Console
Sources
Network
Performance
Memory
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
Name
reels_tray/
?query_hash=db...
?query_hash=6ff...
badge/
logging_client_e...
bz
falco
?__a=1
logging_client_e...
falco
batch_fetch_web/
?query_hash=d4...
?query_hash=8c...
?query_hash=8c...
logging_client_e...
falco
16 / 97 requests
17
Headers
Preview
Response
Initiator
Timing
Cookies
{data: {user: {id: "5672009939",…}}, status: "ok"}
data: {user: {id: "5672009939",…}}
user: {id: "5672009939",…}
edge_web_feed_timeline: {page_info: {has_next_page: true,…},…}
edges: [{,…}, {,…}, {,…}, {,…}, {,…}, {,…}, {,…}, {,…}, {,…}, {,…}, {,…}, {,…}]
0: {,…}
1: {,…}
2: {,…}
3: {,…}
node: {__typename: "GraphVideo", id: "2646664183338840310", dimensions: {heigh
attribution: null
clips_music_attribution_info: null
coauthor_producers: []
comments_disabled: false
dash_info: {is_dash_eligible: false, video_dash_manifest: null, number_of_qu
dimensions: {height: 1080, width: 1080}
display_resources: [{,…}, {,…}, {,…}]
display_url: "https://scontent-cph2-1.cdninstagram.com/v/t51.2885-15/fr/e15/
edge_media_preview_comment: {count: 1, page_info: {has_next_page: false, end
edge_media_preview_like: {count: -1, edges: [{node: {id: "553627065",…}}]}
edge_media_to_caption: {edges: [{node: {,…}}]}
edges: [{node: {,…}}]
0: {node: {,…}}
node: {,…}
text: "Priden er netop slut. I ti dage har vi debatteret, festet og v
edge_media_to_sponsor_user: {edges: []}
edge_media_to_tagged_user: {edges: []}
fact_check_information: null
fact_check_overall_rating: null
follow_hashtag_info: null

OBJECTS WITH PROPERTIES IN ARRAYS




Course roster: WU-E21s - 1. se
eaaa.instructure.com/courses/13351/users
Incognito
Account
Admin
Dashboard
Courses
Calendar
Inbox
History
Commons
Student links
WU-E21s › People
Student view
Home
Announcements
Modules
People
BigBlueButton (Formerly Confer-ences)
Assignments
Discussions
Grades
Pages
Files
Syllabus
Outcomes
Rubrics
Quizzes
Collaborations
Settings
Everyone
Groups
+ Group set
Search people
All roles
+ People
Name
Login ID
SIS ID
Section
Role
Last Activity
Lasse Aakjær
eaalaak@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 22:27
Nicklas Eibye Andersen
eaanean@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 10:00
Piotr Andrzej Pospiech
eaapipo@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 12:54
Lasse Bickmann
eaalbic@students.eaaa.dk
WU-E21s - 1. semester
Student
Thomas Hyllegaard Busk
eaathb@students.eaaa.dk
WU-E21s - 1. semester
Student
24 Aug at 17:29
Jesper Nissen Byg
eaajnb@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 10:49
Barbora Byrtusová
eaababy@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 14:27
Rasmus Cederdorff
race@eaaa.dk
WU-E21s - 1. semester
Teacher
26 Aug at 17:03
Elements
Console
Sources
Network
Performance
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
500 ms
1000 ms
1500 ms
2000 ms
2500 ms
3000 ms
3500 ms
Name
unread_count
group_categories...
users?include_ina...
collect?v=1&_v=j9...
Headers
Preview
Response
Initiator
Timing
Cookies
[{id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02:00"
0: {id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02
1: {id: "17929", name: "Nicklas Eibye Andersen", created_at: "2019-08-07T0
2: {id: "17476", name: "Piotr Andrzej Pospiech", created_at: "2019-08-02T0
3: {id: "30257", name: "Lasse Bickmann", created_at: "2021-08-05T00:58:06+
4: {id: "30252", name: "Thomas Hyllegaard Busk", created_at: "2021-08-05T0
5: {id: "11879", name: "Jesper Nissen Byg", created_at: "2018-08-08T02:10:
6: {id: "17960", name: "Barbora Byrtusová", created_at: "2019-08-07T02:34:
7: {id: "14427", name: "Rasmus Cederdorff", created_at: "2018-09-06T02:23:
8: {id: "41", name: "Jeffrey David Serio", created_at: "2017-05-10T14:09:2
9: {id: "30251", name: "Sarah Øster Dybvad", created_at: "2021-08-05T00:57
10: {id: "17477", name: "Kristine Dzumakajeva", created_at: "2019-08-02T02
11: {id: "17474", name: "Wojciech Arkadiusz Dzwonczyk", created_at: "2019-
12: {id: "11572", name: "Sara Julie Ejstrup Mathiesen", created_at: "2018-
13: {id: "43", name: "Lars Bøge Eskildsen", created_at: "2017-05-10T14:09:
14: {id: "15614", name: "Jesson Alsaybar Getapal", created_at: "2019-01-22
15: {id: "17500", name: "Marian Alexandru Hanghiuc", created_at: "2019-08-
16: {id: "17651", name: "Lasse Bundsgaard Hansen", created_at: "2019-08-03
17: {id: "17618", name: "Mads Villads Hoe", created_at: "2019-08-03T02:26:
18: {id: "17495", name: "Ana-Maria Iacovache", created_at: "2019-08-02T02:
19: {id: "23531", name: "Søren Bo Jørgensen", created_at: "2020-08-04T02:4
20: {id: "17493", name: "Sona Juhásová", created_at: "2019-08-02T02:27:01+
21: {id: "117", name: "Anne Kirketerp", created_at: "2017-05-10T14:09:50+0
4 / 135 requests
7.9


Course roster: WU-E21s - 1. se
eaaa.instructure.com/courses/13351/users
Incognito
Account
Admin
Dashboard
Courses
Calendar
Inbox
History
Commons
Student links
WU-E21s › People
Student view
Home
Announcements
Modules
People
BigBlueButton (Formerly Confer-ences)
Assignments
Discussions
Grades
Pages
Files
Syllabus
Outcomes
Rubrics
Quizzes
Collaborations
Settings
Everyone
Groups
+ Group set
Search people
All roles
+ People
Name
Login ID
SIS ID
Section
Role
Last Activity
Lasse Aakjær
eaalaak@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 22:27
Nicklas Eibye Andersen
eaanean@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 10:00
Piotr Andrzej Pospiech
eaapipo@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 12:54
Lasse Bickmann
eaalbic@students.eaaa.dk
WU-E21s - 1. semester
Student
Thomas Hyllegaard Busk
eaathb@students.eaaa.dk
WU-E21s - 1. semester
Student
24 Aug at 17:29
Jesper Nissen Byg
eaajnb@students.eaaa.dk
WU-E21s - 1. semester
Student
26 Aug at 10:49
Barbora Byrtusová
eaababy@students.eaaa.dk
WU-E21s - 1. semester
Student
25 Aug at 14:27
Rasmus Cederdorff
race@eaaa.dk
WU-E21s - 1. semester
Teacher
26 Aug at 17:03
Elements
Console
Sources
Network
Performance
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
Name
unread_count
group_categories...
users?include_ina...
collect?v=1&_v=j9...
unread_count
courses?include[]...
accounts
7 / 138 requests
Headers
Preview
Response
Initiator
Timing
Cookies
[{id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02:00"
0: {id: "17636", name: "Lasse Aakjær", created_at: "2019-08-03T02:26:25+02
1: {id: "17929", name: "Nicklas Eibye Andersen", created_at: "2019-08-07T0
2: {id: "17476", name: "Piotr Andrzej Pospiech", created_at: "2019-08-02T0
3: {id: "30257", name: "Lasse Bickmann", created_at: "2021-08-05T00:58:06+
4: {id: "30252", name: "Thomas Hyllegaard Busk", created_at: "2021-08-05T0
5: {id: "11879", name: "Jesper Nissen Byg", created_at: "2018-08-08T02:10:
6: {id: "17960", name: "Barbora Byrtusová", created_at: "2019-08-07T02:34:
7: {id: "14427", name: "Rasmus Cederdorff", created_at: "2018-09-06T02:23:
avatar_url: "https://eaaa.instructure.com/images/thumbnails/649049/1EeRF
created_at: "2018-09-06T02:23:57+02:00"
custom_links: []
email: "race@eaaa.dk"
enrollments: [{course_id: "13351", id: "320419", user_id: "14427", type:
id: "14427"
integration_id: null
login_id: "race@eaaa.dk"
name: "Rasmus Cederdorff"
short_name: "Rasmus Cederdorff (adjunkt – race@eaaa.dk)"
sis_user_id: null
sortable_name: "Cederdorff, Rasmus"
8: {id: "41", name: "Jeffrey David Serio", created_at: "2017-05-10T14:09:2
9: {id: "30251", name: "Sarah Øster Dybvad", created_at: "2021-08-05T00:57

IT'S ALL OBJECTS & ARRAYS

DATA STRUCTURES

OBJECTS & ARRAYS

ARRAYS

LOOPS

```
for (let teacher of teachers) {
  console.log(teacher);
}
```

```
▶ {name: "Birgitte Kirk Iversen", mail: "bki@baaa.dk"}    main.js:20
▶ {name: "Michael Hvidtfeldt", mail: "mhv@baaa.dk"}       main.js:20
▶ {name: "Rasmus Cederdorff", mail: "race@baaa.dk"}       main.js:20
```

FOR OF LOOP

ITERATE OVER ARRAYS OR OTHER ITERABLE OBJECTS

https://scrimba.com/learn/introductiontojavascript/for-loops-cMMM8U9

https://scrimba.com/learn/introductiontojavascript/challenge-for-loops-cPkpJrcv

LOOPS

```
for (const familyMember of familyMembers) {
  console.log(familyMember);
}

for (let index = 0; index < familyMembers.length; index++) {
  const familyMember = familyMembers[index];
  console.log(familyMember);
}
```

HTTPS://WWW.W3SCHOOLS.COM/JS/JS_LOOP_FOR.ASP

HTTPS://JAVASCRIPT.INFO/ARRAY#LOOPS

HTTPS://JAVASCRIPT.INFO/WHILE-FOR

ARRAY METHODS

https://javascript.info/array-methods#filter

filter

The `find` method looks for a single (first) element that makes the function return `true`.

If there may be many, we can use arr.filter(fn).

The syntax is similar to `find`, but `filter` returns an array of all matching elements:

```
let results = arr.filter(function(item, index, array) {
  // if true item is pushed to results and the iteration continues
  // returns empty array if nothing found
});
```

For instance:

```
let users = [
  {id: 1, name: "John"},
  {id: 2, name: "Pete"},
  {id: 3, name: "Mary"}
];

// returns array of the first two users
let someUsers = users.filter(item => item.id < 3);

alert(someUsers.length); // 2
```

Transform an array

Let's move on to methods that transform and reorder an array.

map

The arr.map method is one of the most useful and often used.

It calls the function for each element of the array and returns the array of results.

FUNCTIONS

...A BLOCK OF CODE TO PERFORM A SPECIFIC TASK & TO MAKE OUR CODE REUSABLE

A WAY OF STORING OUR CODE SO WE CAN USE IT AGAIN AND AGAIN AND REUSE IT

BEST PRACTICE: WRITE REUSABLE CODE

https://scrimba.com/p/pPPeCy/cpKZdcQ

FUNCTIONS

FUNCTION DECLARATION

```
function log(message) {
  console.log(message);
}

log("Hi Frontenders!");
```

https://javascript.info/function-basics

FUNCTIONS

FUNCTION DECLARATION

```
console.log("Hi Frontenders!");
console.log("Good job!");
console.log("I'm testing something!");
console.log("Hola");
```

```
function log(message) {
  console.log(message);
}

log("Hi Frontenders!");
log("Good job!");
log("I'm testing something!");
log("Hola");
```

https://javascript.info/function-basics

FUNCTIONS

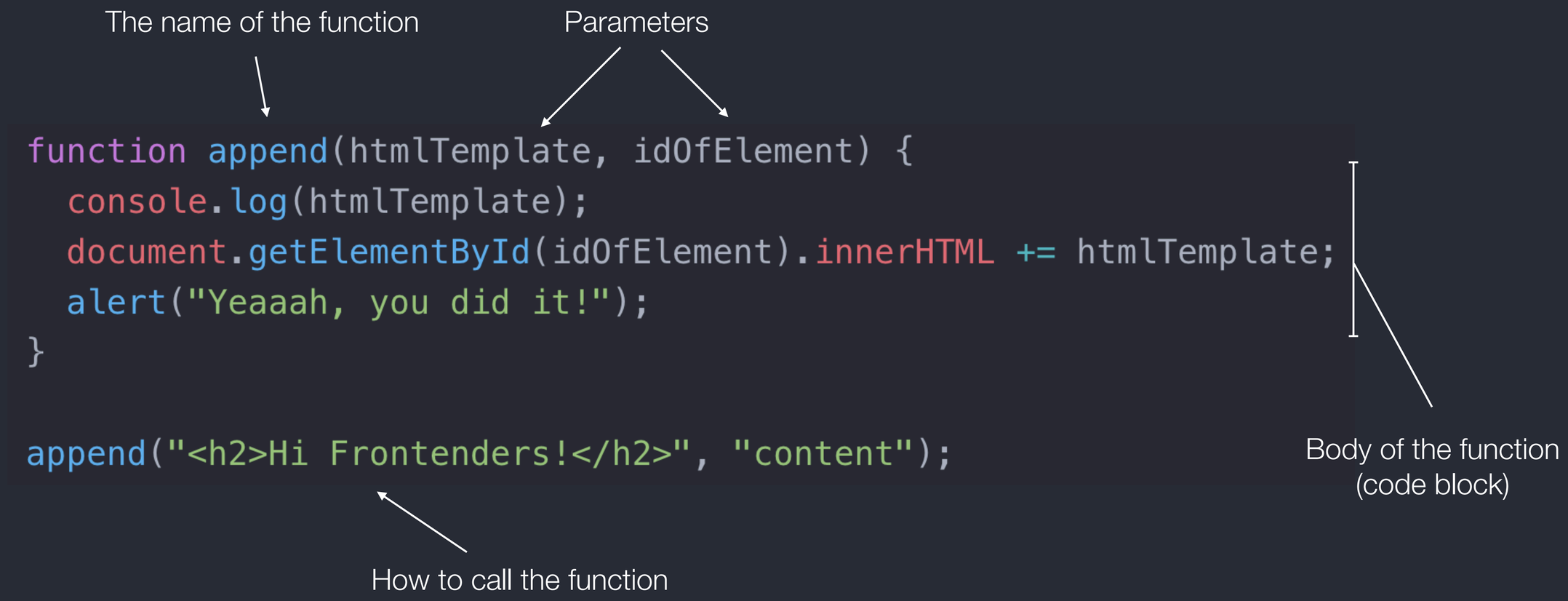
FUNCTION DECLARATION

```
function append(htmlTemplate, idOfElement) {
  console.log(htmlTemplate);
  document.getElementById(idOfElement).innerHTML += htmlTemplate;
  alert("Yeaaah, you did it!");
}

append("<h2>Hi Frontenders!</h2>", "content");
```

https://javascript.info/function-basics

FUNCTIONS

FUNCTION DECLARATION



https://javascript.info/function-basics

FUNCTIONS

ARRAY & LOOPS

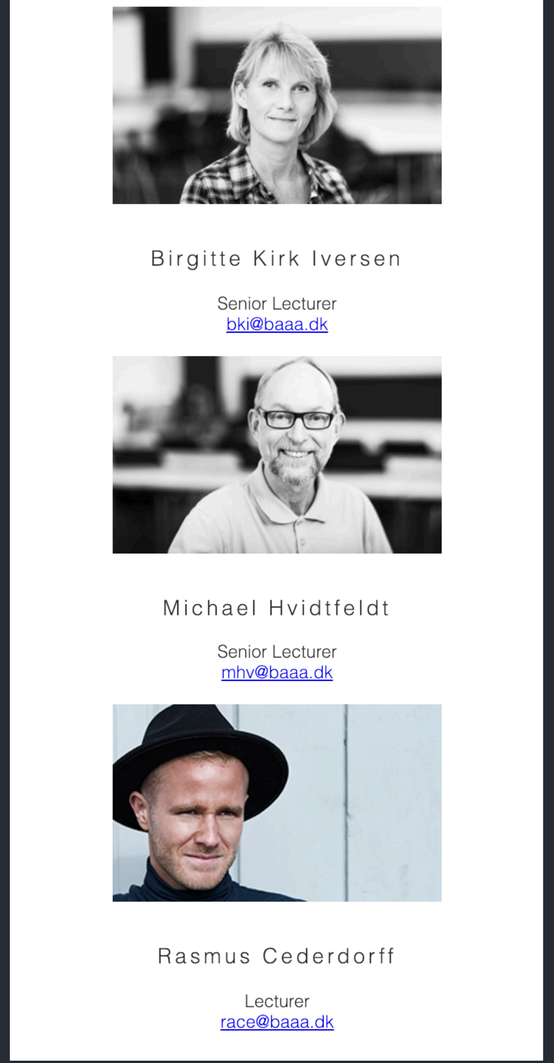
The name of the function

Parameters

```
function appendTeachers(teachers) {
  for (let teacher of teachers) {
    console.log(teacher);
    document.querySelector("#grid-teachers").innerHTML +=
      "<article>" +
      "<img src='" + teacher.img + "'>" +
      "<h3>" + teacher.name + "</h3>" +
      teacher.position + "<br>" +
      "<a href='mailto:" + teacher.mail + "'>" + teacher.mail + "</a>" +
      "</article>";
  }
}

appendTeachers(teachers);
```

Body of the function
(code block)

How to call the function



FUNCTIONS

```
function logPersons(persons) {
  for (var i = 0; i < persons.length; i++) {
    console.log(persons[i]);
  }
}
```

FUNCTION DECLARATION

```
const logPersons = function(persons) {
  for (var i = 0; i < persons.length; i++) {
    console.log(persons[i]);
  }
}
```

FUNCTION EXPRESSION

```
const logPersons = (persons) => {
  for (var i = 0; i < persons.length; i++) {
    console.log(persons[i]);
  }
}
```

ARROW FUNCTION

`TEMPLATE STRING`

BACKTICK STRING / TEMPLATE LITERALS

...EMBED VARIABLES AND EXPRESSIONS IN A STRING

https://scrimba.com/p/p4Mrt9/c4vJdha

`TEMPLATE STRING`

BACKTICK STRING / TEMPLATE LITERALS

- Extended functionality
- Simplifies concatenating strings
- Embed values and expression into a string with `${...}`
- Simplifies the syntax and the reading
- Let us create more readable HTML templates

```
let name = "Alicia";
console.log(`Hello, ${name}`);
```

```
Hello, Alicia                                    main.js:8
>
```

`TEMPLATE STRING`

```
let name = "Alicia";
let age = 6;

console.log(name + " is " + age + " years old.");

console.log(`${name} is ${age} years old.`);
```

```
Alicia is 6 years old.          main.js:10
Alicia is 6 years old.          main.js:12
>
```

`TEMPLATE STRING`

REGULAR STRING EXPRESSION

```
function appendTeachers(teachers) {
  for (let teacher of teachers) {
    console.log(teacher);
    document.querySelector("#grid-teachers").innerHTML +=
      "<article>" +
      "<img src='" + teacher.img + "'>" +
      "<h3>" + teacher.name + "</h3>" +
      teacher.position + "<br>" +
      "<a href='mailto:" + teacher.mail + "'>" + teacher.mail + "</a>" +
      "</article>";
  }
}
```

TEACHERS

Birgitte Kirk Iversen

Senior Lecturer
bki@baaa.dk

Michael Hvidtfeldt

Senior Lecturer
mhv@baaa.dk

Rasmus Cederdorff

Lecturer
race@baaa.dk

`TEMPLATE STRING`

...EMBED VARIABLES AND EXPRESSIONS IN A STRING

```
function appendTeachers(teachers) {
  for (let teacher of teachers) {
    console.log(teacher);
    document.querySelector("#grid-teachers").innerHTML +=
      "<article>" +
      "<img src='" + teacher.img + "'>" +
      "<h3>" + teacher.name + "</h3>" +
      teacher.position + "<br>" +
      "<a href='mailto:" + teacher.mail + "'>" + teacher.mail + "</a>" +
      "</article>";
  }
}
```

```
function appendTeachers(teachers) {
  for (let teacher of teachers) {
    console.log(teacher);
    document.querySelector("#grid-teachers").innerHTML += `
      <article>
        <img src='${teacher.img}'>
        <h3>${teacher.name}</h3>
        ${teacher.position}<br>
        <a href='mailto:${teacher.mail}'>${teacher.mail}</a>
      </article>`;
  }
}
```

`ES6 STRING HTML`

https://marketplace.visualstudio.com/items?itemName=hjb2012.vscode-es6-string-html

```
function appendTeachers(teachers) {
  for (let teacher of teachers) {
    console.log(teacher);
    document.querySelector("#grid-teachers").innerHTML += `
      <article>
        <img src='${teacher.img}'>
        <h3>${teacher.name}</h3>
        ${teacher.position}<br>
        <a href='mailto:${teacher.mail}'>${teacher.mail}</a>
      </article>`;
  }
}
```

```
function appendTeachers(teachers) {
  for (let teacher of teachers) {
    console.log(teacher);
    document.querySelector("#grid-teachers").innerHTML +=/*html*/ `
    <article>
      <img src='${teacher.img}'>
      <h3>${teacher.name}</h3>
      ${teacher.position}<br>
      <a href='mailto:${teacher.mail}'>${teacher.mail}</a>
    </article>`;
  }
}
```

CANVAS USERS CASE #1

BACK

CREATE A GRID DISPLAYING USERS FROM CANVAS

1. Make a copy of your previous project.
2. In your `app.js` define a variable called `users` holding an array.
3. Inside of the array, define at least 4 user objects with properties similar to the data structure from https://cederdorff.github.io/web-frontend/canvas-users/data.json (no use of fetch yet).
4. Use `console.log()` to test the array.
5. Create a function called `appendUsers(…)`. The function must loop through the `users` array and append each user to your grid container in the DOM.
    1. Use a `for of` loop. Inside of the loop you have to create a template for each user using a `template string`. The template is similar to the static defined from the previous exercise.
    2. Delete the static define HTML content in your grid.
    3. Make sure to execute the function and that all users are appended to the DOM.

FETCH(...)

HTTP REQUEST IN JAVASCRIPT

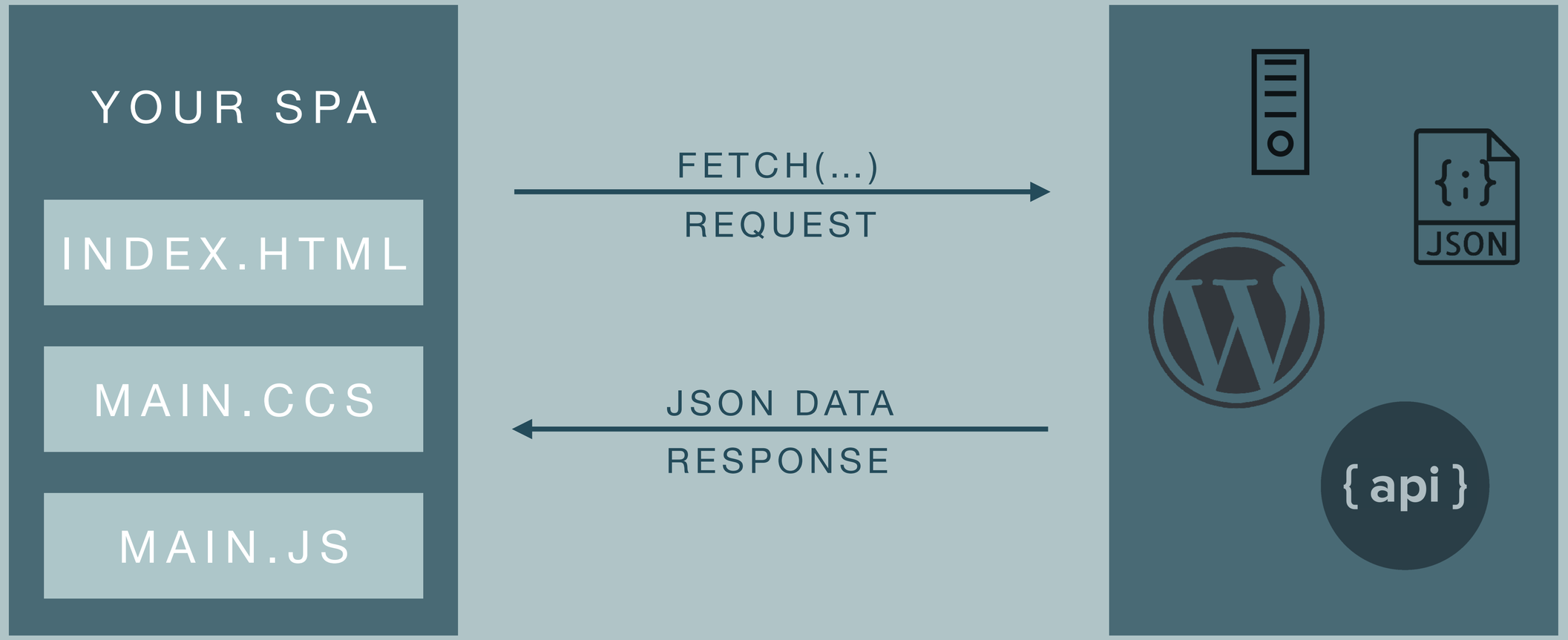
...A WAY TO GET & POST DATA FROM & TO A DATA SOURCE

FETCH

WEB APP

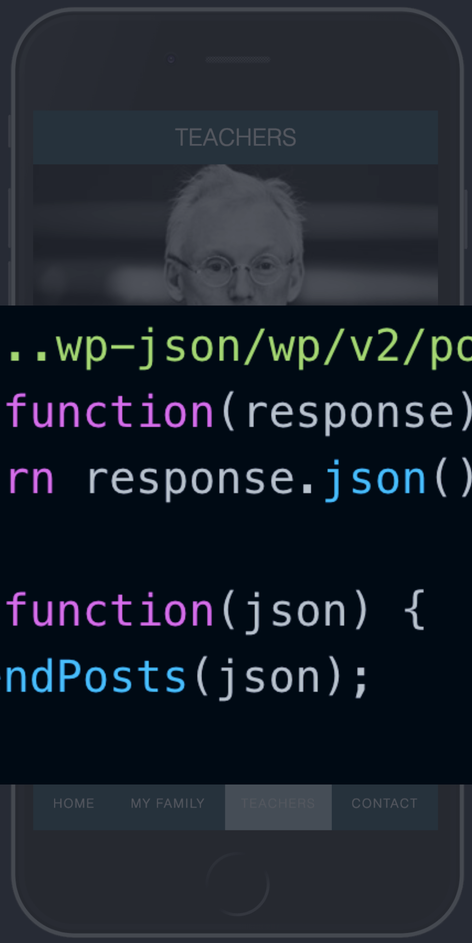
(API)

DATA SOURCE



FETCH

WEB APP




```
fetch("...wp-json/wp/v2/posts")
  .then(function(response) {
    return response.json();
  })
  .then(function(json) {
    appendPosts(json);
  });
```

REQUEST (fetch)

RESPONSE (then)

DATA SOURCE

```js
/*
Fetches json data from the file persons.json
*/
fetch('json/persons.json')
  .then(function (response) {
    return response.json();
  })
  .then(function (jsonData) {
    console.log(jsonData);
    appendPersons(jsonData)
  });

/*
Appends json data to the DOM
*/
function appendPersons(persons) {
  let htmlTemplate = "";
  for (let person of persons) {
    htmlTemplate += /*html*/`
      <article>
        <img src="${person.img}">
        <h4>${person.name}</h4>
        <p>${person.age} years old</p>
        <p>Hair color: ${person.hairColor}</p>
        <p>Relation: ${person.relation}</p>
      </article>
    `;
  }
  document.querySelector("#persons").innerHTML = htmlTemplate;
}
```


```json
[
  {
    "name": "Peter Madsen",
    "age": 52,
    "hairColor": "blonde",
    "relation": "dad",
    "img": "img/dad.jpg"
  },
  {
    "name": "Ane Madsen",
    "age": 51,
    "hairColor": "brown",
    "relation": "mom",
    "img": "img/ane.jpg"
  },
  {
    "name": "Rasmus Madsen",
    "age": 28,
    "hairColor": "blonde",
    "relation": "brother",
    "img": "img/IMG_0526_kvadrat.jpg"
  },
  {
    "name": "Mie Madsen",
    "age": 25,
    "hairColor": "brown",
    "relation": "blonde",
    "img": "img/mie.jpg"
  },
  {
    "name": "Mads Madsen",
    "age": 18,
    "hairColor": "dark",
    "relation": "blonde",
    "img": "img/mads.jpg"
  },
  {
    "name": "Jens Madsen",
    "age": 14,
    "hairColor": "blonde",
    "relation": "uncle",
    "img": "img/jenspeter.jpg"
  }
]
```

```js
/*
Fetches json data from the file persons.json
*/
fetch('json/persons.json')
  .then(function (response) {
    return response.json();
  })
  .then(function (jsonData) {
    console.log(jsonData);
    appendPersons(jsonData)
  });

/*
Appends json data to the DOM
*/
function appendPersons(persons) {
  let htmlTemplate = "";
  for (let person of persons) {
    htmlTemplate += /*html*/`
      <article>
        <img src="${person.img}">
        <h4>${person.name}</h4>
        <p>${person.age} years old</p>
        <p>Hair color: ${person.hairColor}</p>
        <p>Relation: ${person.relation}</p>
      </article>
    `;
  }
  document.querySelector("#persons").innerHTML = htmlTemplate;
}
```

REQUEST (fetch)

RESPONSE (then)

```json
[
  {
    "name": "Peter Madsen",
    "age": 52,
    "hairColor": "blonde",
    "relation": "dad",
    "img": "img/dad.jpg"
  },
  {
    "name": "Ane Madsen",
    "age": 51,
    "hairColor": "brown",
    "relation": "mom",
    "img": "img/ane.jpg"
  },
  {
    "name": "Rasmus Madsen",
    "age": 28,
    "hairColor": "blonde",
    "relation": "brother",
    "img": "img/IMG_0526_kvadrat.jpg"
  },
  {
    "name": "Mie Madsen",
    "age": 25,
    "hairColor": "brown",
    "relation": "blonde",
    "img": "img/mie.jpg"
  },
  {
    "name": "Mads Madsen",
    "age": 18,
    "hairColor": "dark",
    "relation": "blonde",
    "img": "img/mads.jpg"
  },
  {
    "name": "Jens Madsen",
    "age": 14,
    "hairColor": "blonde",
    "relation": "uncle",
    "img": "img/jenspeter.jpg"
  }
]
```

```
/*
Fetches json data from the file persons.json
*/
fetch('json/persons.json')
  .then(function (response) {
    return response.json();
  })
  .then(function (jsonData) {
    console.log(jsonData);
    appendPersons(jsonData)
  });

/*
Appends json data to the DOM
*/
function appendPersons(persons) {
  let htmlTemplate = "";
  for (let person of persons) {
    htmlTemplate += /*html*/`
      <article>
        <img src="${person.img}">
        <h4>${person.name}</h4>
        <p>${person.age} years old</p>
        <p>Hair color: ${person.hairColor}</p>
        <p>Relation: ${person.relation}</p>
      </article>
    `;
  }
  document.querySelector("#persons").innerHTML = htmlTemplate;
}
```

```
},
{
  "name": "Rasmus Madsen",
  "age": 28,
  "hairColor": "blonde",
  "relation": "brother",
  "img": "img/IMG_0526_kvadrat.jpg"
},
{
  "name": "Mie Madsen",
  "age": 25,
  "hairColor": "brown",
  "relation": "blonde",
  "img": "img/mie.jpg"
},
{
  "name": "Mads Madsen",
  "age": 18,
  "hairColor": "dark",
  "relation": "blonde",
  "img": "img/mads.jpg"
},
{
```

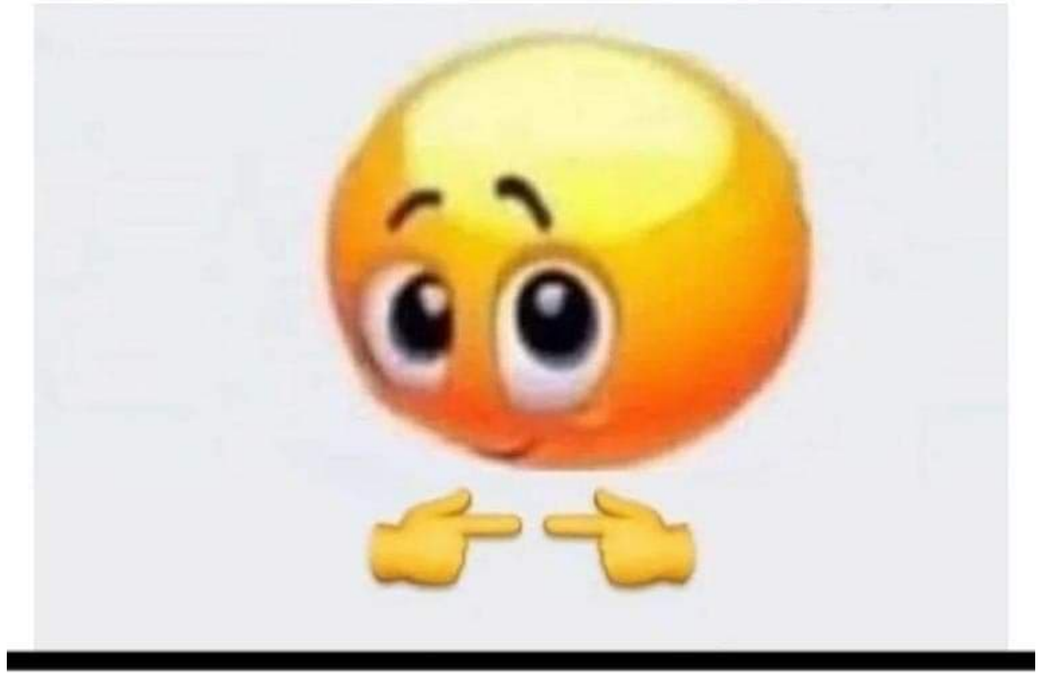
FETCH

... IN THREE DIFFERENT WAYS

```
// Simple javascipt 😊

//Synchronous fetch using async/await.

// Usual way
✓ const jsonData = fetch('URL')
                   .then(response ⇒ response.json())
                   .then(json ⇒ console.log(json));

// Using await
✓ const jsonData = await fetch('URL').then(res ⇒ res.json())


// Shorter syntax 😍
✓ const jsonData = await (await fetch('URL')).json();
```

https://www.instagram.com/p/B0nxQjXj9Zi/

will you be async to my await?



FETCH WITH CALLBACKS OR ASYNC/AWAIT

```
// fetch with callbacks
fetch("https://cederdorff.github.io/web-frontend/canvas-users/data.json")
    .then(function (response) {
        return response.json();
    })
    .then(function (data) {
        console.log(data);
    });
//
//
// or with async/await
const response = await fetch("https://cederdorff.github.io/web-frontend/canvas-users/data.json");
const data = await response.json();
console.log(data);
```

CANVAS USERS CASE #2

BACK

FETCH USERS FROM JSON & DISPLAY IN GRID

1. Make a copy of your previous project.
2. In your `app.js` create a function called `fetchUsers()`.
3. In `fetchUsers()` implement the logic to fetch users from the following url: https://cederdorff.github.io/web-frontend/canvas-users/data.json
4. Use `console.log` to test the result (the fetched data).
5. Remove the *in script* defined `users` array (from previous exercise) and save the fetched result in the `users` variable.
6. Execute and use `appendUsers(…)` to append the newly fetched users from the json file.

CANVAS USERS CASE #2

BACK

EXTRA

1. Implement search functionality searching on `name.includes(searchValue)`.
2. Implement sort functionality in order to sort by `name`, `sortableName`, etc.
3. Make it possible to filter users on `enrollment_type` - students and teachers
4. Implement create functionality using `array.push()`. Remember generating a dummy `id` when adding a new user.
5. Create and implement an update view using the `id` of the user to find and update properties of the user in the `users` array.
6. Implement delete functionality using the `id` to filter the `users` array.
7. Create a detail view displaying all properties of the selected user.

NEXT UP

MORE JAVASCRIPT

WEB APP DEVELOPMENT

SINGLE PAGE APPLICATIONS & ROUTING

MORE FETCH, JSON & ASYNC JS

CODE EVERY DAY!