

race.html is loading...



# HTML

Opskriften

Datamatiker

**kea**  
KØBENHAVNS ERHVERVSAKADEMI

**YOU ARE A PROGRAMMER?  
IN WHICH LANGUAGE YOU CODE?**

**HTML**



# Dagens Formål

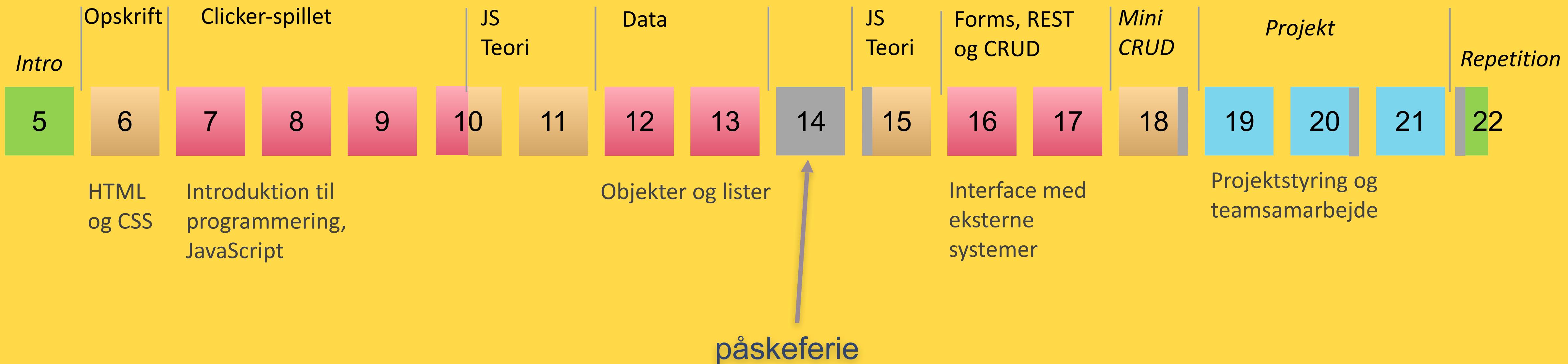
- Introduktion til HTML
- Struktur af en webside
- Semantiske HTML-elementer
- Hands-on med HTML og strukturering af jeres opskrift
- Opret nyt projekt, commit og push til GitHub

# Agenda

- Aflevering: Yndlingsopskrift
- Webudvikling Generelt
  - HTML, CSS og JS
  - CRUD App
  - Client-server & HTTP
- Introduktion til HTML
- HTML Semantic Elements
- Opskrift 1: HTML, struktur og indhold
- Commit og push til GitHub

# 1. SEMESTERS STRUKTUR

Semesteret består af 16 undervisningsuger (inkl. en masse helligdage)



Eksamens dato: 12.-14. juni

# Opskrift

HTML & CSS

CSS Grid

Indhold (tekst, billeder, fremgangsmåde, mm.)

Deadline:

12-02-2023 23:59

The screenshot shows a recipe for "American chocolate cookies" on the Arla website. The page features a header with the Arla logo and navigation links for "ARLA NU", "OPSKRIFTER", "PRODUKTER", "OM ARLA", "BÆREDYGTIGHED", and "CLICK&COOK®". Below the header, a breadcrumb trail reads "Arla > Opskrifter > Cookies". The main title is "Cookies" with a subtitle "Klassiske amerikanske cookies - spredte udenpå og bløde indeni...". It includes a cooking time of "45 MIN", a "FRYSSEGNET" badge, a 4-star rating with 1329 reviews, and a "Kommenter" link. The text describes the classic American cookie as soft and chewy on the inside and spread on the outside, mentioning that calling them Chocolate chip cookies or just Cookies makes no difference. It also notes the addition of chocolate chunks. Below the text are four action buttons: "GEM", "TILFØJ TIL INDKØBSLISTE", "DEL", and "PRINT". There are also links to "BilkaGo", "coop.dk", and "REMA 1000". The central part of the page displays a large image of several chocolate chip cookies stacked on a piece of parchment paper with some chocolate chunks scattered around. To the left, there's a section titled "Ingredienser" with a "1 portion" button and a note "Ca. 20 stk.". A table lists the ingredients: Blædt smør (150 g), Sukker (ca. 1½ dl) (125 g), Brun farin (ca. 1½ dl) (125 g), Vaniljesukker (2 tsk), and Æg (1). To the right, a section titled "Sådan gør du" provides a step-by-step guide for making the cookies.

Ingrediens	Mængde
Blædt smør	150 g
Sukker (ca. 1½ dl)	125 g
Brun farin (ca. 1½ dl)	125 g
Vaniljesukker	2 tsk
Æg	1

**Sådan gør du**

- Pisk smør, sukker, brun farin, vanilje og æg godt sammen.
- Bland hvedemel, natron, salt og bagepulver og rør det i smørblandingen sammen med chokolade.
- Del dejen i 20 stykker a ca. 45 g og form dem til kugler.
- 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.
- Bag kagerne til de er gyldne, men bløde i midten.

# Opskrift

Aflevering: Lav din yndlingsopskrift i  
HTML og CSS

Delopgaver:

- Opskrift 1: HTML, struktur og indhold
- Opskrift 2: CSS - basal styling
- Øvelse: Deploy din opskrift til GitHub Pages
- Opskrift 3: CSS Grid layout

The screenshot shows a digital assignment board titled "Opskrift". The board includes a header with navigation links like "Startside", "Rum", "Dashboards og projektrum", "Mere", and a user profile for "Rasmus Cederdorff". Below the header, the main content area is titled "Emne / Opskriften".  
**BESKRIVELSE**  
I dette forløb introducerer vi teknologierne HTML og CSS. Vi skal lære værktøjerne bedre at kende og få erfaring med deployment - at publicere jeres løsning, så den er tilgængelig online på webbet. Alt dette skal vi arbejde med i forbindelse med at I implementerer jeres yndlingsopskrift. Læs mere om afleveringsopgaven [her](#).

**Opskriften - HTML** [OPSKRIFTEN]  
I dag skal vi se nærmere på HTML, og hvordan vi anvender HTML til at strukturere et website. Derudover skal I introduceres til jeres første afleveringsopgave: Aflevering: Lav di...  
Start → | 6. feb. | 0/4

**Opskriften - CSS** [OPSKRIFTEN]  
Vi skal dykke ned i CSS og hvordan styling beskrives med CSS, så du kan style din opskrift. Emner: Introduktion til CSS CSS-syntaks Regler, deklarationer og selectorer CSS Box Mode...  
Start → | 7. feb. | 0/4

**Deployment** [OPSKRIFTEN]  
Du har indtil nu kørt, testet og vist din opskrift lokalt på din egen maskine. I dag skal vi arbejde med, hvordan vi kan publicere jeres opskrift, så den bliver tilgængelig online. Det vil...  
Start → | 8. feb. | 0/3

**Opskriften - CSS layout** [OPSKRIFTEN]  
Vi skal arbejde mere med CSS og styling. Dagens fokus er CSS grid layout, der gør muliggør et fleksibelt layout af din HTML-side. Efter undervisningen vil du være i stand til at anvende...  
Start → | 9. feb. | 0/3

# Krav

- HTML og CSS
- HTML5 Semantiske elementer (tags)
- CSS Grid
- Git, GitHub og GitHub Desktop til at dele din kode
- GitHub Pages til at publicere din løsning
- Værktøjer som VS Code og extensions til at udvikle din løsning
- ... læs mere på Fronter

Fronter

kea-fronter.itslearning.com/ContentArea/ContentArea.aspx?LocationID=5795&LocationType=1&Ele...

Startside Rum Dashboards og projektrum Mere ▾

Rasmus Cederdorff

Dat23v1 - 1. sem Oversigt Planer Ressourcer Deltagere Mere ▾ Tilføj

Afleveringsopgave Rediger

Lav din yndlingsopskrift i HTML og CSS



I denne afleveringsopgave skal du lave din yndlingsopskrift og publicere den via GitHub Pages. Opskriften skal implementeres med HTML og CSS på baggrund af præsenterede og indlærte teknikker fra undervisningen. Du kan finde inspiration til din opskrift her:

- <https://www.valdemarsro.dk/opskrifter/>
- <https://www.arla.dk/opskrifter/>
- <https://www.louisesmadblog.dk/opskrifter/>

Det er ikke indholdet (tekst og billeder) som er i fokus men at få erfaring med teknologierne HTML, CSS, Git og værktøjerne. Du må derfor gerne anvende tekst, billeder og andet indhold fra eksisterende madopskrifter. Hver dog opmærksom på, at det ikke er alle sider, du må benytte billeder fra.

# CSS Grids

**CSS Grid Teachers**

The screenshot shows a grid layout with three columns. Each column contains a portrait photo and the name and contact information for a teacher:

- Birgitte Kirk Iversen**  
Senior Lecturer  
[hki@mail.dk](mailto:hki@mail.dk)
- Martin Aagaard Nøhr**  
Lecturer  
[mnor@mail.dk](mailto:mnor@mail.dk)
- Peter Lind**  
Senior Lecturer  
[pell@kea.dk](mailto:pell@kea.dk)

Below the grid, there are three more portrait photos.

**Header**

The screenshot shows a header section divided into three equal-width columns. Each column is labeled "Column". Below the header, there is a footer section labeled "Footer".

**Amerikanske chocolate chip cookies**

**Cookies**

Klassiske amerikanske cookies - sprøde udenpå og bløde indeni... 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 prægtfuldt. Prøv denne opskrift på den perfekte Chocolate chip cookie, næste gang lækkersulten banker på døren!

**Ingredienser**

**Cookies**

- Bledd smør, 150 g
- Sukker, (ca. 1½ dl) 125 g
- Brun farin, (ca. 1½ dl) 125 g
- Vanille-sukker, 2 tsk
- Eg, 1
- Hvedemel (ca. 4½ dl), 250 g
- Natron, 1 tsk
- Grit salt, ¼ tsk
- Bagepulver, ½ tsk

**Sådan gør du**

1. Pisk smør, sukker, brun farin, vanille og æg godt sammen.  
2. Bland hvedemel, natron, salt og bagepulver og før det i smørblandingen sammen med chokolade.  
3. Del dejen i 20 stykker a ca. 45 g og form dem til kugler.  
4. Sat kuglerne på plader med bagepapir - tryk dem let flate. Kagerne fylder en del ud, så sørг for, at der er god plac til mellem 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.

**Bagetid**

# HTML struktur, CSS Grid & JS Fetch (CRUD)

CRUD App



Lars Bogetoft  
Head of Education  
[larb@kea.dk](mailto:larb@kea.dk)

[UPDATE](#) [DELETE](#)



Peter Lind  
Senior Lecturer  
[pel@kea.dk](mailto:pel@kea.dk)

[UPDATE](#) [DELETE](#)



Magdalena "Lenka" Otap  
Lecturer  
[mago@kea.dk](mailto:mago@kea.dk)

[UPDATE](#) [DELETE](#)



Rasmus Cederdorff  
Senior Lecturer  
[race@kea.dk](mailto:race@kea.dk)

[UPDATE](#) [DELETE](#)

Create a new User

Magdalena "Lenka" Otap

Lecturer

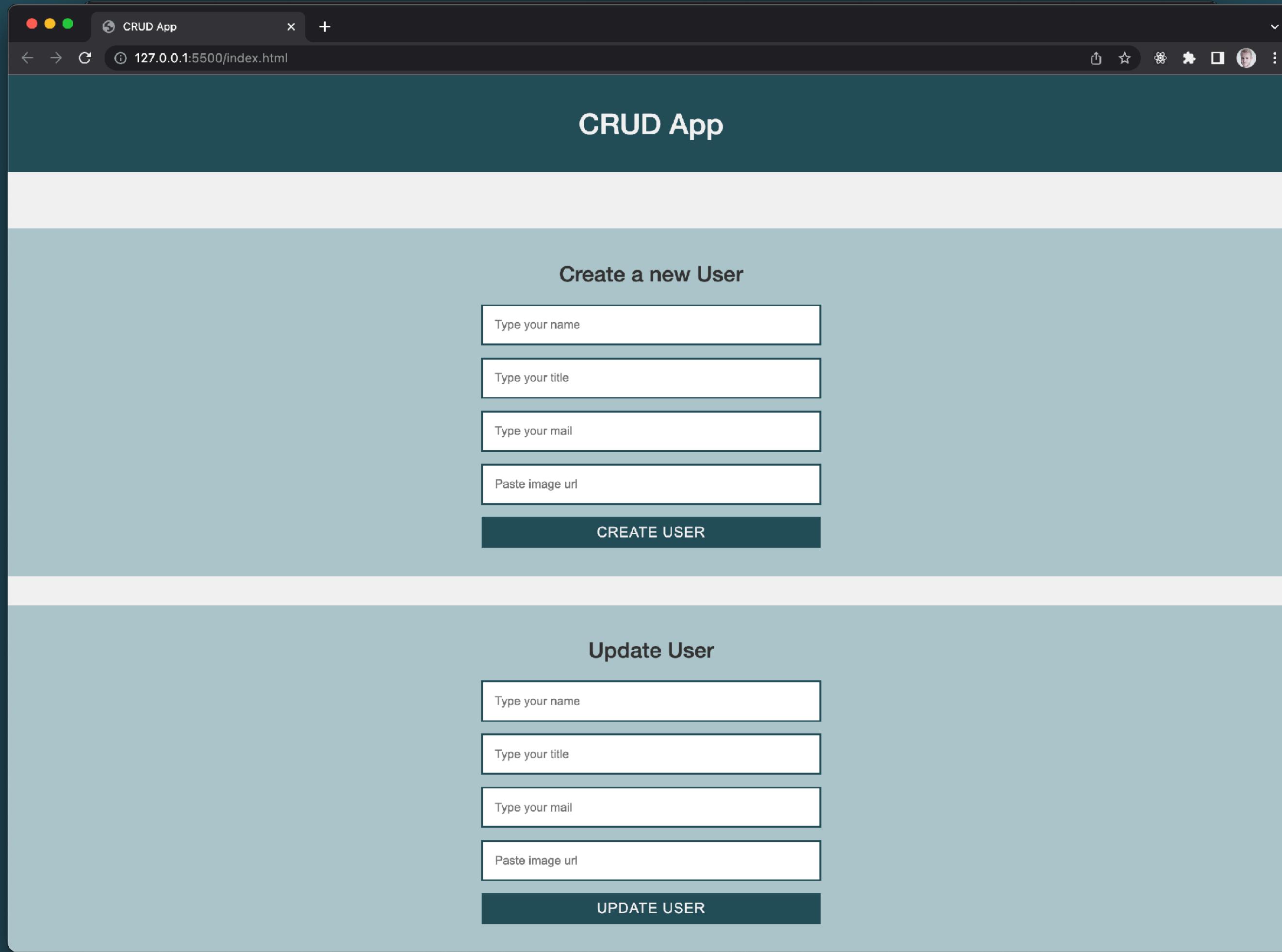
mago@kea.dk

<https://share.cederdorff.com/images/mago.jpg>

[CREATE USER](#)

Update User

# HTML struktur & CSS Grid



# HTML struktur & JS Fetch (CRUD)

CRUD App



**Lars Bogetoft**

Head of Education

[larb@kea.dk](mailto:larb@kea.dk)

[Update](#) [Delete](#)



Magdalena ‘Lenka’ Otap



**Rasmus Cederdorff**

Senior Lecturer

[race@kea.dk](mailto:race@kea.dk)

[Update](#) [Delete](#)

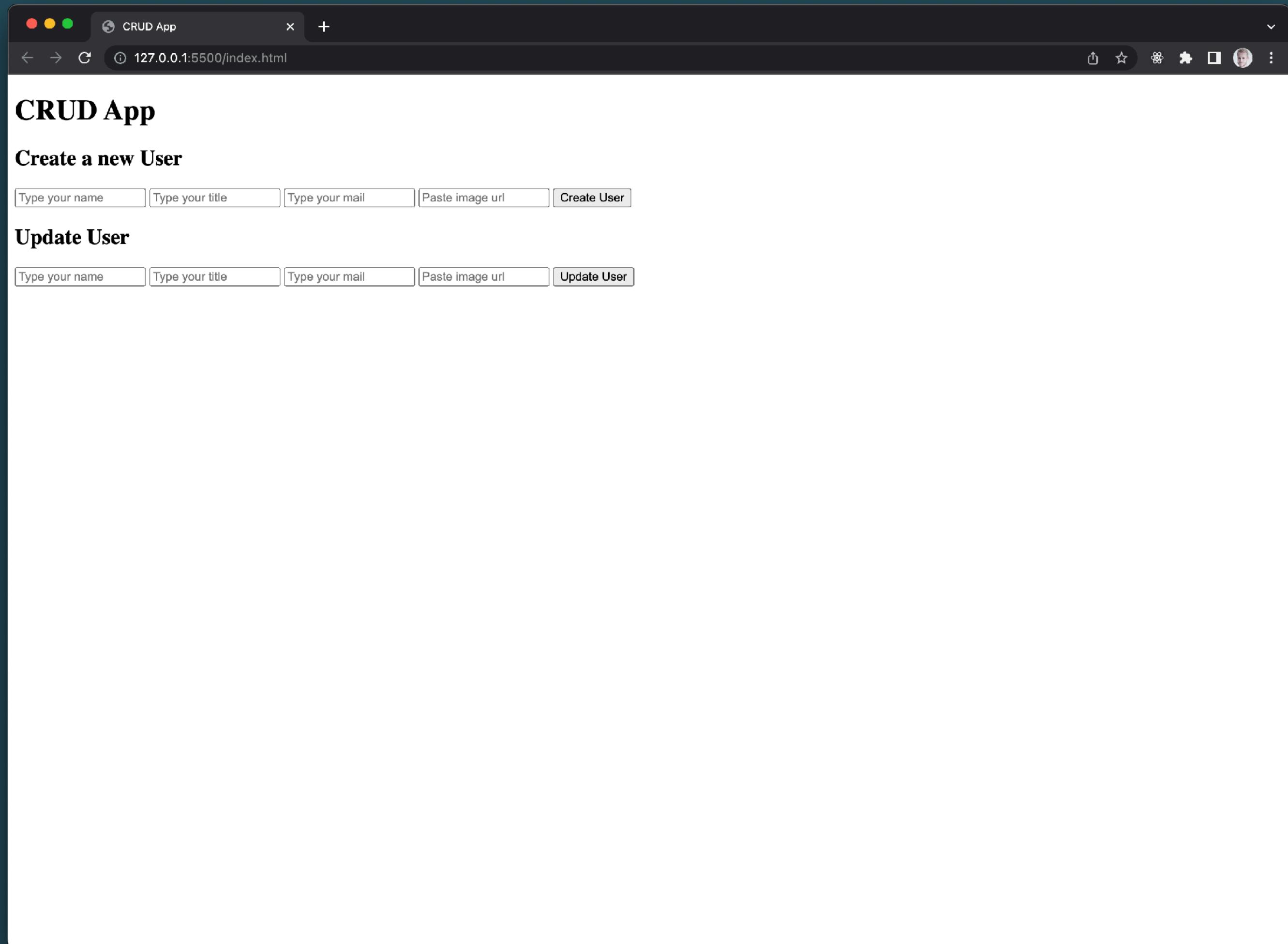
**Create a new User**

Type your name | Type your title | Type your mail | Paste image url | Create User

**Update User**

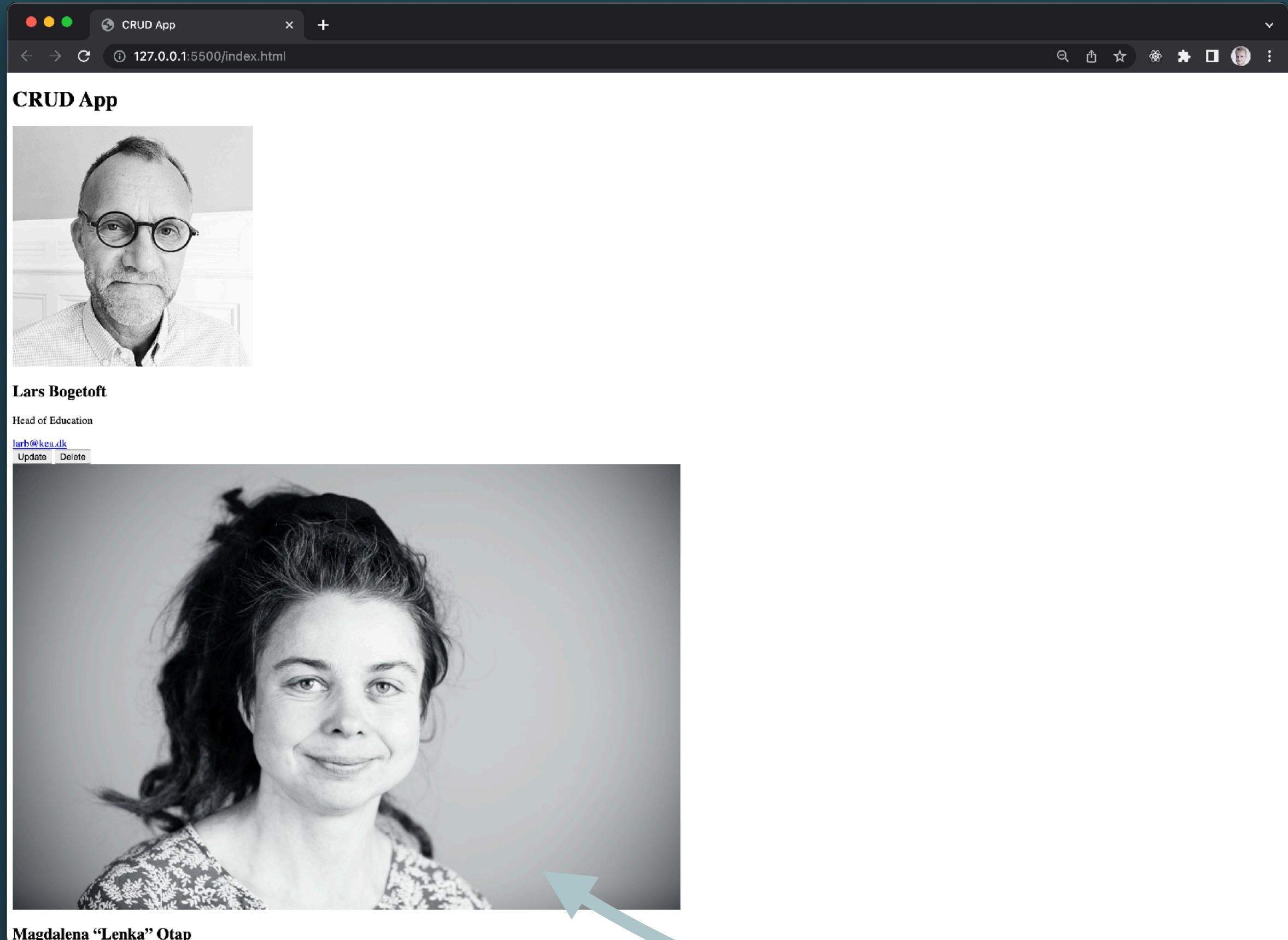
Type your name | Type your title | Type your mail | Paste image url | Update User

# HTML struktur



# JS til at generere HTML

## Og DOM-Manipulation



```
JS app.js      app.js — dat.js
01-introduction > crud-intro > JS app.js > ⚡ displayUsers
 7
 8 // ===== READ ===== //
 9 // Read (GET) all users from Firebase (Database) using REST API
10 async function readUsers() {
11   const response = await fetch(`${endpoint}/users.json`);
12   const data = await response.json();
13   const users = Object.keys(data).map(key => ({ id: key, ...data[key] }));
14   return users;
15 }
16
17 // Create HTML and display all users from given list
18 function displayUsers(list) {
19   // reset <section id="users-grid" class="grid-container">...</section>
20   document.querySelector("#users-grid").innerHTML = "";
21   //loop through all users and create an article with content for each
22   for (const user of list) {
23     document.querySelector("#users-grid").insertAdjacentHTML(
24       "beforeend",
25       /*html*/
26       <article>
27         
28         <h2>${user.name}</h2>
29         <p>${user.title}</p>
30         <a href="mailto:${user.mail}">${user.mail}</a>
31         <div class="btns">
32           <button class="btn-update-user">Update</button>
33           <button class="btn-delete-user">Delete</button>
34         </div>
35       </article>
36     );
37   }
38 }
```

# Frontend stack

**HTML**



**JS**



**CSS**



STRUCTURE  
CONTENT

FUNCTIONALITY  
BEHAVIOR

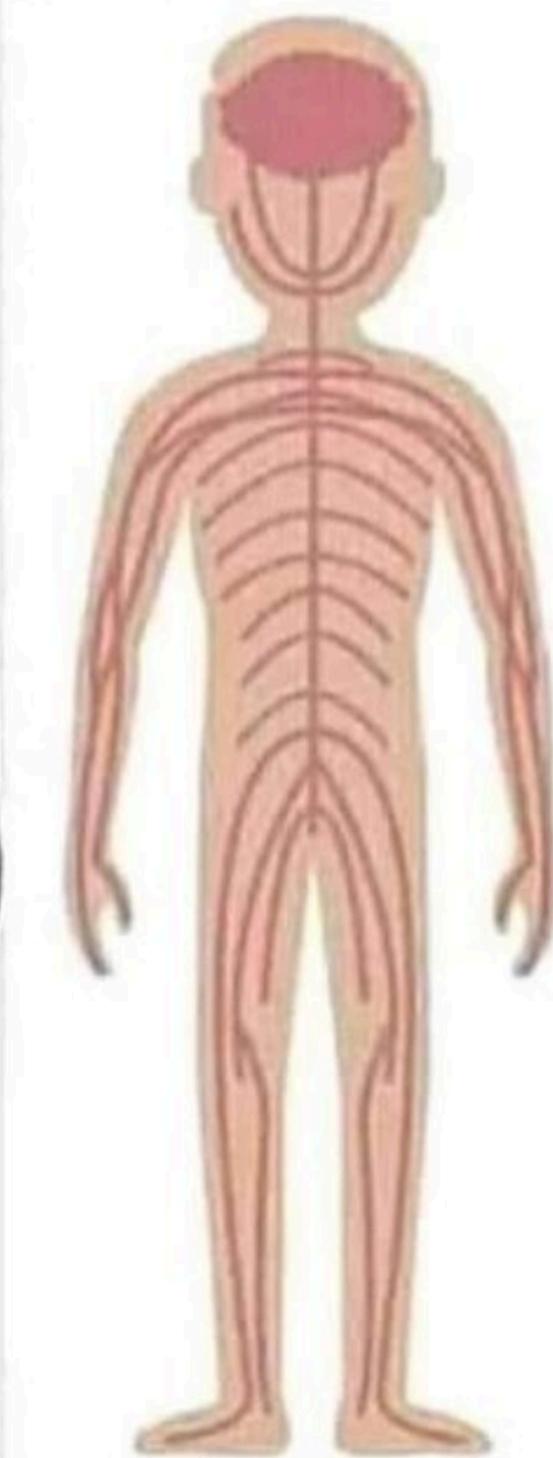
LAYOUT/STYLING  
PRESENTATION

HTML



Struktur  
Skelet

JS



Funktionalitet  
Nervesystem

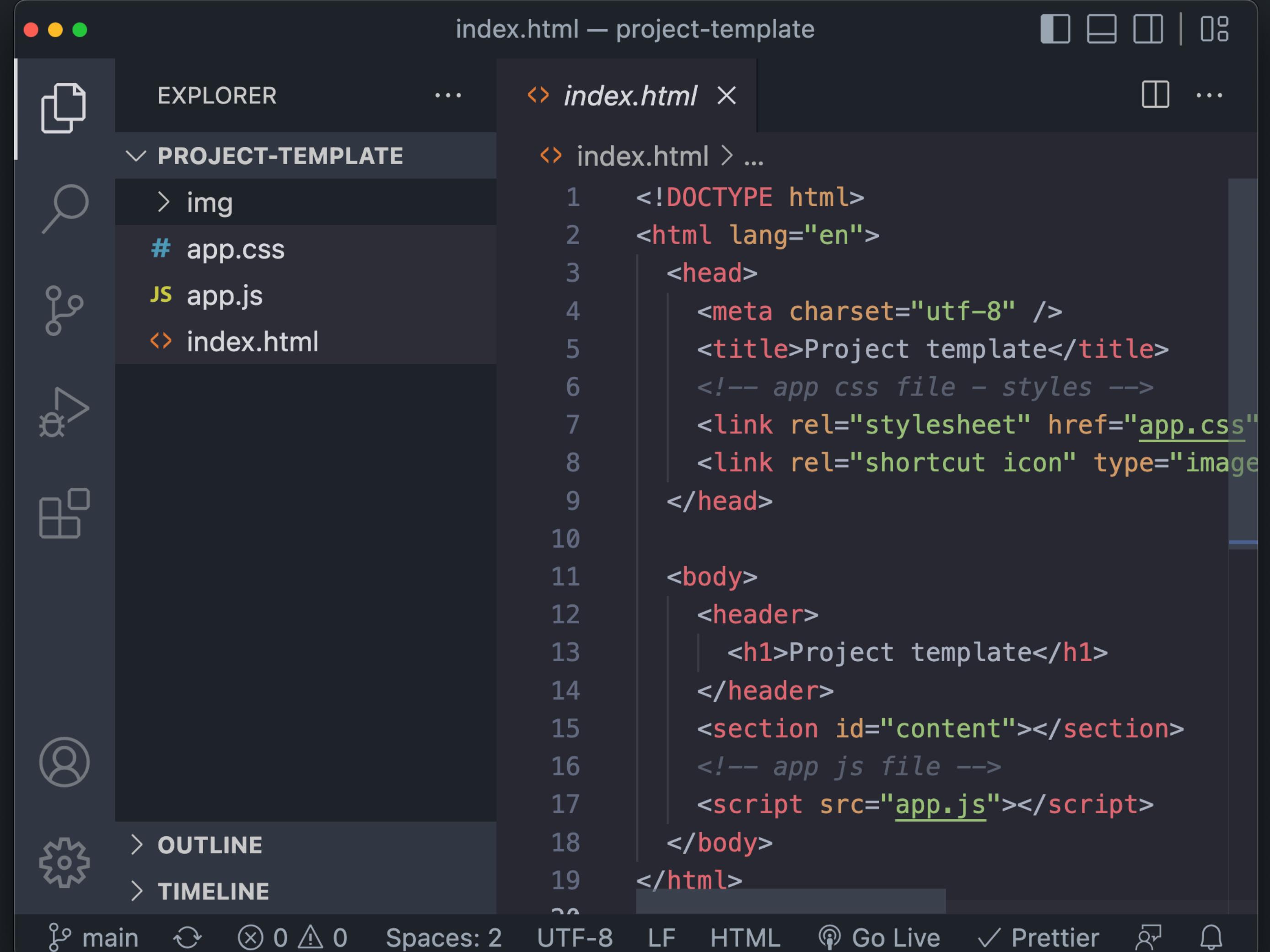
CSS



Styling  
Tøj & makeup

# Frontend Project Structure

- Project structure
  - HTML
  - CSS
  - JavaScript
- Keep a good structure & strive for consistency
- Separation of concerns



The screenshot shows a code editor interface with a dark theme. On the left, the Explorer sidebar displays a project structure under 'PROJECT-TEMPLATE' containing 'img', '# app.css', 'JS app.js', and 'index.html'. The 'index.html' file is selected and shown in the main editor area. The editor shows the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>Project template</title>
    <!-- app css file - styles -->
    <link rel="stylesheet" href="app.css" type="text/css" />
    <link rel="shortcut icon" type="image/x-icon" href="img/icon.png" />
</head>
<body>
    <header>
        <h1>Project template</h1>
    </header>
    <section id="content"></section>
    <!-- app js file -->
    <script src="app.js"></script>
</body>
</html>
```

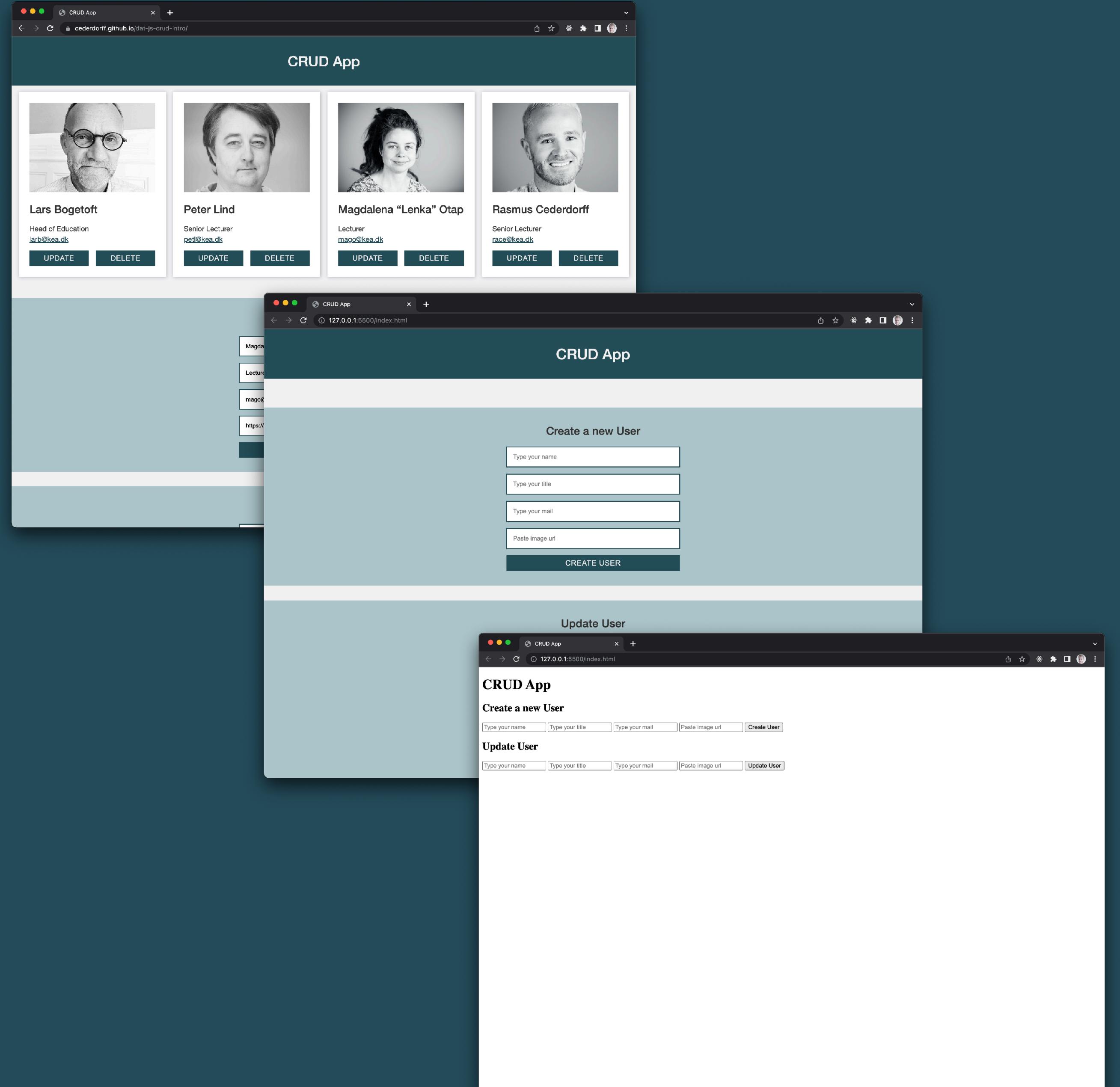
The status bar at the bottom indicates the file is 'index.html – project-template', has 08 tabs open, and shows icons for main, refresh, search, 0 errors, 0 warnings, spaces: 2, LF, HTML, Go Live, Prettier, and a few others.

# HTML, CSS & JavaScript

You can literally build anything with it!

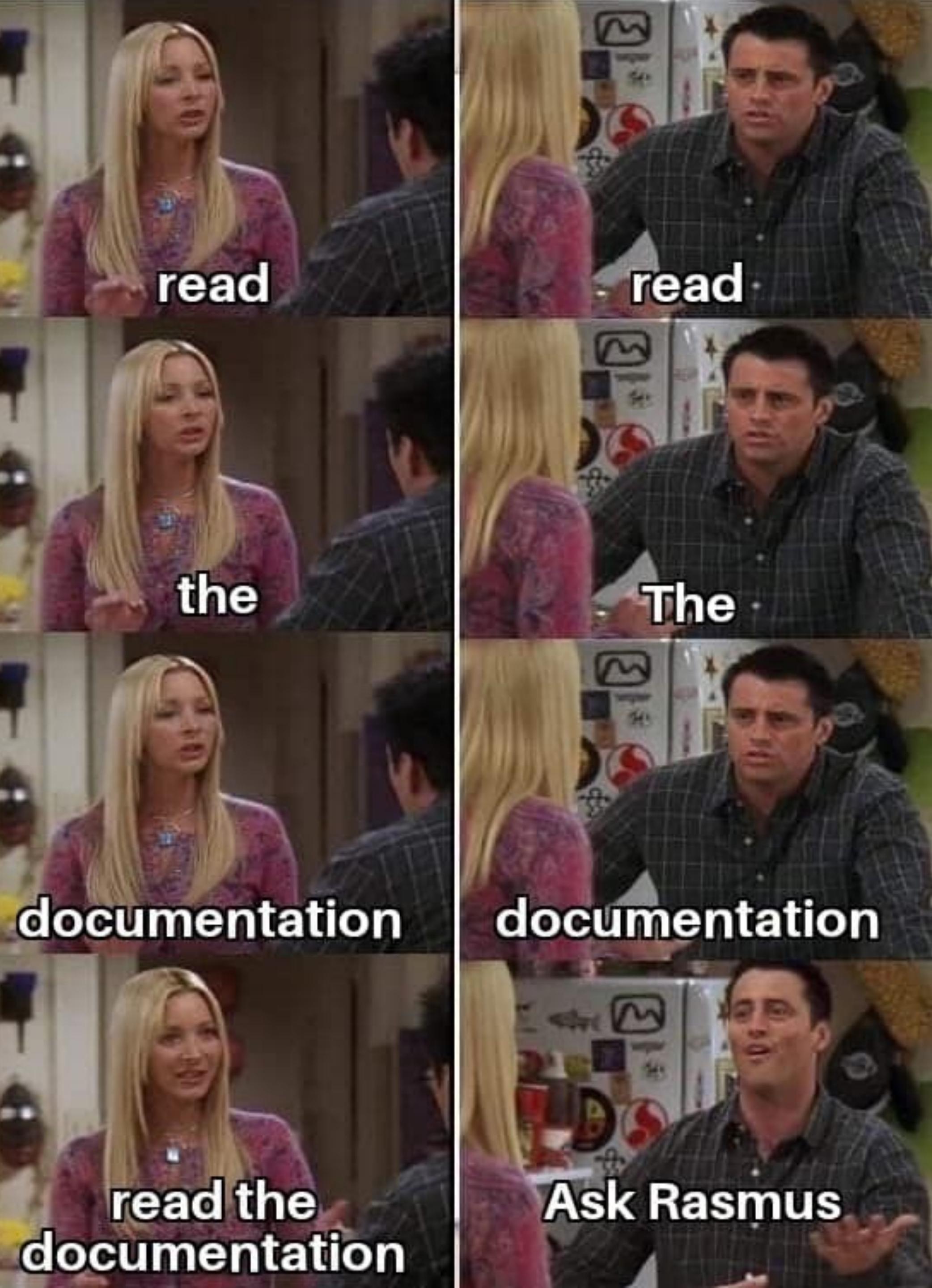
# CRUD App

- Åben din CRUD App løsning fra sidst (også i browseren med Live Server).
- Sluk for JavaScript
- Tænd for JavaScript
- Sluk for CSS
- Tænd for CSS  
Sluk for CSS og JavaScript



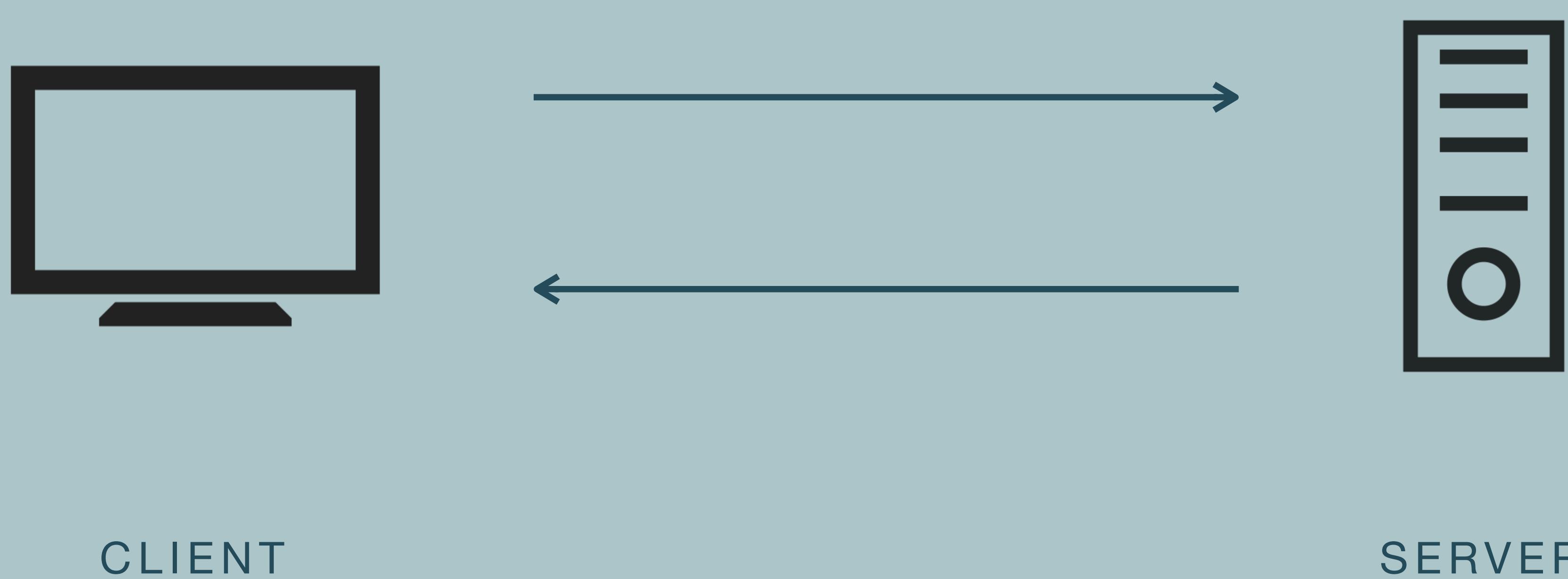
# But RACE?

How is it  
possible? How  
does it work?



# Client-Server Model

Communication between web **clients** and web **servers**.



# Client-Server Model

Communication between web **clients** and web **servers**.

**Browsers**  
Or any type of  
program or device

CLIENT

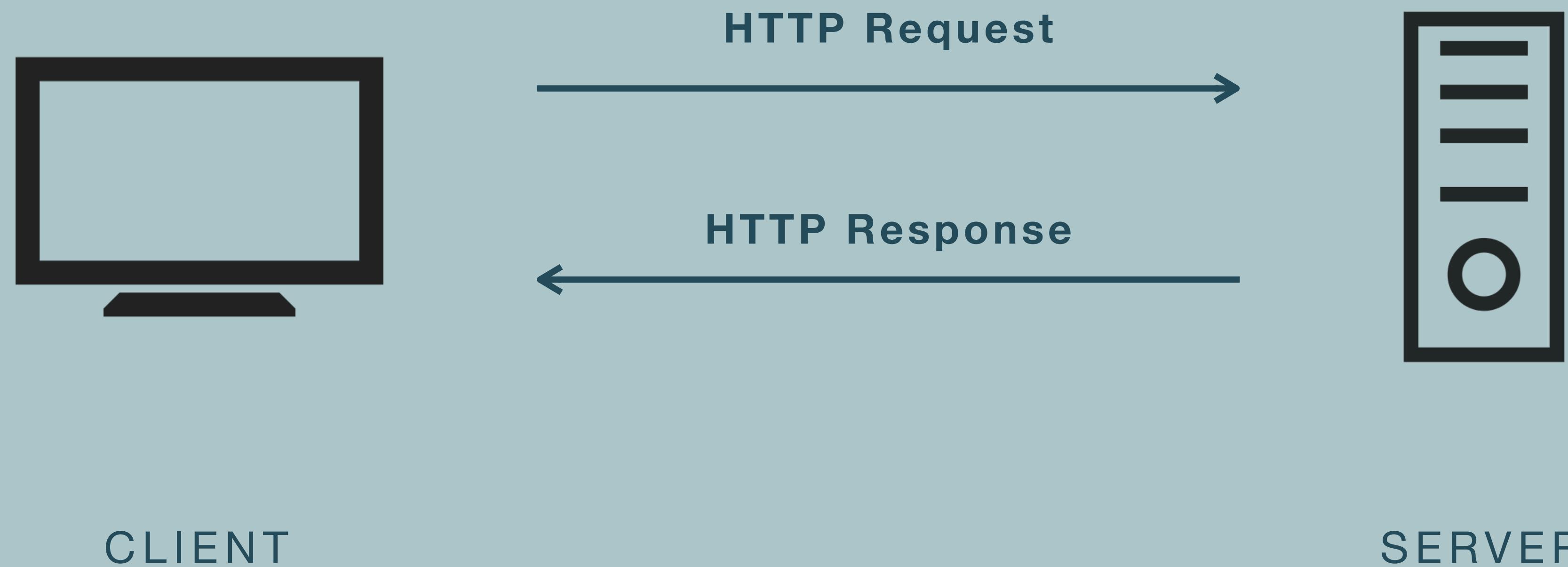


**Cloud Computers**  
Often computers  
in the cloud

SERVER

# Client-Server Model

Communication between web **clients** and web **servers**.



# Hyper Text Transfer Protocol

- A protocol and standard for fetching data, HTML and other resources (text, images, videos, scripts, JSON).
- The foundation of the web.



What is HTTP

Not Secure | w3schools.com/whatis/whatis\_http.asp

HTML CSS JAVASCRIPT SQL PYTHON

# HTTP Request / Response

Communication between clients and servers is done by **requests** and **responses**:

1. A client (a browser) sends an **HTTP request** to the web
2. A web server receives the request
3. The server runs an application to process the request
4. The server returns an **HTTP response** (output) to the browser
5. The client (the browser) receives the response

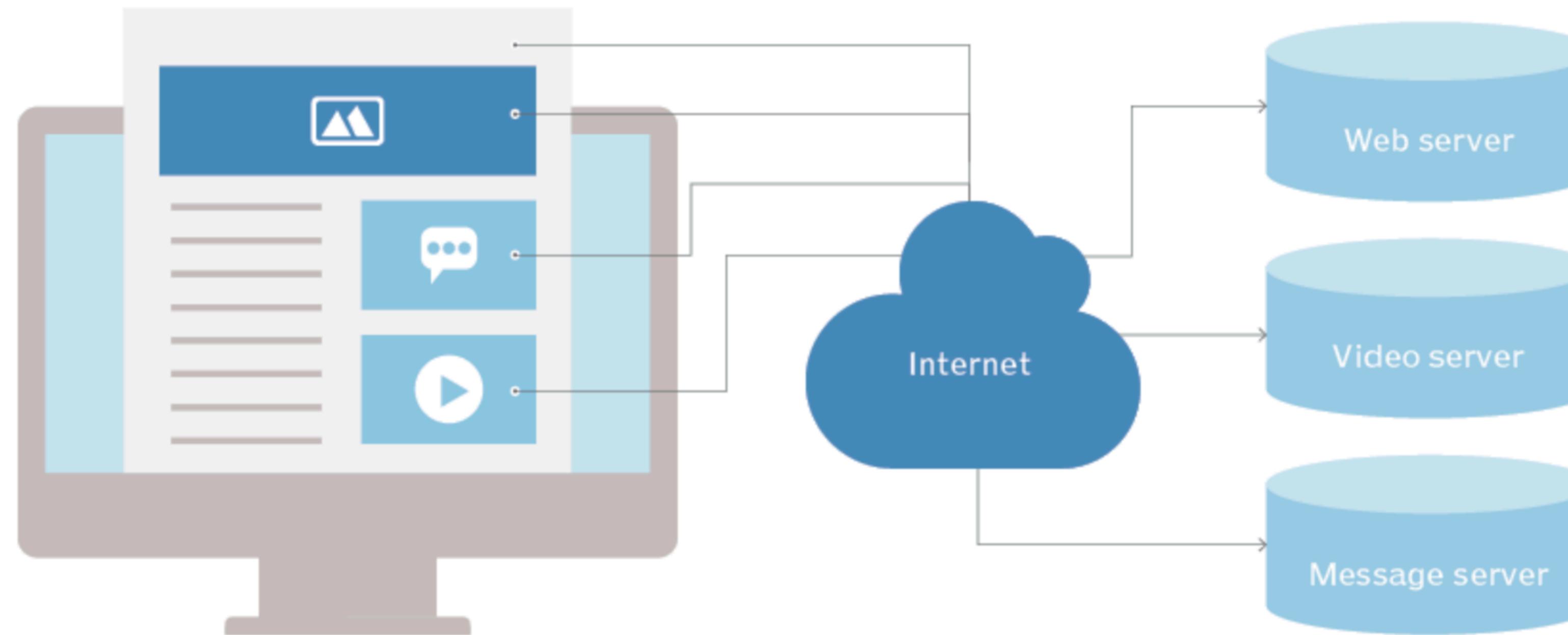
---

## The HTTP Request Circle

A typical HTTP request / response circle:

1. The browser requests an HTML page. The server returns an HTML file.
2. The browser requests a style sheet. The server returns a CSS file.
3. The browser requests an JPG image. The server returns a JPG file.
4. The browser requests JavaScript code. The server returns a JS file
5. The browser requests data. The server returns data (in XML or JSON).

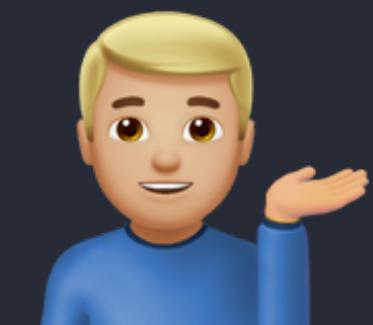
# Want more: How HTTP Works



<https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protocol>

# The Purpose of the Browser

... to read HTML documents and display them correctly.



# Network Tab

The screenshot shows a web browser window with the URL [kompetence.kea.dk/kurser-fag/webudvikling-frontend](https://kompetence.kea.dk/kurser-fag/webudvikling-frontend). The main content area displays information about a 'WEBUDVIKLING FRONTEND' course, including a 'SE HOLDSTART OG TILMELD DIG →' button and a 'LAV DIN EGEN PDF-BROCHURE' section with a 'FØJ TIL PDF' button. A red callout box highlights the 'NY UDDANNELSE: DIPLOM I WEBUDVIKLING' section. The developer tools Network tab is open, showing a list of 46 requests with their details and a waterfall chart.

Name	Status	Type	Initiator	Size	Time	Waterfall
webudvikling-fro...	200	do...	Other	27....	90...	[Waterfall Bar]
quixtrap.css	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
quix.css	200	styl...	webudvi...	(dis...)	1 ms	[Waterfall Bar]
jquery.min.js?77...	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
style.css?77958...	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
jquery-noconflic...	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
jquery-migrate.m...	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
front.css?77958...	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
script.js?779581...	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
core.js?7795818...	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
keepalive.js?779...	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
content.css?779...	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
bootstrap.min.js	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
font-awesome.m...	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
kea-min.js?1.0.28	200	script	webudvi...	(m...)	0 ms	[Waterfall Bar]
logo-main-black...	200	png	webudvi...	(m...)	0 ms	[Waterfall Bar]
css2?family=Po...	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
logo-small.png	200	png	webudvi...	(m...)	0 ms	[Waterfall Bar]
template.css?1.0...	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
finder.css?77958...	200	styl...	webudvi...	(dis...)	2 ms	[Waterfall Bar]
.	200	do...	...	...	...	[Waterfall Bar]

46 requests | 28.7 kB transferred | 1.9 MB resources | Finish: 1.59 s | DOMContentLoaded

- Brug Network-tabben til at undersøge et website (fx [kea.dk](#), [dr.dk](#), [google.dk](#), [dat-js-crud-intro](#) eller et helt andet).
- Åben websitet i Chrome (eller en anden browser).
- Åben Developer Tool (se: [How to open the dev tool in your browser](#)) og gå til Network/Netværk.
- Genindlæs siden imens du står i Network-tabben og undersøg hvilke ressourcer, der bliver hentet.
- Hvilke(n) type ressourcer er der tale om?
- Overvej hvordan ressourcerne hentes.



## Network-tabben

**100** *SECONDS OF*

**HTML**



# HTML, CSS & JavaScript

HTML might be the most important one!

# 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 (tags)
- HTML elements tell the browser how to display the content
- HTML elements can be headings, paragraphs, links, etc.

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)

```
index.html ×  
1  <!DOCTYPE html>  
2  <html>  
3    <head>  
4      <title>Page Title</title>  
5    </head>  
6    <body>  
7      <h1>This is a Heading</h1>  
8      <p>This is a paragraph.</p>  
9    </body>  
10   </html>  
11
```



```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Page Title</title>
5      </head>
6      <body>
7          <h1>This is a Heading</h1>
8          <p>This is a paragraph.</p>
9      </body>
10 </html>
11
```

# This is a Heading

This is a paragraph.

# Anatomy of an HTML document

```
<> index.html ×  
1  <!DOCTYPE html>  
2  <html>  
3    <head>  
4      <title>Page Title</title>  
5    </head>  
6    <body>  
7      <h1>This is a Heading</h1>  
8      <p>This is a paragraph.</p>  
9    </body>  
10   </html>  
11
```

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

```
index.html ×  
1  <!DOCTYPE html>  
2  <html>  
3  <head>  
4    <title>Page Title</title>  
5  </head>  
6  <body>  
7    <h1>This is a Heading</h1>  
8    <p>This is a paragraph.</p>  
9  </body>  
10 </html>  
11
```

```
<html>  
  
<head>  
  
  <title>Page title</title>  
  
</head>  
  
<body>  
  
  <h1>This is a heading</h1>  
  
  <p>This is a paragraph.</p>  
  
  <p>This is another paragraph.</p>  
  
</body>  
  
</html>
```

# The Purpose of the Browser

... to read HTML documents and display them correctly. The browsers use HTML elements (also called tags) to determine how to display the HTML documents and their content.

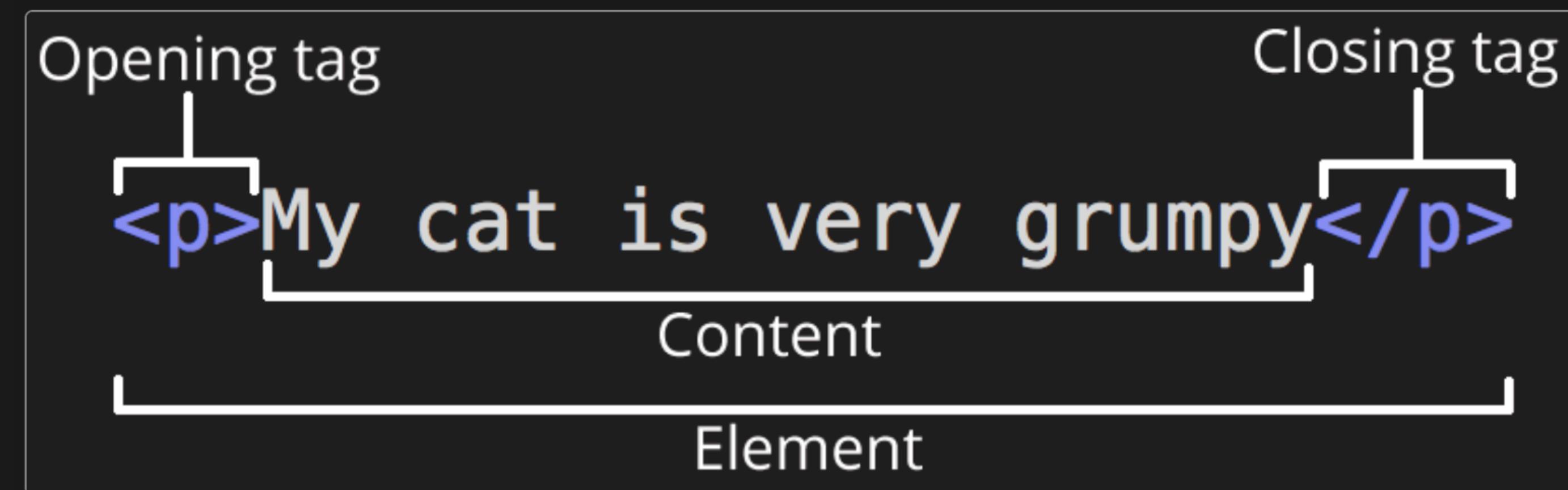
# Anatomy of an HTML element

The screenshot shows a web browser window with a dark theme. The title bar says "Introduction to HTML" and the address bar shows "w3schools.com/html/html\_intro.asp". The navigation bar includes links for HTML (which is highlighted in green), CSS, JAVASCRIPT, SQL, PYTHON, and JAVA. Below the navigation bar, the main content area has a heading "What is an HTML Element?". A text block explains that an HTML element is defined by a start tag, some content, and an end tag, with a code example: <tagname> Content goes here... </tagname>. Another text block states that the HTML element is everything from the start tag to the end tag, with two more code examples: <h1>My First Heading</h1> and <p>My first paragraph.</p>. At the bottom, there is a table titled "Anatomy of an HTML element" with three columns: Start tag, Element content, and End tag.

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	none	none

# Anatomy of an HTML element

Element: opening and closing tag + content



# Void elements

Element: single tag, usually to insert or embed something in a HTML

```

```



[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/Getting\\_started#void\\_elements](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#void_elements)

# HTML Basics

1. Opret et nyt projekt med GitHub Desktop - gerne det projekt du vil bruge til din opskriftsaflevering.
2. Undersøg HTML-elementerne (links til højre).
3. Afprøv HTML-elementer med henblik på anvendelse i opskrift.
4. Du må gerne lade dig inspirere... fx CRUD Appen.

- HTML - The Head Element & Meta tags
- HTML Headings
- HTML Paragraphs
- HTML Text Formatting
- HTML Links
- HTML Images
- HTML Comments
- HTML Lists

# HTML Basics

- HTML - The Head Element & Meta tags: Fx. title, meta description, meta author og andre meta tags.
- HTML Headings: Overskrifter. Husk kun én h1.
- HTML Paragraphs: p-elementer til din tekst.
- HTML Text Formatting: Er der noget som skal være **fed**, *kursiv* eller lignende?
- HTML Links: Links til andre sider
- HTML Images: Billede(r) af din opskrift og eventuelt fremgangsmåde
- HTML Comments: Kommentarer i koden. Øv dig i det.
- HTML Lists: Fx til ingredienser og fremgangsmåde. Undersøg listtyperne.

# Attributes

- Provides additional information and all element can have them.
- Are specified in the start tag
- name/value pairs: `name="value"`

[Attributes - mdn](#)

[HTML Attributes - W3Schools](#)

The screenshot shows a dark-themed web browser window. The title bar reads "Getting started with HTML - Le X". The address bar shows the URL "developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\_to\_HTML/HTML结构性标记语言". The main content area has a header "Attributes" and a sub-section "Elements can also have attributes. Attributes look like this:" followed by a code example. A callout box highlights the word "Attribute" above the code. The code example is `<p class="editor-note">My cat is very grumpy</p>`. Below the code, a text block explains that attributes contain extra information about the element and provides an example of the `class` attribute. At the bottom, there's a section titled "An attribute should have:" with a bulleted list of rules.

## Attributes

Elements can also have attributes. Attributes look like this:

Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```

Attributes contain extra information about the element that won't appear in the content. In this example, the `class` attribute is an identifying name used to target the element with style information.

An attribute should have:

- A space between it and the element name. (For an element with more than one attribute, the attributes should be separated by spaces too.)
- The attribute name, followed by an equal sign.
- An attribute value, wrapped with opening and closing quote marks.

The screenshot shows a web browser window with the title "HTML Attributes". The URL in the address bar is "w3schools.com/html/html\_attributes.asp". The browser interface includes standard controls like back, forward, and search, along with a user profile icon.

The main content area has a dark background with white text. It features a navigation bar at the top with tabs for "HTML", "CSS", "JAVASCRIPT", "SQL", "PYTHON", "JAVA", and "PHP".

## The href Attribute

The `<a>` tag defines a hyperlink. The `href` attribute specifies the URL of the page the link goes to:

**Example**

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

**Try it Yourself »**

You will learn more about links in our [HTML Links chapter](#).

## The src Attribute

The `<img>` tag is used to embed an image in an HTML page. The `src` attribute specifies the path to the image to be displayed:

**Example**

```

```

**Try it Yourself »**

# HTML Attributes

Tilføj attributter til din eksisterende HTML

1. Skim artiklerne til højre.
2. Undersøg din HTML.
3. Bruger du allerede attributter?
4. Kan du tilføje attributter?
5. Har du tilføjet alt til dine img-elementer? href eller target til dine links?
6. Leg med id og class. Kan du gennemskue hvor det giver mening at anvende dem?
7. Du kan eventuelt undersøge hvordan HTML og attributter anvendes i CRUD Appen.

- HTML Attributes
  - HTML class Attribute
  - HTML id Attribute

Tilføj attributter til din eksisterende HTML

1. Skim artiklerne til højre.
2. Undersøg din HTML.
3. Bruger du allerede attributter?
4. Kan du tilføje attributter?
5. Har du tilføjet alt til dine img-elementer? href eller target til dine links?
6. Leg med id og class. Kan du gennemskue, hvor det giver mening at anvende dem?
7. Du kan eventuelt undersøge, hvordan HTML og attributter anvendes i CRUD Appen.

## HTML Attributes

- HTML Attributes
- HTML class Attribute
- HTML id Attribute

A screenshot of a web browser window displaying the MDN Web Docs page for "HTML comments". The browser has a dark mode interface. The title bar shows the page is titled "Getting started with HTML - Le...". The address bar shows the URL "developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\_to\_HTML/Getting\_started#anatomy\_of\_an\_html\_document". The page content starts with a section header "HTML comments" followed by a paragraph explaining the purpose of HTML comments. Below the paragraph, there is an example code block showing how to write an HTML comment.

## HTML comments

HTML has a mechanism to write comments in the code. Browsers ignore comments, effectively making comments invisible to the user. The purpose of comments is to allow you to include notes in the code to explain your logic or coding. This is very useful if you return to a code base after being away for long enough that you don't completely remember it. Likewise, comments are invaluable as different people are making changes and updates.

To write an HTML comment, wrap it in the special markers `<!--` and `-->`. For example:

```
<p>I'm not inside a comment</p>

<!-- <p>I am!</p> -->
```

[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/Getting\\_started#anatomy\\_of\\_an\\_html\\_document](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#anatomy_of_an_html_document)

# HTML Style Guide

[https://www.w3schools.com/html/html5\\_syntax.asp](https://www.w3schools.com/html/html5_syntax.asp)

# HTML Quiz & Exercises

Hvis du skulle komme til at kede dig

The screenshot shows a quiz interface. At the top, a navigation bar includes links for Home, HTML, CSS, JavaScript, SQL, Python, Java, PHP, and Bootstrap. Below this is a sidebar titled "W3Schools Quizzes" with a "Quiz HOME" section containing links for Quiz HTML, Quiz CSS, Quiz JavaScript, Quiz SQL, Quiz MySQL, Quiz Python, Quiz PHP, Quiz jQuery, Quiz Bootstrap, Quiz Bootstrap 4, Quiz Bootstrap 5, Quiz React, Quiz Java, Quiz C, Quiz C++, Quiz C#, Quiz R, Quiz Kotlin, Quiz NumPy, Quiz SciPy, Quiz Pandas, Quiz Django, Quiz XML, Quiz Math, Quiz Git, Quiz TypeScript, and Quiz AWS Cloud Intro. The main content area displays the title "HTML Quiz" and "Question 1 of 40: What does HTML stand for?". Three options are listed: "Hyperlinks and Text Markup Language" (selected), "Hyper Text Markup Language", and "Home Tool Markup Language". A "Next >" button is visible at the bottom right. A small sidebar on the right shows a snippet of code and a "Visual Learner?" video thumbnail.

HTML Quiz

The screenshot shows an exercises page. At the top, a navigation bar includes links for Home, HTML, CSS, JavaScript, SQL, Python, Java, PHP, and Bootstrap. Below this is a sidebar with sections for "HTML Graphics", "HTML Canvas", "HTML SVG", "HTML Media", "HTML APIs", "HTML Examples", and "HTML References". The main content area displays the title "HTML Exercises" and a message: "You can test your HTML skills with W3Schools' Exercises." It features a "Previous" and "Next" navigation bar. Below this, a section titled "Exercises" states: "We have gathered a variety of HTML exercises (with answers) for each HTML Chapter. Try to solve an exercise by editing some code. Get a "hint" if you're stuck, or show the answer to see what you've done wrong." Another section titled "Count Your Score" says: "You will get 1 point for each correct answer. Your score and total score will always be displayed." A large "Start HTML Exercises" button is prominently displayed at the bottom. A small sidebar on the right shows a snippet of code and a "Visual Learner?" video thumbnail.

HTML Exercises

# HTML Semantic Elements

Clearly describes the meaning to both the browser and the developer.

Semantic elements = elements with a meaning

The screenshot shows a web browser window with the title "HTML Semantic Elements" and the URL "w3schools.com/html/html5\_semantic\_elements.asp". The page features a navigation bar with links for HTML, CSS, JAVASCRIPT, SQL, PYTHON, and JAVA. The main content is titled "Semantic Elements in HTML" and includes a note: "Below is a list of some of the semantic elements in HTML." A table lists 15 semantic elements with their descriptions:

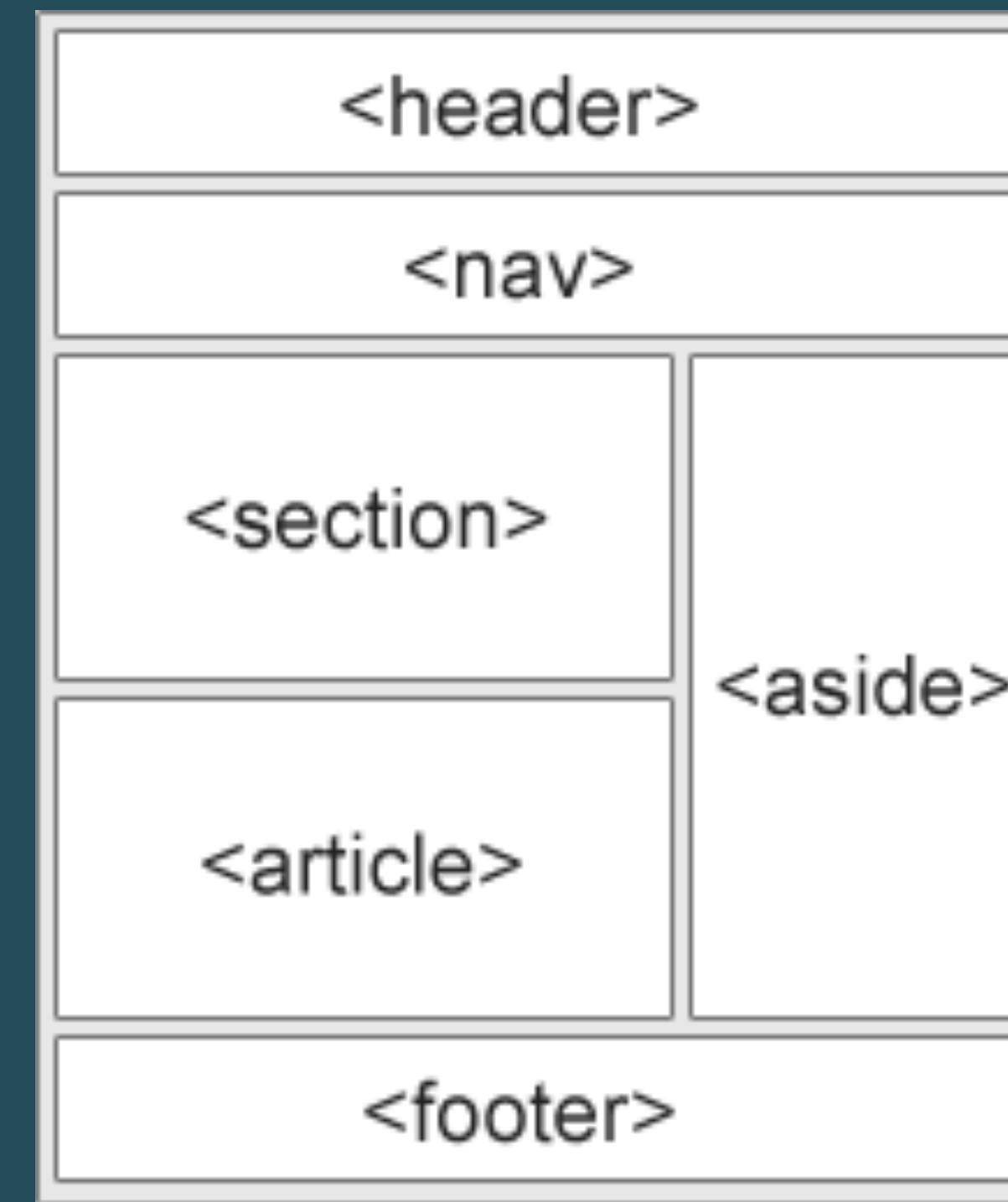
Tag	Description
<code>&lt;article&gt;</code>	Defines independent, self-contained content
<code>&lt;aside&gt;</code>	Defines content aside from the page content
<code>&lt;details&gt;</code>	Defines additional details that the user can view or hide
<code>&lt;figcaption&gt;</code>	Defines a caption for a <code>&lt;figure&gt;</code> element
<code>&lt;figure&gt;</code>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<code>&lt;footer&gt;</code>	Defines a footer for a document or section
<code>&lt;header&gt;</code>	Specifies a header for a document or section
<code>&lt;main&gt;</code>	Specifies the main content of a document
<code>&lt;mark&gt;</code>	Defines marked/highlighted text
<code>&lt;nav&gt;</code>	Defines navigation links
<code>&lt;section&gt;</code>	Defines a section in a document
<code>&lt;summary&gt;</code>	Defines a visible heading for a <code>&lt;details&gt;</code> element
<code>&lt;time&gt;</code>	Defines a date/time

At the bottom, a yellow bar contains the text: "For a complete list of all available HTML tags, visit our [HTML Tag Reference](#)".

# 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>



# 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>
```

## NON SEMANTIC

```
<div class='header'>  
  <div class='section'>  
    <div class='article'>  
      <div class='article'>  
    </div>  
  </div>  
<div class='aside'>  
</div>  
<div class='footer'>
```

## SEMANTIC

```
<header>  
<section>  
<article>  
<article>  
<aside>  
<footer>
```



# <div> tags?

Many websites 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?

Apple x + apple.com

Store Mac iPad iPhone Watch AirPods TV & Home Only on Apple Accessories Support Q □

# iPhone 14 Pro

Pro. Beyond.

[Learn more >](#) [Buy >](#)



# iPhone 14

Big and bigger.

iPhone 14 Plus available starting 10.7

[Learn more >](#) [Shop >](#)

Elements Console » F 1 ⚙️ ⋮ ×

```
><nav id="ac-globalnav" class="js no-touch no-windows no-firefox 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>
<div id="ac-gn-curtain" class="ac-gn-curtain"></div>
<div id="ac-gn-placeholder" class="ac-gn-placeholder">
</div>
<script type="text/javascript" src="/ac/globalnav/7/en_US/scripts/ac-globalnav.built.js"></script>
...
  ><div id="ac-gn-viewport-emitter" data-viewport-emitter-dispatch data-viewport-emitter-state="{"viewport":"medium","orientation":"portrait","retina":true}">...</div> == $0
<script src="/metrics/ac-analytics/2.15.1/scripts/ac-analytics.js" type="text/javascript" charset="utf-8"></script>
  ><main class="main" role="main">...</main>
  ><footer class="js" lang="en-US" id="ac-globalfooter" data-analytics-region="global footer" role="contentinfo" aria-...
...
  -image.no-reduced-motion.no-edge.no-ie.css-mask.inline-video.deskt ...

```

Styles Computed Layout Event Listeners DOM Breakpoints »

Filter :hov .cls + ↻

```
element.style {
}
#ac-gn-viewport-emitter { ac-globalna...built.css:1
  overflow: hidden;
  position: absolute;
  top: 0;
  left: 0;
  width: 0;
  height: 0;
  visibility: hidden;
  z-index: -1;
}
```

Google

google.com

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 ↗ 1 ↘ 1 ⚙ ⋮ X

```
<!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;02vyse:.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>
    ... <div class="L3eUgb" data-hveid="1"> flex == $0
      <div class="o3j99 n1xJcf Ne6nSd">...</div> flex
      <div class="o3j99 LLD4me yr19Zb LS80J">...</div> flex
      <div class="o3j99 ikrT4e om7nvf">...</div>
      <div class="o3j99 qarstb">...</div>
      <div class="o3j99 c93Gbe">...</div>
    </div>
    <div class="Fgvgjc">
      <style data-iml="1634195875111">
        .Fgvgjc{height:0;overflow:hidden}</style>
      <div class="gTMtLb fp-nh" id="lb">...</div>
      <div jscontroller="fKZehd" style="display:none" data-u="0" jsdata="C4mkuf;_;AzQv+I" jsaction="rcuQ6b:npT2md">
      <span style="display:none">...</span>
      <script nonce="4eyH3+1nxzKy0S3HAsRNUA==">...</script>
      <div>...</div>
    </div>
  html body div.L3eUgb
  Styles Computed Layout Event Listeners DOM Breakpoints >
```

Filter :hov .cls + [ ]

```
element.style {
}
.L3eUgb {
  display: flex;
  flex-direction: column;
}
```

The screenshot shows a web browser window with the title "HTML Semantic Elements" from the website w3schools.com. The page content is as follows:

# HTML Semantic Elements

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

On the right side of the browser, a developer tools sidebar is open, showing the DOM structure and CSS styles applied to the page. The DOM tree starts with `<body>` and includes nodes for the main content area, sidebar, and various semantic elements like `<header>`, `<nav>`, and `<article>`. The CSS styles panel shows rules for `html` and `body` elements, including font-family: Verdana, sans-serif; and font-size: 15px;.

# <div> tags?

Use semantic elements as much as possible (!)

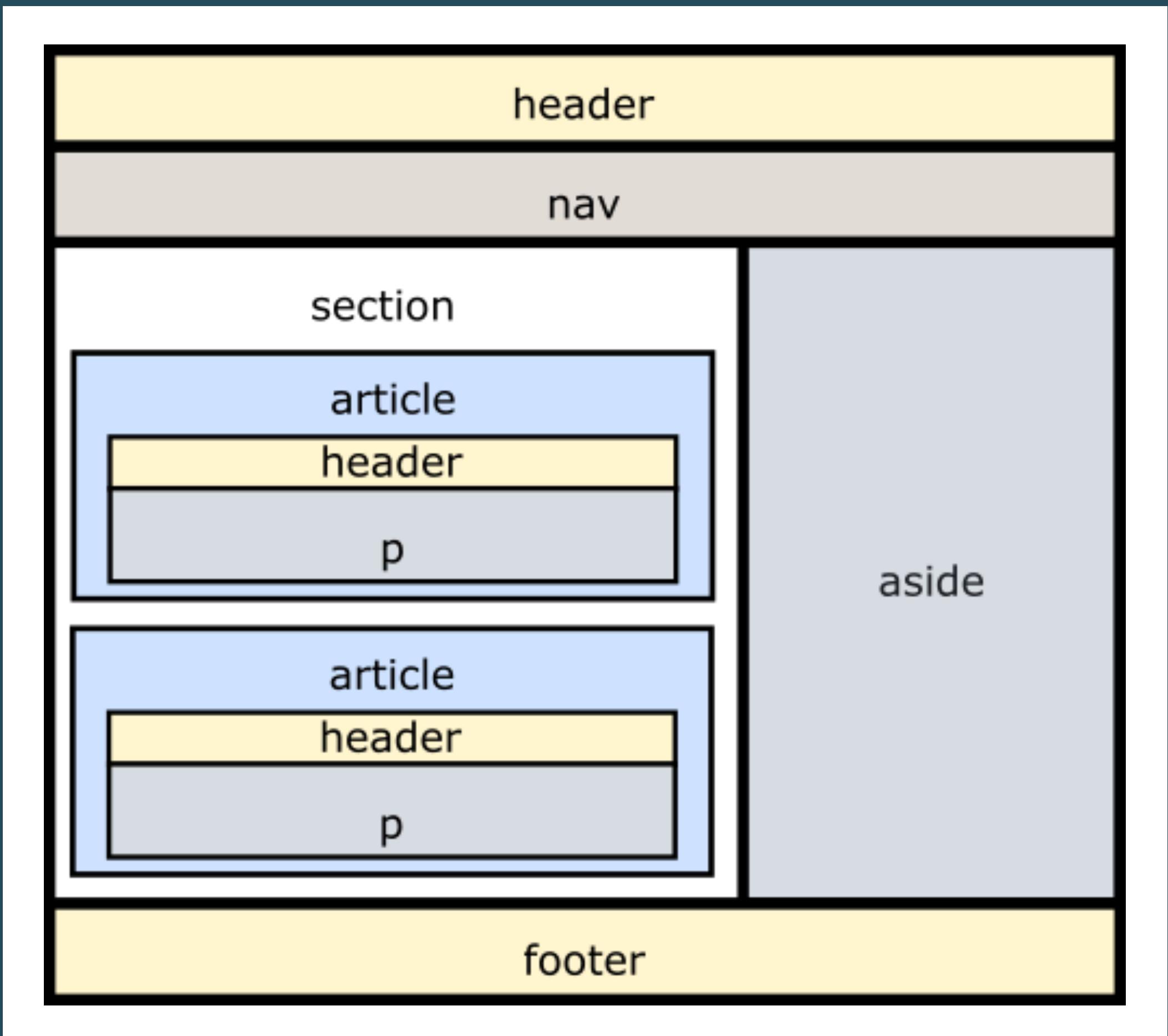
... but don't let it stop you from building

something amazing 

- RACE

# HTML Semantics Exercise

1. Create or use an existing project with an `index.html`.
2. Create a structure matching the image to the right. Use HTML Semantic tags.
3. Add content to the structure, like headings, links, text (paragraphs), images and contact info. Explore and use [HTML Element Reference](#) & [HTML Semantics](#).



# Opskrift

Aflevering: Lav din yndlingsopskrift i HTML og CSS

Delopgaver:

- **Opskrift 1: HTML, struktur og indhold**
- **Opskrift 2: CSS - basal styling**
- **Øvelse: Deploy din opskrift til GitHub Pages**
- **Opskrift 3: CSS Grid layout**

Fronter

kea-fronter.itslearning.com/ContentArea/ContentArea.aspx?LocationID=5796&LocationType=1&...

Startside Rum Dashboards og projektrum Mere ▾

Rasmus Cederdorff

Dat23v2 - 1. sem Oversigt Planer Ressourcer Deltagere Mere ▾ Tilføj

Afleveringsopgave Rediger Skjul ▾

Opskrift 1: HTML, struktur og indhold

```
graph TD; A[<header>] --- B[<nav>]; B --- C[<section>]; C --- D[<article>]; C --- E[<aside>]; D --- F[<footer>]
```

I denne opgave skal du lave den overordnede struktur og finde indhold (tekst og billeder) til din yndlingsopskrift. Du skal anvende HTML-elementer til at beskrive din struktur og dit indhold.

Denne opgave er første del af [Lav din yndlingsopskrift i HTML og CSS](#).

Krav

Der er et krav, at du anvender følgende:

- HTML og [HTML5 Semantiske elementer](#) (tags) til at beskrive din struktur.
- Tekst og billeder. Din opskrift skal som minimum bestå af:
  - En kort beskrivelse af opskriften
  - En eller flere lister med ingredienser knyttet til opskriften
  - En fremgangsmåde, herunder tilberedningstid
  - Tekst, billeder og/ eller andre visuelle elementer som understøtter overskriften
- En skitse af din opskrift – gerne med papir og blyant så du har en “plan” over din HTML-struktur

Anbefalet procedure

1. Find den opskrift og det indhold (tekst, billeder, fremgangsmåde, mm.) du vil anvende i din løsning. Hvis du finder eksisterende materiale fra webbet, skal du være opmærksom på, at du har tilladelse til at anvende materialet (tekst og billeder).
2. Skisér dit layout – gerne med papir og blyant. Overvej dit layout og husk at opbygning skal ende ud med at være i et grid.

# Opskrift 1: HTML, struktur og indhold

1. Skitsér din opskrift med papir og blyant
2. Opret et nyt projekt med GitHub Desktop
3. Når du strukturerer din HTML: Begynd med den overordnede struktur og overvej semantiske elementer. Se evt her.
4. Følg anbefalet procedure i afleveringsopgave.
5. Brug dagens slides og materiale til at sikre et velstruktureret HTML-dokument med dit ønskede indhold.

Fronter

kea-fronter.itslearning.com/ContentArea/ContentArea.aspx?LocationID=5796&LocationType=1&...

Startside Rum Dashboards og projektrum Mere

Dat23v2 - 1. sem Oversigt Planer Ressourcer Deltagere Mere Tilføj

Afleveringsopgave Rediger

Opskrift 1: HTML, struktur og indhold

<header>

<nav>

<section>

<article>

<aside>

<footer>

I denne opgave skal du lave den overordnede struktur og finde indhold (tekst og billeder) til din yndlingsopskrift. Du skal anvende HTML-elementer til at beskrive din struktur og dit indhold.

Denne opgave er første del af [Lav din yndlingsopskrift i HTML og CSS](#).

Krav

Der er et krav, at du anvender følgende:

- HTML og [HTML5 Semantiske elementer](#) (tags) til at beskrive din struktur.
- Tekst og billeder. Din opskrift skal som minimum bestå af:
  - En kort beskrivelse af opskriften
  - En eller flere lister med ingredienser knyttet til opskriften
  - En fremgangsmåde, herunder tilberedningstid
  - Tekst, billeder og/ eller andre visuelle elementer som understøtter overskriften
- En skitse af din opskrift – gerne med papir og blyant så du har en “plan” over din HTML-struktur

Anbefalet procedure

1. Find den opskrift og det indhold (tekst, billeder, fremgangsmåde, mm.) du vil anvende i din løsning. Hvis du finder eksisterende materiale fra webbet, skal du være opmærksom på, at du har tilladelse til at anvende materialet (tekst og billeder).
2. Skisér dit layout – gerne med papir og blyant. Overvej dit layout og husk at opbygning skal ende ud med at være i et grid.

# Semantic HTML & SEO(indexability)

Why tell the browser what the HTML elements represents?

... to improve

# Search Engine Optimization

Tell what your content is about and  
by hierarchy, tell what's important

# Why Semantic HTML?

“Because semantic HTML uses elements for their given purpose, it’s easier for both people and machines to read and understand it.”

“Semantic HTML means using correct HTML elements for their correct purpose as much as possible. Semantic elements are elements with a meaning; if you need a button, use the `<button>` element (and not a `<div>` element).”

### Semantic

```
<button>Report an Error</button>
```

### Non-semantic

```
| <div>Report an Error</div>
```

**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.

1. Headings <h1> to <h6>
2. Document Structure
3. Textual Meaning (Bold, Italics, Highlight)
4. Media Type
5. Correlation Tags

5 Ways to Write Semantic HTML and Improve Webpage SEO and Accessibility

[Read more here](#)

# Commit og push til GitHub

Git, GitHub & GitHub Desktop

# Git, GitHub & GitHub Desktop

Version Control, Collaboration & how to manage  
your code projects

# Git

Version Control System - A piece of Software.

A system that tracks all changes in your code and files.

# GitHub

Cloud-based Hosting Service to host your Git Repositories (projects). Your repositories are stored on an online server.

GitHub uses git and helps people like you and me to collaborate on projects. On GitHub you can discover other users and repositories.

# GitHub Desktop

An Application on your computer.

A GUI to manage your project with Git and GitHub. In other words, GitHub Desktop allows you to run Git and manage your projects in a desktop app with a graphical interface, instead of typing git commands.

# Commit your changed

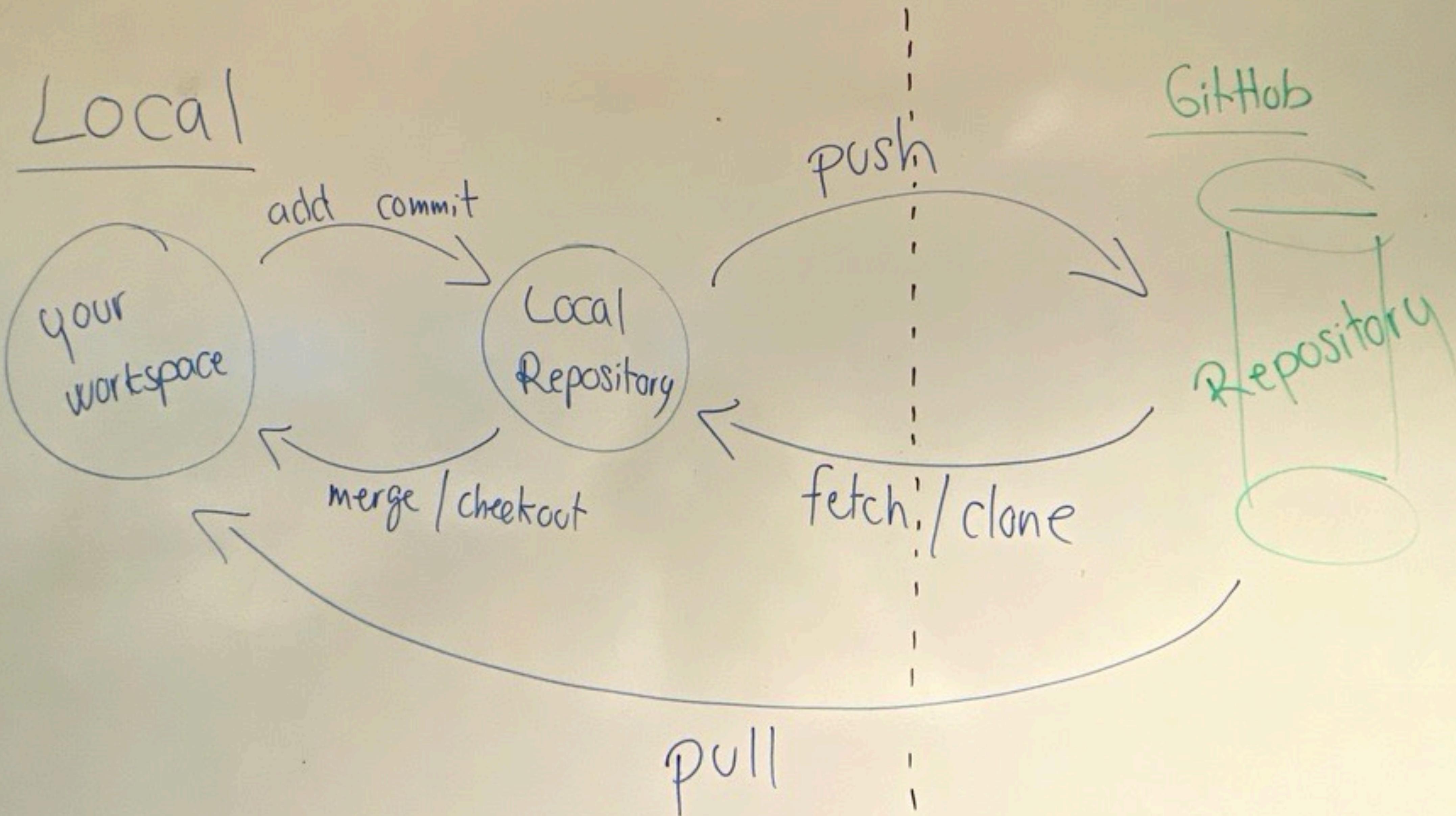
git commit: Snapshot of your repository. Contains the changed files and a message. Commits are “saving points” making it possible to keep track of code changes and go back “in history”.

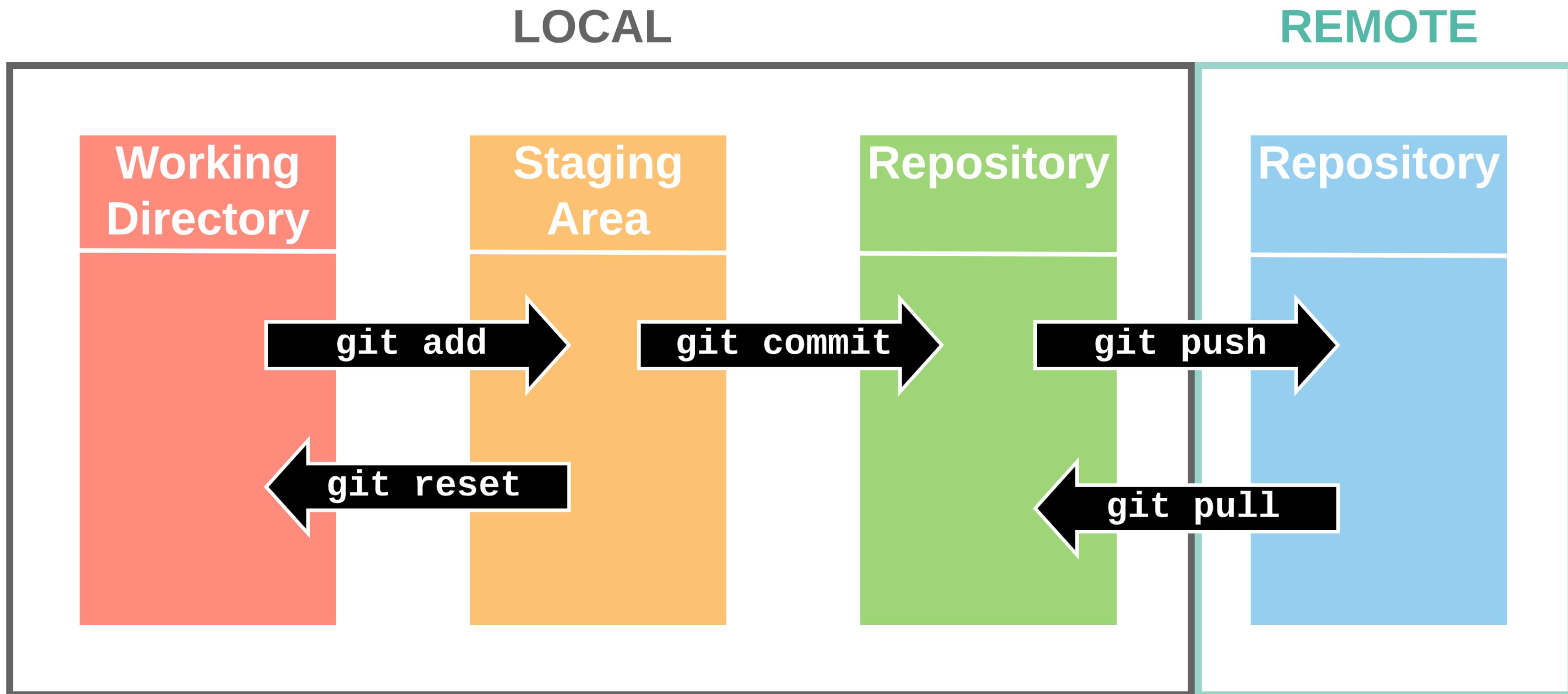
Se punkt 8 i Opret et nyt projekt med GitHub Desktop

# Push your commit

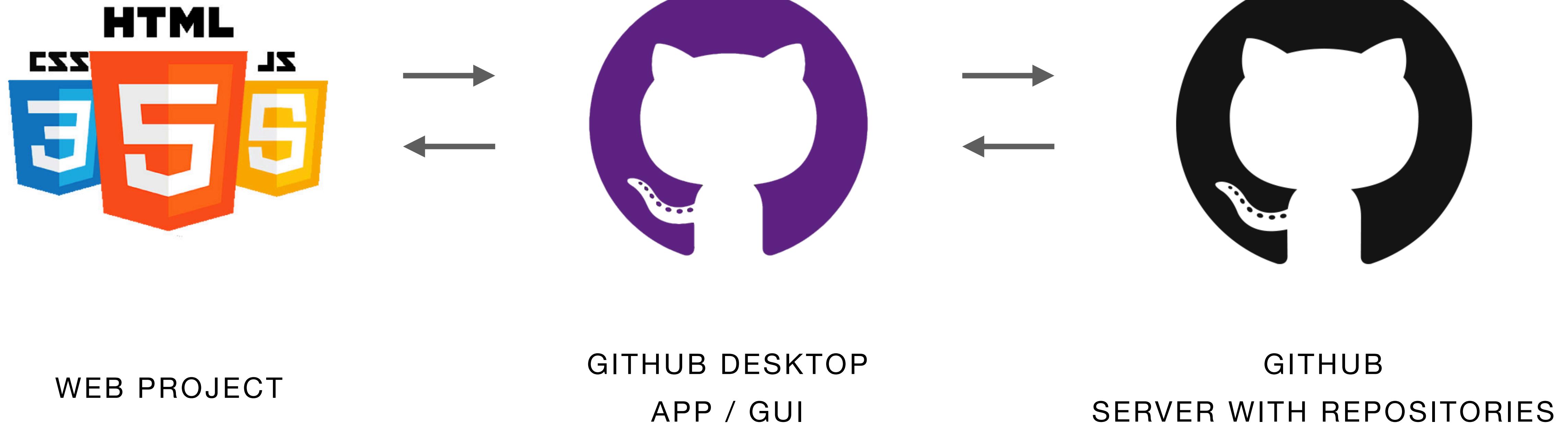
git push: Uploads (pushes) your local commits to the remote repository - pushes your (local) commits to your repository on GitHub (online).

Se punkt 8 i Opret et nyt projekt med GitHub Desktop





# Git



# Hvordan er “formen”?

Tak for i dag

Datamatiker

**kea**  
KØBENHAVNS ERHVERVSAKADEMI