

# Eksamens og repetition

1. semester – sommer 2023





Promise from setTimeout

```
const afterSomeTime = (time) => new Promise(resolve => {
  setTimeout(() => {
    resolve(true);
  }, time);
};

const callAfterSomeTime = (callback, time) => afterSomeTime(time).then(callback);

callAfterSomeTime(() => console.log('Hello after 1500ms'));

const getData = async () => fetch('https://jsonplaceholder.typicode.com/users')
  .then(response => response.json())
  .then(data => data[0])
  .catch(error => console.error(error));

document
  .querySelector('#name')
  .addEventListener('click', function() {
    const name = document.querySelector('#name').value;
    const user = await getData();
    const posts = await fetch(`https://jsonplaceholder.typicode.com/posts?userId=${user.id}`);
    const comments = await fetch(`https://jsonplaceholder.typicode.com/comments?postId=${posts.id}`);
    //display comments on DOM
  });
}

// send to backend
const user = await fetch(`https://jsonplaceholder.typicode.com/users?name=${name}`);
const posts = await fetch(`https://jsonplaceholder.typicode.com/posts?userId=${user.id}`);
const comments = await fetch(`https://jsonplaceholder.typicode.com/comments?postId=${posts.id}`);
//display comments on DOM
```

# Dages Emner

- Præsentation af eksamensformen (repetition)
- Eksamensplan (WISEflow)
- Præsentation af eksamensemner
- Repository med environments – til fork
- Eksamensøvelser
- Prøveeksamen
- Tips og tricks til eksamen (struktur, genvejstaster, kendte mønstre)
- Emner der skal repeteres

# Eksamensform

1. semester – sommer 2023

med  
kode-tips!

# Først – hvad siger dokumentationen?

- Studieordningen er vores “dokumentation”!
- Men det er den dårligste dokumentation, vi har set indtil videre.
- Husk dog at studieordningen er altid højeste autoritet – uanset hvad vi synes, hører, kommer til at sige, eller lige husker!

when you ask Rasmus for help and he says "Read documentation"





my firebase doesn't  
work. Can you help me?



read the  
documentation



But...

read the  
documentation

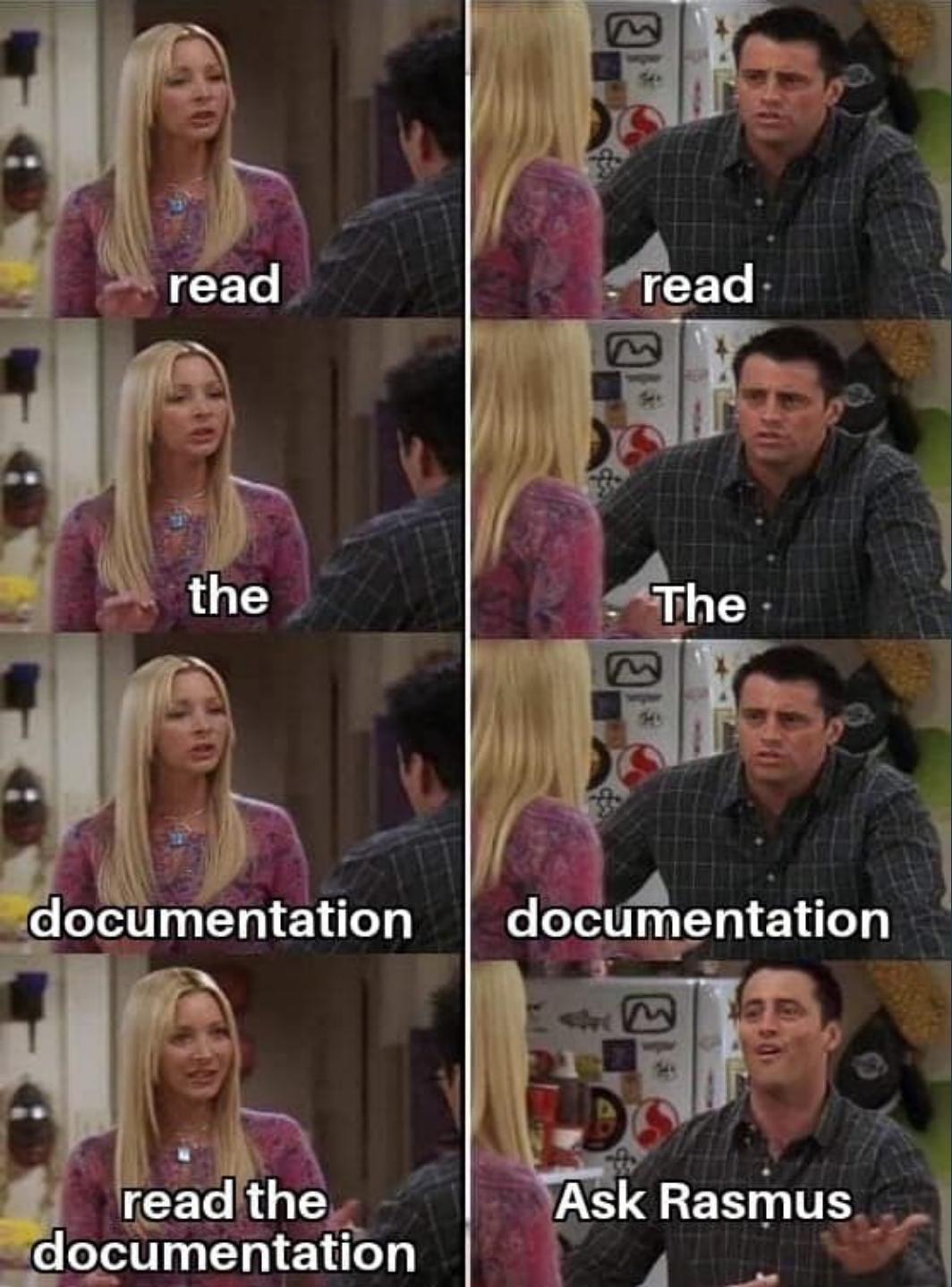


## Læs om 1. delprøve

1. delprøve: *Ekstern individuel mundtlig eksamen i fagelementet Programmering og bedømt efter 7-trins skalaen. Delprøven vægter 25% af den samlede førsteårs karakter.*

[...]

1. delprøve: *20 minutters mundtlig eksamen inkl. voting baseret på spørgsmål i læringsmålene*



Og hvad  
betyder  
det så?

# PETL & RACE to the rescue!

- Eksamen dækker **1. semester programmering**: 15 ECTS.
- Det er en **individuel eksamen** – uden aflevering af produkt eller rapport.
- Eksamen er **mundtlig**.
- For at blive indstillet til eksamen, skal man have **godkendt bundne forudsætninger** = afleveret 5 projekter i løbet af semesteret.
- Eksamen varer i 20 minutter – de 5 minutter går til voting og karaktergivning, så reelt 15 minutters eksamination.
- Alle hjælpemidler tilladt - selv ChatGPT. Husk dog på at tiden er knap!
- Karakteren er på 7-skalaen, og tæller 25% af den samlede karakter for det første år (2.semester eksamen tæller de resterende 75%). Det er derfor 15 ECTS ud af 60 ECTS (1. semester == 30 ECTS).

# Eksamen er med ekstern censor

- Censor for V1 er Mohammad Homayoon Fayezi der underviser på Erhvervsakademi Zealand i Roskilde.
- Censor for V2 er Per Jensen.
- Det betyder normalt ikke noget hvem censor er til disse korte eksamener. Censor er der for at sikre, at I får en fair behandling af eksaminator.
- Husk at eksamen handler om **jer**. Det er jeres show, og I bestemmer, hvad der skal vises, fortælles og demonstreres!

# Eksamensform

- **20 minutter** i alt (+ de 5 minutters “Teknisk forberedelse”):
  - **5 minutter:** “Teknisk forberedelse” før den reelle eksamen starter - stil op og træk spørgsmål.
  - **12-13 minutter:** Du løser eksamensopgaven, og ja live coding 😎
  - **2-3 minutter:** Afsluttende spørgsmål hvis der er tid
  - **5 minutter:** Votering og karaktergivning

# Eksamensform

- **5 minutter før start**
  - Kom ind i lokalet, hils på censor og eksaminator.
  - Sæt computer til projektor og få teknikken til at spille.
  - Hav VS Code startet op.
  - Husk at have fork'et repository med alle “environments”: <https://github.com/cederdorff/dat-js-exam-spring-2023>
  - Hav styr på hvordan du åbner en specifik mappe i “environments” - og kun den mappe!
  - Træk et spørgsmål 

# Eksamensform - fortsat

- **Start på eksamen**
  - Læs spørgsmålet hurtigt igennem, så du danner dig et overblik. Læs det gerne højt.
  - Stil spørgsmål hvis der er noget i formuleringen, der undrer dig.
  - Find og åbn den “environment”-mappe, som er angivet på eksamensspørgsmålet.
  - Start index.html-filen med Live Server, så du er klar!

# Eksamensform - fortsat

- Eksamensspørgsmålet består af en **1-2-3 model:**
  - Delspørgsmål 1
  - Delspørgsmål 2
  - Delspørgsmål 3

# Eksamensform - fortsat

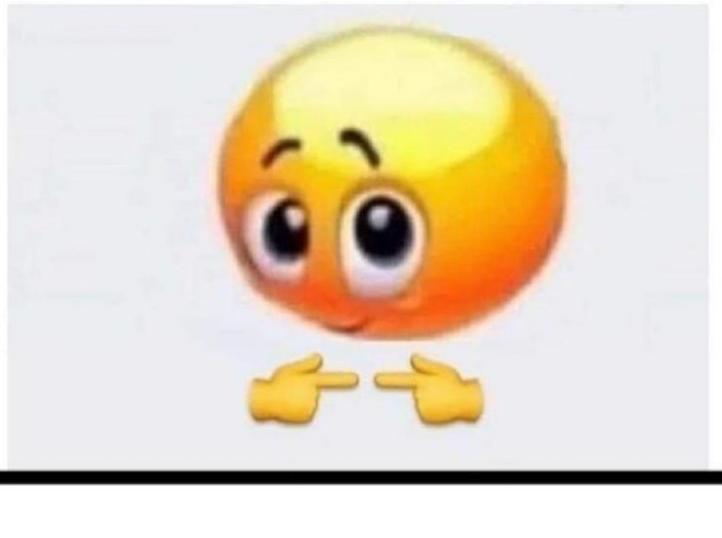
- **Løs delspørgsmål 1**
  - Består typisk af noget med at fetch'e en liste og klargøre noget data til brug i de følgende delspørgsmål.
  - Kan også være noget med at tilføje objekter til et eksisterende array.
- **Løs delspørgsmål 2**
  - Ofte noget med at vise listen fra delspørgsmål 1, dvs DOM-manipulation.
  - Det *kan* være beskrevet hvad, og hvordan data skal vises.
  - Og der kan være filtrering og sortering (fx vis kun objekt med prop der har x værdi).
  - Kan indeholde betingelser (if-sætninger) og loops samt form og submit.
- **Løs delspørgsmål 3**
  - Typisk noget med at filtrere, sortere, validere eller manipulere med eksisterende data.
  - Kan indeholde betingelser (if-sætninger), loops og DOM-manipulation.

# Eksamensform - fortsat

- **12-13 minutter efter start**
  - Hvis der stadig er tid tilbage, og/eller du ikke har kunnet få al koden til at virke, kan det være at censor og eksaminator stiller nogle uddybende forståelsesspørgsmål.
- **15 minutter efter start**
  - Du bliver sendt ud af døren – husk at tage din computer, taske og alle øvrige ting med ud. Og gå helt ud på gangen og vent.
- **Inden 20 minutter efter start**
  - Du bliver kaldt ind igen, og får en karakter på 7-skalaen. Der vil næppe være tid til en uddybning af karakteren.

# Kode-tips

will you be async to my await?



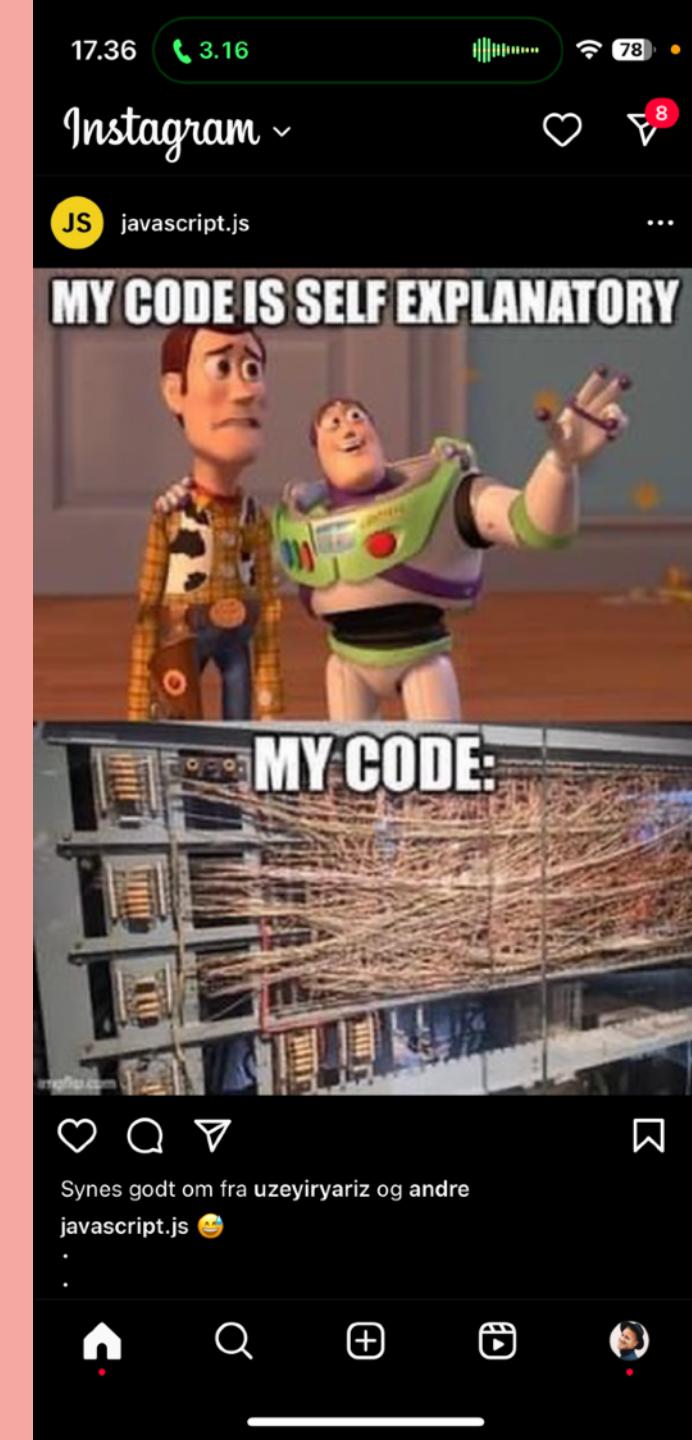
- Der står aldrig i opgaven, at der skal laves en `window.addEventListener("load", ...)` – men gør det alligevel.
- Det er ikke sikkert, at der står specifikke JS-termer som fx `fetch`, `function`, `await`, `const`, `let`, osv. som skal have for at løse en opgave.
- Det er heller ikke sikkert at der står præcist hvor mange funktioner du skal inddøle koden i.
- Men lav dem endelig hvis det giver mening og matcher med “best practice”, og hvad du har lært.
- Der står heller ikke (nødvendigvis) hvorvidt variabler eller funktioner skal være globale eller lokale, eller om en funktion skal returnere noget eller ej - men sorg for alligevel at bruge den ”korrekteste” form.
- Det er aldrig nødvendigt at oprette flere filer, end der er angivet i opgaven. Brug derfor ikke tid på at lave flere filer eller moduler, medmindre det er angivet i opgaven.

# Kode-tips

- En væsentlig del af eksamen er at vælge og bruge datatyper for parametre og returværdier – så der står aldrig hvilken type du skal bruge, men du forventes at argumentere for den du vælger.
- Sørg så vidt muligt for at en funktion altid returnerer noget, i stedet for at skrive det ud (`console.log`) – og skriv det i stedet ud i start-/`initApp`-funktionen med et kald til den pågældende funktion.
- Medmindre der specifikt står i opgaven at en funktion skal `console.log`'e noget!
- Du må selvfølgelig gerne have test-udskrifter (`console.logs`) undervejs i din kode – bare husk at sige at du gør det for at afprøve om den virker.

# Tips

- Husk løbende at vise og forklare, hvad du gør.
- “Tænk højt”. Men det er selvfølgelig ok at sige: “jeg koder lige det her, og så fortæller jeg”.
- Vis fx at du kan kontrollere variabler og resultater løbende med en `console.log` igennem dit arbejde med at implementere koden.
- Husk at du gerne må stille spørgsmål og indgå i dialog med eksaminator (og censor). Det trækker ikke nødvendigvis ned.
- Træn, øv, kode, programmer og gentag! Øv dig i lignende opgaver, som eksemplerne [her](#).
- Øv dig I at Google og finde det rigtige i [JavaScript.info](#) og W3Schools.
- Tag en dyb indånding og pust ud. Det skal nok gå, og det er ikke dit liv, der står på spil (heller ikke din højre arm 🤪)!



# FAQ

- **Bliver man eksamineret ud fra semesterprojektet?**
  - Nej, det er særskilte eksamensspørgsmål.
- **Hvem bliver man eksamineret af? Er begge undervisere (Rasmus og Peter) til stede + censor eller kun en?**
  - DAT23V2 bliver eksamineret af RACE og en censor. DAT23V1 af PETL og en censor.
- **Hvordan bliver tiden fordelt under eksamen?**
  - Eksamenen varer 20 min inklusiv votering og feedback, dvs at du er "på" i 15 min.
- **Hvor længe har jeg til at fremlægge?**
  - Se ovenstående - du er på i 15 min.
- **Er der information, man får af vide på forhånd inden eksamen, der kan hjælpe med at præcisere ens eksamenslæsning og ikke bare terpe alt hvad vi har lært de sidste 4 måneder?**
  - Reelt alt, men eksemplerne på eksamensspørgsmålene giver et billede af fokus: objekter, arrays, JSON, fetch, funktioner, DOM-Manipulation og det der støtter disse.  
Øv dig med alle øve-eksamensspørgsmålene og environments.
- **Trækker man et emne og/eller spørgsmål?**
  - Du trækker et eksamensspørgsmål, som består af tre delspørgsmål. Se evt. eksemplerne.
- **Har man tid til forberedelse efter man har trukket?**
  - Nej. Der er (op til) fem minutter til at stille op og sikre at computeren spiller før du trækker spørgsmålet. Så snart du ser spørgsmålet, starter dine 15 minutter.



## Øve eksamens-repo

The screenshot shows a GitHub repository page for 'petlatkea/Dat-js-exam-exercises'. The repository is public and contains 1 branch and 0 tags. The code tab is selected, showing a commit history with four commits. The commits are:

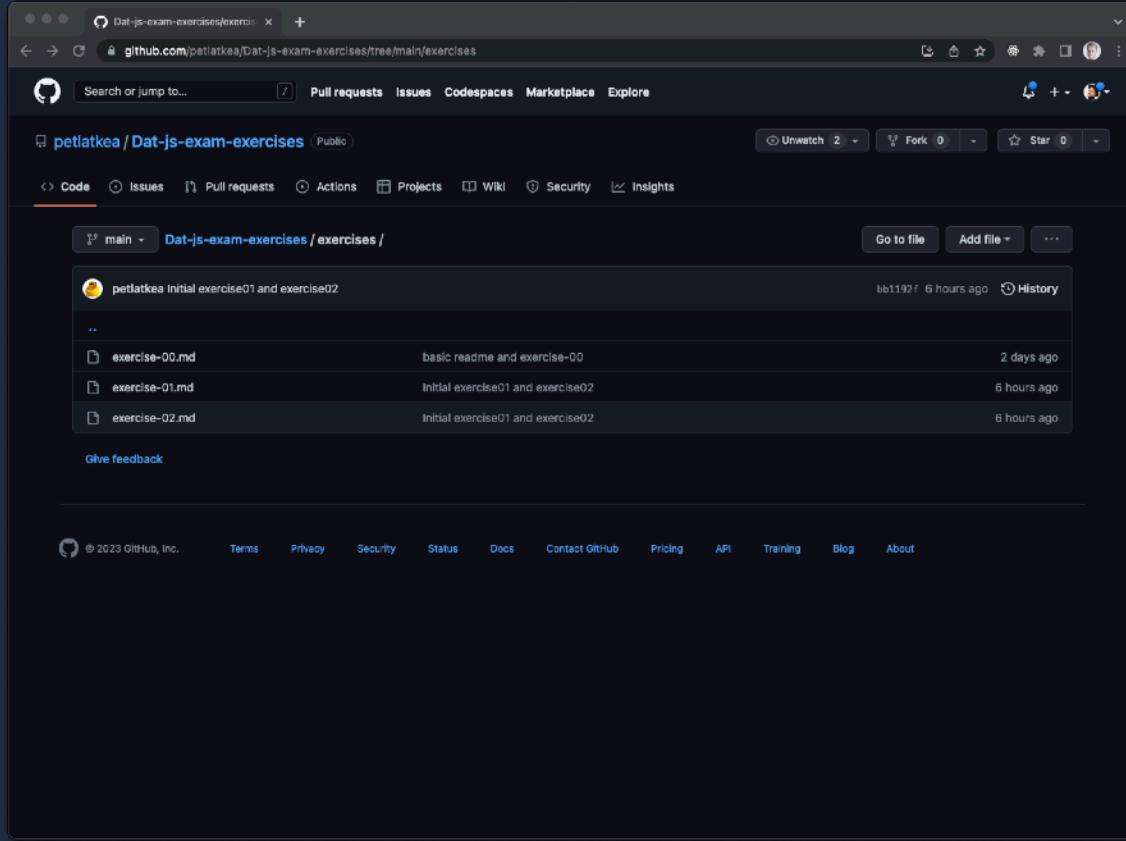
- petlatkea Initial exercise01 and exercise02 (bb1192f, 5 hours ago)
- environments Initial exercise01 and exercise02 (5 hours ago)
- exercises Initial exercise01 and exercise02 (5 hours ago)
- .gitignore first commit - environment-01 (2 days ago)

The README.md file is also listed. The repository description is "Repository for Dat.js exam-exercises." It has 0 stars, 2 watching, and 0 forks. There are no releases or packages published. The languages used are HTML (75.2%), CSS (23.2%), and JavaScript (1.6%). Suggested Workflows are based on the tech stack.

# Lad os prøve!

<https://github.com/petlatkea/Dat-js-exam-exercises>

## Øve eksamens-repo

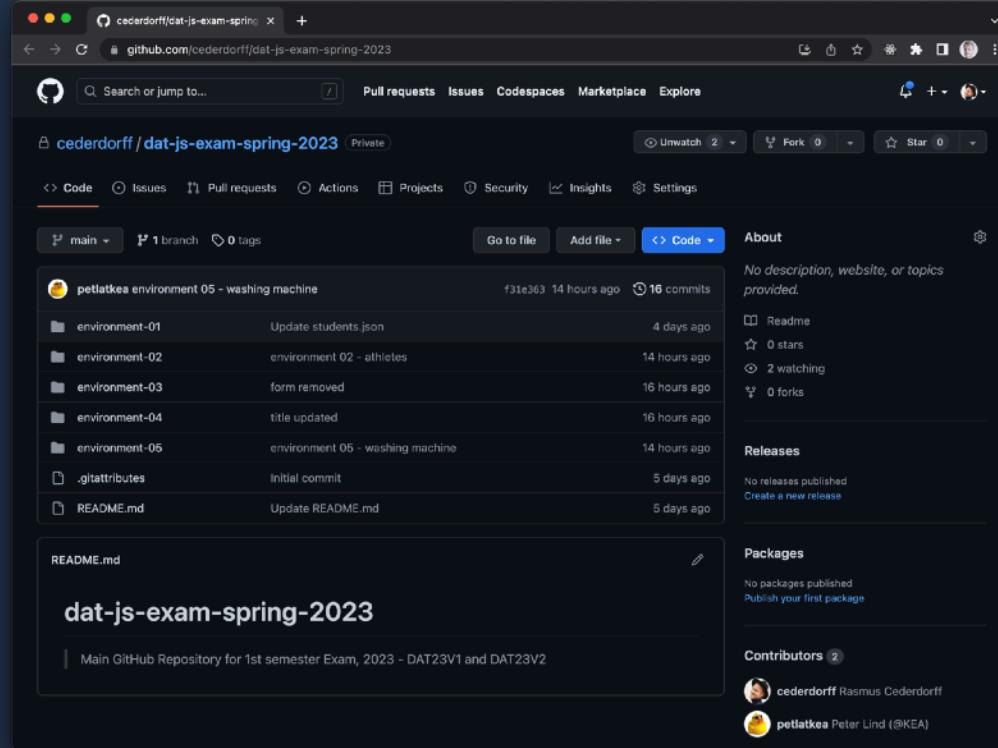


# Environments

- Indeholder altid filerne:
  - index.html,
  - style.css
  - script.js
- Kan indeholde JSON-fil.
- Starter-HTML, hvor CSS og JS er forbundet.
- HTML-elementer som kan anvendes.
- Minimal styling.
- JS indeholder kun “use strict”;

<https://github.com/petlatkea/Dat-js-exam-exercises/>

## Det rigtige eksamens-repo



# Fork “det rigtige” eksamens-repo

<https://github.com/cederdorff/dat-js-exam-spring-2023>