

Fik I lavet loop-view med dyr
og stilet den med css-grid?

frø



Type: amfobie

Levested: vandhullet

gris



Type: pattedyr

Levested: grisefarmen

ræv



Type: pattedyr

Levested: skoven

sild



Type: fisk

Levested: havet

krokodille



Type: krybdyr

Levested: floden

json-filer

Agenda

1. JSON-filer
2. Hent json-data ind fra fil
3. Asynkrone events i javascript (promises)
4. Data fra google sheets til json
5. Plugin til visning af json-filer i Chrome
6. Opgave: persondata
7. Intro til temaopgaven

JSON-filer

JSON = **J**ava**S**cript**O**bject**N**otation

Fil med efternavn/extension: **json** - fx: **personer.json**

Et objekt, eller et array af objekter kan gemmes i en **json-fil**.

```
let undervisere=[
  {"navn":"Helle Frederiksen","email":"helf","github": "helefred"},
  {"navn":"Klaus Mandal","email":"klmh", "github": "MondaleMondale"},
  {"navn":"Martin Bregnhøi", "email":"mabe", "github": "martinbregnhøi"},
  {"navn":"Kamilla Viktor", "email":"kvi", "github": "kvikea"}
];
```

kan pastes ind i ny fil og gemmes under navnet fx. **personer.json**.

json-filer **kan hentes** ind i html-dokumenter med javascript

JSON is based on JavaScript object literals The literal value of an object exposes the properties or attributes in a way which we can see (and read). By Lindsay Bassett: February 18, 2016: <https://www.oreilly.com/ideas/json-is-based-on-javascript-object-literals>

What is JSON? Introduction (Part 1/4): https://www.youtube.com/watch?v=BGfmpvM4Zp0&list=PLfdiltiRHWHKQOby9HEyYtB_Y9g7z6yL

JSON-filer - hvorfor??

Når data gemmes i datafiler, opnår vi at **adskille** html-kode og data:

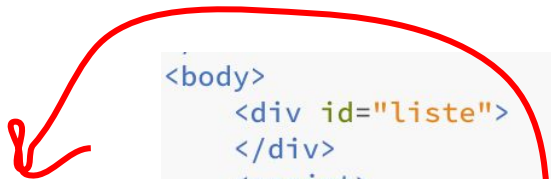
- Dokumentets struktur i **html-fil**
- Layout i **css-fil**
- Js i **js-fil**
- **Data** i **json-fil**

Json-filer bruges til **udveksling af data**

Læg i json:

undervisere.json

```
[
  {
    "navn": "Helle Frederiksen",
    "email": "helf",
    "github": "helefred"
  }, {
    "navn": "Klaus Mandal",
    "email": "klmh",
    "github": "MondaleMondale"
  }, {
    "navn": "Martin Bregnhøi",
    "email": "mabe",
    "github": "martinbregnhoi"
  }, {
    "navn": "Kamilla Viktor",
    "email": "kvi",
    "github": "kvikea"
  }
]
```



```
<body>
  <div id="liste">
  </div>
  <script>
    let personer=[];
    document.addEventListener("DOMContentLoaded", start);
    function start(){
      let dest=document.querySelector("#liste");
      function visPersoner(){
        personer.forEach(person =>{
          dest.innerHTML += `
            <div class="person">
              <h3>${person.fornavn} ${person.efternavn}</h3>
              <p>Email: ${person.email}</p>
              <p>Github: ${person.github}</p>
            </div>
          `;
        })
      }
      visPersoner();
    }
  </script>
</body>
```

Om json:

<https://www.oreilly.com/ideas/json-is-based-on-javascript-object-literals>

https://www.youtube.com/watch?v=BGfmpvM4Zp0&list=PLfdiltiRHWKQOby9HEyYtB_Y9g7z6yL

Øvelse 1: Opret json-fil

1. I mappen **undervisningsopgaver**, laver du en ny undermappe, **json** til dagens opgaver
2. Tag en kopi af din css-grid-stylede dyre-fil fra i går, og kald kopien: **01-dyrljson.html**
3. Tag værdien af arrayet med alle dyr, og læg det ud i en ny json-fil, **dyr.json**

Øv, nu fungerer siden ikke længere - vi har brug for at hente data ind igen fra json-filen - det kommer vi til nu!

Hent og vis json-fil

Hent JSON-fil

Javascript kan hente en json-fil ind vha en særlig **async function**

I funktionen hentes filen med **fetch**

man kan få js til at vente med at gå videre, til filen er indlæste med **await**

Erklæring af personer
(ingen værdi endnu)

```
let undervisere=[];
```


Hent JSON-fil

Javascript kan hente en json-fil ind vha en særlig **async function**

I funktionen hentes filmen med **fetch**

man kan få js til at vente med at gå videre, til filen er indlæst med **await**

Erklæring af personer
(ingen værdi endnu)

```
let undervisere=[];
```

```
async function getJson(){  
    let jsonData = await fetch("undervisere.json");  
    undervisere=await jsonData.json();  
    visUndervisere();  
}
```

Hent JSON-fil

Javascript kan hente en json-fil ind vha en særlig **async function**

I funktionen hentes filmen med **fetch**

man kan få js til at vente med at gå videre, til filen er indlæste med **await**

Erklæring af personer
(ingen værdi endnu)

async function

```
let undervisere=
```

```
async function getJson(){  
  let jsonData = await fetch("undervisere.json");  
  undervisere=await jsonData.json();  
  visUndervisere();  
}
```

Hent JSON-fil

Javascript kan hente en json-fil ind vha en særlig **async function**

I funktionen hentes filen med **fetch**

man kan få js til at vente med at gå videre, til filen er indlæste med **await**

Erklæring af personer
(ingen værdi endnu)

async function

fetch: hent filen
await: vent med at
fortsætte, til den er hentet

```
let undervisere= [],
```

```
async function getJson(){  
  let jsonData = await fetch("undervisere.json");  
  undervisere=await jsonData.json();  
  visUndervisere();  
}
```

Hent JSON-fil

Javascript kan hente en json-fil ind vha en særlig **async function**

I funktionen hentes filen med **fetch**

man kan få js til at vente med at gå videre, til filen er indlæste med **await**

Erklæring af personer
(ingen værdi endnu)

async function

fetch: hent filen
await: vent med at
fortsætte, til den er hentet

```
let undervisere=[],  
  
async function getJson(){  
    let jsonData = await fetch("undervisere.json");  
    undervisere=await jsonData.json();  
    visUndervisere();  
}
```

Variablen **undervisere** sættes til
dataobjektets json-indhold.
await: vent med at fortsætte, til
indholdet er hentet

Hent JSON-fil

Javascript kan hente en json-fil ind vha en særlig **async function**

I funktionen hentes filmen med **fetch**

man kan få js til at vente med at gå videre, til filen er indlæst med **await**

Erklæring af personer
(ingen værdi endnu)

async function

fetch: hent filen
await: vent med at
fortsætte, til den er hentet

```
let undervisere=[],  
  
async function getJson(){  
  let jsonData = await fetch("undervisere.json");  
  undervisere=await jsonData.json();  
  visUndervisere();  
}
```

når arrayet med alle
objekterne er på plads,
kan de vises

Variablen **undervisere** sættes til
dataobjektets json-indhold.
await: vent med at fortsætte, til
indholdet er hentet

Hele koden:

undervisere.json

```
[
  {
    "navn": "Helle Frederiksen",
    "email": "helf",
    "github": "helefred"
  }, {
    "navn": "Klaus Mandal",
    "email": "klmh",
    "github": "MondaleMondale"
  }, {
    "navn": "Martin Bregnhøi", |
    "email": "mabe",
    "github": "martinbregnhoi"
  }, {
    "navn": "Kamilla Viktor",
    "email": "kvi",
    "github": "kvikea"
  }
]
```

```
<body>
  <div id="liste">
  </div>
  <script>
    let personer=[];
    document.addEventListener("DOMContentLoaded", start);
    function start(){
      let dest=document.querySelector("#liste");

      async function getJson(){
        let jsonData = await fetch("undervisere.json");
        personer=await jsonData.json();
        visPersoner();
      }






      function visPersoner(){alert("hi")
        personer.forEach(person =>{
          dest.innerHTML += `
            <div class="person">
              <h3>${person.fornavn} ${person.efternavn}</h3>
              <p>Email: ${person.email}</p>
              <p>Github: ${person.github}</p>
            </div>
          `;
        })
      }

      getJson()
    }
  </script>
</body>
```

Øvelse 2: Indlæs og vis json

Arbejd videre med **01-dyrljson.html** :

- dyrene skal indlæses fra json-filen og vises på siden

frø  Type: amfibie Levested: vandhullet	gris  Type: pattedyr Levested: grisefarmen	ræv  Type: pattedyr Levested: skoven
sild  Type: fisk Levested: havet	krokodille  Type: krybdyr Levested: floden	

Asynchrone events /
promises

Ajax og promises

Fetch kan hente en fil ind til js.

Det er en operation, som tager laaang tid i js's målestok.

Derfor vil programmet bare drøne videre - hvis man ikke får det til at vente.

Await får js til at tage en pause, indtil filen er hentet.

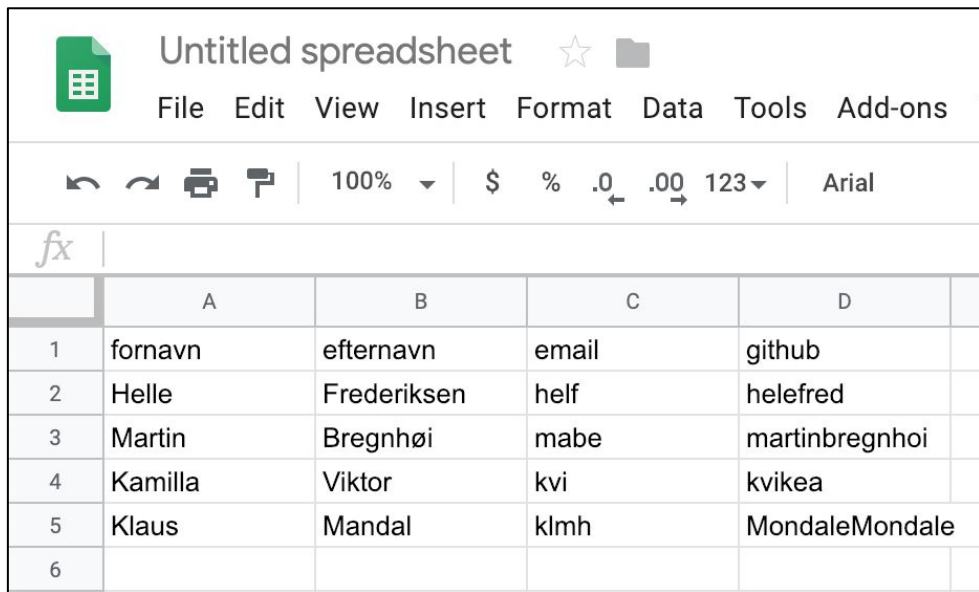
Det kaldes et **promise**, når js er blevet sat på pause

Et asynkront kald kaldes også et **AJAX -kald**(Asynkron JavaScript XML)

Fra google sheets til json-fil

Google sheet

skærmbilleder: sheet og json



	A	B	C	D
1	fornavn	efternavn	email	github
2	Helle	Frederiksen	helf	helefred
3	Martin	Bregnhøi	mabe	martinbregnhoi
4	Kamilla	Viktor	kvi	kvikea
5	Klaus	Mandal	klmh	MondaleMondale
6				

```
[
  - {
    fornavn: "Helle",
    efternavn: "Frederiksen",
    email: "helf",
    github: "helefred"
  },
  - {
    fornavn: "Martin",
    efternavn: "Bregnhøi",
    email: "mabe",
    github: "martinbregnhoi"
  },
  - {
    fornavn: "Kamilla",
    efternavn: "Viktor",
    email: "kvi",
    github: "kvikea"
  },
  - {
    fornavn: "Klaus",
    efternavn: "Mandal",
    email: "klmh",
    github: "MondaleMondale"
  }
]
```

Fra sheet til json

sheetet skal være **public**: vælg **file > publish to the web** og det skal være delt: klik på **“Share”** - vælg **“Anyone with the link can view”**

url-adresse: https://mandalskeawebpace.dk/claude_php/clean_up_spreadsheet.php?id=xxx
[Form hvor du kan taste ID ind.](#)



A red hand-drawn circle highlights the spreadsheet ID '1viVpG5GCWeZgQpTh6AKT-wo1Bpq3RenX2jj5dvmp7Qg' in the URL bar. The URL bar shows a lock icon, the full URL, and a 'gid=0' at the end.

sheetets id tages fra dets url:

hvor skal url-adressen ind i javascript:

```
let jsonData = await
```

```
fetch("https://mandalskeawebpace.dk/claude_php/clean_up_spreadsheet.php?id=xx");
```

Plugin til visning af json-filer

På [Chrome webstore](#), skal du hente en extension, som formatterer jsonfiler pænt.

Udvid fx med denne extension: **JSONView**

Personøvelse

Data til personliste-opgaven

Personlisteopgaven går ud på at hente data fra et regneark, og vi tager udgangspunkt i jeres egne oplysninger - udfyld en række på:

https://docs.google.com/spreadsheets/d/1OLID_GRoz71piHvnW306omRAyHS7vtLOILQt-K0YF_c/edit?usp=sharing

I skal udfylde: navn, github, billede, alder, køn, email, hold, historie

github: kun navnet på jeres konto - ikke en url-adresse til den

billede: Det skal være url'en på et billede, du har liggende af dig selv på nettet - fx. dit profilbillede på Facebook.

email: brug din KEA-mailadresse.

alder: et tal (ikke bogstaver, og skriv ikke "år" bagefter)

køn: m eller k (eller noget andet, som du selv formulerer, men ikke M, K eller F)

hold: a eller b (ikke A eller B)

Sådan henter du adressen på et billede af dig selv, som ligger på nettet: højreklik på det og vælg **Copy Image Address**

Personlisteopgave - aflever på Fronter!!!!

link til løsningen på øvelsen på github

Personlistens data ligger nu på et regneark, og vi kan hente data fra regnearket via Klaus' Dataservice

Regnearket key-værdi er: **1OLID_GRoz71piHvnW306omRAyHS7vtLOILQt-K0YF_c**

Lav en webside, **personListe.html**, som for hver person viser:

navn i en h2-overskrift, billede, og et fungerende link til github-kontoen.

Opsæt den dynamiske personliste, med css grid.

Du skal aflevere **et link til din løsning på github** senest kl. 23.59 i aften.

Intro til modul-opgave (Babushka)

Modulopgave og materialer ligger på Fronter