

JSON

Og hentning af data fra Google sheets

Agenda

- Opfølgning på torsdagens opgaver
 - JSON data
 - Hent JSON-data ind fra fil
 - Asynkrone events i JavaScript (promises)
 - Hent JSON-data ind fra et Google sheets
 - Plugin til visning af json-filer i Chrome
 - Øvelse: persondata
 - Intro til 1. temaopgave: BABUSHKA
-

JSON

Et data-udvekslings-format
som ligner JavaScript objekter

- JSON er en **syntaks til lagring og udveksling af data**.
- JSON er **tekst** skrevet med **JavaScript Object Notation**.
- Når man udveksler data mellem en browser og en server, kan dataene kun være **tekst**.
- Vi kan konvertere JavaScript-objekter til JSON og sende JSON til serveren.
- Og vi kan konvertere JSON modtaget fra en server til JavaScript-objekter i vores scripts.

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

Hvad er det nu et objekt er...?

- En repræsentation af noget,
 - ofte et objekt i den "rigtige" verden som en person, et produkt, et køretøj el.lign. med en række fælles egenskaber.
- En abstraktion.
- En logisk gruppering.
- En datastruktur.



Hvorfor JSON?

Fordi det er smart

Når data gemmes i særligte
datafiler, opnår vi at **adskille**
html-kode mv. fra data:

- Struktur i html
 - Layout i css
 - Handling i JavaScript
 - Data i JSON
-

JSON Syntaks { " " : " " }

JSON datatyper

- Strenge
- Tal
- Objekter
- Arrays
- Booleans
- Null

```
const person = {
    "navn": "Martin" // egenskab/værdi par delt af :
}

const bil = {
    "mærke": "Renault",
    "model": "Clio",
    "udstyr": ["fartpilot", "læderrat", "aircondition"],
    "brændstof": "Benzin",
    "motor": 1.2,
    "hjul": 4,
    "sæder": 5
}
```

JSON Syntaks { " " : " " }

- JSON-syntaks stammer som sagt fra JavaScript ObjektNotation:
 - Data er organiseret i nøgle / værdi-par {"id" : "007"}
 - Data adskilles med kommaer
 - Krøllede parenteser { } omkranser objekter
 - Firkantede parenteser [] indeholder arrays
- Et nøgle/værdi-par består af et feltnavn i dobbelt citationstegn, efterfulgt af et kolon, efterfulgt af en værdi: {"navn" : "Klaus"}
- I JSON skal nøgler (feltnavne) være en streng i dobbelt citationstegn!
- I JavaScript *kan* nøgler være uden citationstegn: {navn: "Martin"}

Næsten lig med
JavaScript objekt
syntaks - men ikke
helt.

Objekter i arrays, arrayas i objekter...

```
undervisere.json (T7) — Brackets
```

```
1 ▼ [  
2 ▼ {  
3   "fornavn": "Martin",  
4   "efternavn": "Bregnøi",  
5   "mail": "mabe@kea.dk",  
6   "emner": ["JavaScript", "CSS", "Projektstyring"]  
7     },  
8 ▼ {  
9   "fornavn": "Klaus",  
10  "efternavn": "Mandal Hansen",  
11  "mail": "klmh@kea.dk",  
12  "emner": ["JavaScript", "HTML", "Tøjmode"]  
13    },  
14 ▼ {  
15  "fornavn": "Louise Ea",  
16  "efternavn": "Holbek",  
17  "mail": "loeh@kea.dk",  
18  "emner": ["SoMe", "SEO", "BMC"]  
19    }  
20 ]
```

NB! Sæt IKKE
komma efter sidste
egenskab og efter
sidste objekt!

Adgang til værdierne i JSON

- Objekter og egenskaber nås ved hjælp af punktum .
- Arrays tilgås ved hjælp af firkantede parenteser []

```
undervisere.json (T7) — Brackets  
1 ▼ [  
2 ▼ {  
3   "fornavn": "Martin",  
4   "efternavn": "Bregnhøi",  
5   "mail": "mabe@kea.dk",  
6   "emner": ["JavaScript", "CSS", "Projektstyring"]  
7     },  
8 ▼ {  
9   "fornavn": "Klaus",  
10  "efternavn": "Mandal Hansen",  
11  "mail": "klmh@kea.dk",  
12  "emner": ["JavaScript", "HTML", "Tøjmode"]  
13    },  
14 ▼ {  
15   "fornavn": "Louise Ea",  
16   "efternavn": "Holbek",  
17   "mail": "loeh@kea.dk",  
18   "emner": ["SoMe", "SEO", "BMC"]  
19     }  
20 ]
```

```
const person = {  
  "navn": "Martin",  
  "titel": "Lektor"  
}
```

Præcis på samme
måde som med
JavaScript objekter

```
let navn = person.navn;  
let fornavn = undervisere[2].fornavn;  
let emne = undervisere[0].emner[2];
```

Øvelse 1: Opret json-fil

Individuel øvelse

- I mappen undervisningsopgaver, laver du en ny undermappe "json" til dagens opgaver.
 - Tag en kopi af 06-visMedTemplate.html fra i går, og kald kopien: **01-dyrljson.html**
 - Flyt indholdet af arrayet alleDyr ud i en ekstern json-fil kaldet "alleDyr.json".
 - Nu virker din side ikke længere. Vi har brug for at hente data ind igen fra json-filen. Det kommer vi til om lidt...
-

Hente og vise JSON data

Med `fetch()`

For at hente data fra en JSON-fil ind i vores HTML, skal vi bruge en *asynkron* JavaScript metode ved navn “`fetch`”

Hvad er en asynkron metode?

Også kaldet **AJAX**
(Asynkron JavaScript og XML)

Med **AJAX** kan du:

- Opdatere en webside uden at indlæse siden igen.
- Anmode om data fra en server, efter siden er indlæst.
- Modtage data fra en server, efter siden er indlæst.
- Sende data til en server i baggrunden.

https://www.w3schools.com/xml/ajax_intro.asp

fetch() er en asynkron metode

Det er det smarte ved AJAX, at scripts ikke går i stå, da der kan være flere processer i gang i forskellige tempi (asynkront)

- Fetch er et API til hentning af ressourcer på tværs af netværket.
- Fetch(arg) tager **et obligatorisk argument: stien til den ressource**, du vil hente, f.eks. **JSON**.
- Fetch returnerer et *løfte* (**promise**).
- Et **promise** er et objekt, der repræsenterer den eventuelle færdiggørelse eller fiasko af en asynkron operation.
- **Promise tillader to eller flere asynkrone handlinger at køre parallelt, hvor hver efterfølgende operation starter, når den forrige operation lykkes, med resultatet fra det forrige trin.**
- Udføres ved hjælp af en “promise chain”.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises

Hente JSON med fetch()

```
• person.json (T7) — Brackets
```

```
1 ▼ {  
2     "navn": "Martin",  
3     "titel": "Lektor"  
4 }
```

```
▼ {navn: "Martin", titel: "Lektor"} ⓘ  
  navn: "Martin"  
  titel: "Lektor"  
► __proto__: Object
```

```
const fil = "person.json";  
  
async function hentdata(fil) {  
    const resultat = await fetch(fil);  
    const json = await resultat.json();  
    vis(json);  
}  
  
function vis(json) {  
    console.log(json);  
}  
  
hentdata(fil);
```

Vise JSON data i HTML

```
<body>
  <h1>Undervisere</h1>
  <main></main>

  <template>
    <article>
      <h2 class="fornavn">NAVN</h2>
      <h3 class="efternavn">EFTERNAVN</h3>
      <p class="mail">MAIL</p>
    </article>
  </template>

  <script>
    "use strict";

    const fil = "undervisere.json"; ←

    async function hentdata(fil) {
      const resultat = await fetch(fil);
      const json = await resultat.json();
      vis(json);
    }

    function vis(json) {
      const beholder = document.querySelector("main");
      const skabelon = document.querySelector("template");
      json.forEach(underviser => {
        const klon = skabelon.cloneNode(true).content;
        klon.querySelector(".fornavn").textContent=underviser.fornavn;
        klon.querySelector(".efternavn").textContent=underviser.efternavn;
        klon.querySelector(".mail").textContent=underviser.mail;
        beholder.appendChild(klon);
      });
      hentdata(fil);
    }
  </script>
</body>
```

undervisere.json (T7) — Brackets

```
1  [
2    {
3      "fornavn": "Martin",
4      "efternavn": "Bregnøi",
5      "mail": "mabe@kea.dk",
6      "emner": ["JavaScript", "CSS", "Projektstyring"]
7    },
8    {
9      "fornavn": "Klaus",
10     "efternavn": "Mandal Hansen",
11     "mail": "klmh@kea.dk",
12     "emner": ["JavaScript", "HTML", "Tøjmode"]
13   },
14   {
15     "fornavn": "Louise Ea",
16     "efternavn": "Holbek",
17     "mail": "loeh@kea.dk",
18     "emner": ["SoMe", "SEO", "BMC"]
19   }
20 ]
```

Undervisere

Martin

Bregnøi

mabe@kea.dk

Klaus

Mandal Hansen

klmh@kea.dk

Louise Ea

Holbek

loeh@kea.dk

Øvelse 2: Indlæs og vis json

Arbejd videre med
01-dyrljson.html.

Dydrene skal nu 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

Fra Google sheet til json-fil

Google sheet eksempel

fiktivt-persongalleri-f-2020

File Edit View Insert Format Data Tools Help Last edit was seconds ago

1 id navn efternavn køn alder religion hobby billede

0	Captain Raymond	Holt	m	57	gay	Being mad at Jake	<https://media1.tenor.com/images/856a1542624b00124ddd7777858e9e4c/tenor.gif?itemid=1564522481099-90a7c3d1c8a7?xlib=rb-1.2.1&ixid=ey>
1	Margueritte	den fjerde	k	99	meget overtroisk	natur	<https://images.unsplash.com/photo-1564522481099-90a7c3d1c8a7?xlib=rb-1.2.1&ixid=ey>
2	Peter	den Store	andet	55	Ateist	mord	https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/Peter_der-Grosse_1838.jpg/2
3	Kirk	Douglas	m	103	Kristen	Modeljernbane	https://cdn.vox-cdn.com/thumbor/862nJr_kkQ3hqg-357JyhAOKYXY=/0x0:2400x3000/182
4	Roger	Rabbit	m	7	Den Store Modernkanin	mælkehøster	<https://i.pinimg.com/474x/26/63/21/2663211c300f80e98c5902375bdd7510--jessica-rabbit-i>
5	Lauren	Bacall	k	90	Kristen	Bogart	<https://i.guim.co.uk/img/static/sys-images/Guardian/Pix/pictures/2014/8/13/140789093920>
6	Bruce	Wayne	m	40	Ateist	Store biler	<https://encrypted-tbn0.gstatic.com/images?q=tbn%3AAnd9GcQAwEO4xwGlhxBvkzcBY1>
7	Tine	Pingvin	k	22	Pinguinist	Nail art	https://cdn.shopify.com/files/1/0115/4891/7819/products/19568_600x_crop_center.jpg?v=
8	Kenneth	Tordenskjold	m	67	Skoven	Tommelfingretning	<https://i.pinimg.com/originals/81/96/cc/8196ccfc969805c84c5da0570e36ff16.jpg>
9	Kometta	Fransis	k	14	Rummet	stjerne	<https://images.unsplash.com/photo-1451187580459-43490279cfa?xlib=rb-1.2.1&ixid=ey>
10	Vikram	Mukherjee	m	32	Jøde	Sniffe im	https://d2nzqyyfd6k6c7.cloudfront.net/styles/nova_evo_landscape/s3/article/thumbnail/ross
11	John	Dorian	m	28	??	Red liv	<https://vignette.wikia.nocookie.net/scrubs/images/c/c2/S1-HQ-JD.jpg/revision/latest?cb=20>
12	Leronora	Bjørn	andet	100	ukrutt	klippe hæk	<https://encrypted-tbn0.gstatic.com/images?q=tbn%3AAnd9GcTuVY87A983cOJl5Gq5fwF>
13	Elvira	skraaplet	m	14	maaneist	strække halsen	<https://images.unsplash.com/photo-1574870111867-089730e5a72b?xlib=rb-1.2.1&ixid=ey>
14	Doktor	John	m	103	Kristen	Fodbold	<https://images.unsplash.com/photo-1550831107-1553da8c84647?xlib=rb-1.2.1&ixid=eyJhc>
15	Klaes	Buding	m	37	Budist	Stenhuggeri	https://pbs.twimg.com/profile_images/932312301230870528/twoltXlx_400x400.jpg
16	Bojack	Horseman	m	55	UKendt	Drikke sig fuld	<https://cdn.vox-cdn.com/thumbor/saSOhrPbJW41KgSaS2If6cttRg=/0x0:1920x1080/2570>
17	Alfons	Åberg	m	10	tror på spøgelser	Lege	https://upload.wikimedia.org/wikipedia/commons/a/ad/Alfons_%C3%85berg_01_-_PNG.pr
18	Tom	Riddle	m	71	Dødsgardist	Kæle med sin slange	<https://v3.ggpht.com/l/AGF-J7860rgdFGsU43Nay4qdegxrDKIXKhrU7M0xg=s900-c-k-c0>
19	Violetta	Letta	k	50	Budist	At strikke	<https://images.unsplash.com/photo-1551861566-513d73054da5?xlib=rb-1.2.1&ixid=ey>
20	Danni	Boiiii	andet	27	Sin egen	Blære sig	[https://img.discogs.com/4hgUkTF6ELL9fvII-1EndTlHtrQ=filt-in/300x300/filters:strip_icc\(\):fc](https://img.discogs.com/4hgUkTF6ELL9fvII-1EndTlHtrQ=filt-in/300x300/filters:strip_icc():fc)
21	Hugh	Mongus	m	56	Javascript	Sidde i mors Kælder og	<http://ddalby.dk/kea/fatman/fatman.jpg>
22	Hans C	Anderen	andet	213	fynbo	klippe klister	<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.hcandersen-homepage.dk%>
23	Meredith	Greys	k	50	Kristen	Arbejde	<https://i.pinimg.com/originals/51/10/83/51108303591973c9edc4e7b18945d60.jpg>

Google sheet'et skal være publiseret

docs.google.com/spreadsheets/d/1mtHhpEkYqVqr1TSnKQ1_sQZWI7FRVjcw3H1oatDOVQ/edit#gid=0

Frontier KEA Blanket KEA mail LI Learning AULA Tabulex Facebook DMI TV Netflix Bank e-Boks Map

fiktivt-persongalleri-E2020

File Edit View Insert Format Data Tools Add-ons

Share

New

Open

Import

Make a copy

Email as attachment

Download

Make available offline

Version history

Rename

Move

Add shortcut to Drive

Move to trash

Publish to the web

Email collaborators

Document details

Spreadsheet settings

Print

Sheet1 Sheet2

Publish to the web

This document is published to the web.

Make your content visible to anyone by publishing it to the web. You can link to or embed your document. [Learn more](#)

Link **Embed**

Entire Document [Web page](#)

<https://docs.google.com/spreadsheets/d/e/2PACX-1vQzrNDQRiLg6NEIJEJi-o>

Or share this link using:

Note: Viewers may be able to access the underlying data for published charts. [Learn more](#)

Published

Published content & settings

Entire Document [Stop publishing](#)

Automatically republish when changes are made

NB: Du skal IKKE bruge den nye URL herfra, men den der står i browserens adressefelt, når du senere skal hente dine data med fetch.

URL til at hente JSON fra et Google sheet

For et hente JSON ud af vores Google sheet, skal vi bruge en særlig URL syntaks:

`https://spreadsheets.google.com/feeds/list/her sættes ID'et ind/od6/public/values?alt=json`

ID'et finder du i browserens adressefelt, når du har dit sheet åbent:

 https://docs.google.com/spreadsheets/d/1XWWbfWszD7f4jHqp51V_oT3pkHuR-ceEUw4YtrvK7F0/edit#gid=0

ID'et skal flettes ind i den særlige syntaks som i det her eksempel:

`https://spreadsheets.google.com/feeds/list/1XWWbfWszD7f4jHqp51V_oT3pkHuR-ceEUw4YtrvK7F0/od6/public/values?alt=json`

Det er ikke kønt, men sådan er Google's API altså lavet. Og det bliver ikke bedre, når vi prøver URL'en af i browseren...



Installer JSON Viewer i Chrome

The screenshot shows the Chrome browser interface with three main components:

- Left Panel (Extension Overview):** Displays the "JSON Viewer 0.18.0" extension. It includes a description: "The most beautiful and customizable JSON/JSONP highlighter that your eyes have ever seen. Open source at <https://goo.gl/fmp hc7>". Below the description is the extension ID: "ID: gbm dgp bip fall nflgajpaliibnhdgobh". A blue link "Inspect views [background page \(Inactive\)](#)" is also present. At the bottom are "Details" and "Remove" buttons, and a toggle switch.
- Middle Panel (Context Menu):** A context menu is open over a page. The visible items are: "Save Page As..." (⌘S), "Create Shortcut...", "Clear Browsing Data..." (⇧⌘⌫), "Extensions" (highlighted in grey), "Task Manager", and "Developer Tools" (⌥⌘I).
- Right Panel (Browser Menu):** The browser's main menu is open, showing the following options:
 - New Tab ⌘T
 - New Window ⌘N
 - New Incognito Window ⇧⌘N
 - History ▶
 - Downloads ⌘L
 - Bookmarks ▶
 - Zoom: - 100% + [zoom icon]
 - Print... ⌘P
 - Cast...
 - Find... ⌘F
 - More Tools ▶
 - Edit
 - Cut
 - Copy
 - Paste
 - Settings ⌘,
 - Help ▶

Chrome File Edit View History Bookmarks People Tab Window Help

Fronter | 2. semester N | JSON 2.sem. | https://spreadsheet.google.com/feeds/list/1Xge7s1Z9dEOTCn1Yx130E4xgvrX0L8Y_iu3WN1yGB1U/od6/public/values?alt=json | JSON (T7, ale) | https://spreadsheet.google.com/feeds/list/1Xge7s1Z9dEOTCn1Yx130E4xgvrX0L8Y_iu3WN1yGB1U/od6/public/values?alt=json | fiktivt-person | fiktivt-person | about:blank | +

spreadsheets.google.com/feeds/list/1Xge7s1Z9dEOTCn1Yx130E4xgvrX0L8Y_iu3WN1yGB1U/od6/public/values?alt=json

```
// 20200820144419
// https://spreadsheets.google.com/feeds/list/1Xge7s1Z9dEOTCn1Yx130E4xgvrX0L8Y_iu3WN1yGB1U/od6/public/values?alt=json
{
  "version": "1.0",
  "encoding": "UTF-8",
  "feed": {
    "xmlns": "http://www.w3.org/2005/Atom",
    "xmlns$openSearch": "http://a9.com/-/spec/opensearchrss/1.0/",
    "xmlns$gsx": "http://schemas.google.com/spreadsheets/2006/extended",
    "id": {
      "$t": "https://spreadsheets.google.com/feeds/list/1Xge7s1Z9dEOTCn1Yx130E4xgvrX0L8Y_iu3WN1yGB1U/od6/public/values"
    },
    "updated": {
      "$t": "2020-02-07T07:48:40.585Z"
    },
    "category": [
      {
        "scheme": "http://schemas.google.com/spreadsheets/2006",
        "term": "http://schemas.google.com/spreadsheets/2006#list"
      }
    ],
    "title": {
      "type": "text",
      "$t": "Ark1"
    },
    "link": [
      {
        "rel": "alternate",
        "type": "application/atom+xml",
        "href": "https://docs.google.com/spreadsheets/u/0/d/1Xge7s1Z9dEOTCn1Yx130E4xgvrX0L8Y_iu3WN1yGB1U/pubhtml"
      }
    ]
  }
}
```

Lidt bedre 😊

Men det
ligner nu ikke
helt vores
egen JSON...?

Vores vs. Google's JSON

```
[  
  {  
    "fornavn": "Martin",  
    "efternavn": "Bregnhøi",  
    "mail": "mabe@kea.dk",  
    "emner": ["JavaScript", "CSS", "Projektstyring"]  
  },  
  {  
    "fornavn": "Klaus",  
    "efternavn": "Mandal Hansen",  
    "mail": "klmh@kea.dk",  
    "emner": ["JavaScript", "HTML", "Tøjmode"]  
  },  
  {  
    "fornavn": "Louise Ea",  
    "efternavn": "Holbek",  
    "mail": "loeh@kea.dk",  
    "emner": ["SoMe", "SEO", "BMC"]  
  }]  
]
```

Der er en del “metadata” i Googles JSON, men når vi kommer ned i “entry” begynder det at ligne et array med objekter ligesom vores.

De data vi selv har tastet ind ligger under de nøgler der begynder med “gsx\$”

```
4 ▾ {  
5   "version": "1.0",  
6   "encoding": "UTF-8",  
7   "feed": {  
8     "xmlns": "http://www.w3.org/2005/Atom",  
9     "xmlns$openSearch": "http://a9.com/-/spec/opensearchrss/1.0/",  
10    "xmlns$gsx": "http://schemas.google.com/spreadsheets/2006/extended",  
11    "id": {↖},  
12    "updated": {↖},  
13    "category": [↖],  
14    "title": [↖],  
15    "link": [↖],  
16    "author": [↖],  
17    "openSearch$totalResults": {↖},  
18    "openSearch$startIndex": {↖},  
19    "entry": [  
20      {  
21        "id": {↖},  
22        "updated": {↖},  
23        "category": [↖],  
24        "title": {↖},  
25        "content": {↖},  
26        "link": [↖],  
27        "gsx$id": {↖},  
28        "gsx$navn": {↖},  
29        "gsx$efternavn": {↖},  
30        "gsx$sex": {↖},  
31        "gsx$alder": {↖},  
32        "gsx$religion": {↖},  
33        "gsx$hobby": {↖},  
34        "gsx$billede": {↖},  
35        "gsx$kaeledyr": {↖},  
36        "gsx$job": {↖},  
37        "gsx$by": {↖},  
38        "gsx$rpc": {↖}  
39      },  
40      {↖},  
41      {↖},  
42    ]  
43  }  
44 }  
45 ]
```

Vi bruger fetch igen

```
const url =  
  "https://spreadsheets.google.com/feeds/list/1mtHhp0EkYqVqr1T  
  SnKQ1_sqZWI7FRVjcw3H1oatDOVQ/1/public/values?alt=json";  
  
async function hentdata() {  
  const resultat = await fetch(url);  
  const json = await resultat.json();  
  vis(json);  
}  
  
function vis(json) {  
  console.log(json);  
}  
  
hentdata();
```

Eneste forskel er, at vi sætter den lange URL ind i stedet for et filnavn. Næste step er så, at få fisket de relevante data ud af den returnerede JSON.



Data ligger i et array under json.feed.entry

```
function vis(json) {  
    const beholder = document.querySelector("main");  
    const skabelon = document.querySelector("template");  
    json.feed.entry.forEach(person => {  
        const klon = skabelon.cloneNode(true).content;  
        klon.querySelector(".fornavn").textContent=person.gsx$navn.$t;  
        klon.querySelector(".efternavn").textContent=person.gsx$efternavn.$t;  
        klon.querySelector("img").src=person.gsx$billede.$t;  
        beholder.appendChild(klon);  
    });  
}
```

Personer

```
<template>  
  <article>  
    <h2 class="fornavn">NAVN</h2>  
    <h3 class="efternavn">EFTERNAVN</h3>  
    <img src="" alt="portræt">  
  </article>  
</template>
```

Raymond Holt  HOT DAMN!	Margueritte den fjerde 	Peter den Store 	Kirk Douglas 
Leronora Bjørn 	Elvira skraaplet 	Doktor John 	Klaes Buding 

```
{version: "1.0", encoding: "UTF-8",  
  encoding: "UTF-8"  
  feed:  
    author: [...]  
    category: [...]  
    entry: Array(34)  
      0:  
        category: [...]  
        content: {type: "text", $t:  
        gsx$alder: {$t: "57"}  
        gsx$billede: {$t: "https://..."}  
        gsx$by: {$t: "Brooklyn nine-"}  
        gsx$efternavn: {$t: "Holt"}  
        gsx$hobby: {$t: "Being mad a..."}  
        gsx$id: {$t: "0"}  
        gsx$job: {$t: "Being awesome"}  
        gsx$kaeledyr: {$t: "Cheddar'..."}  
        gsx$køn: {$t: "m"}  
        gsx$navn: {$t: "Captain Ray..."}  
        gsx$religion: {$t: "gay"}  
        gsx$rnk: {$t: "#1"}}
```

Øvelse 3: Lav et fiktivt persongalleri

Indtast og hent data fra et nyt Google sheet og præsenter dem lækkert for brugeren i formateret HTML

Øvelse 3: Persongalleri

1. Tast data om en fiktiv eller historisk person ind i dette Google sheet:
 - o [fiktivt-persongalleri-E2020](#)
2. Lav en flot webside (persongalleri.html), som for hver person viser:
 - o navn i en h2-overskrift
 - o billede
 - o en eller flere af de andre oplysninger.
3. Opsæt den dynamiske personliste med css grid, så siden bliver responsiv, og ser godt ud på alle skærmstørrelser og devices.

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

Bistro Babushka

Individuel opgave til resten af modulet (næste uge)



Dynamisk website til Babushka

På Fronter, under ressourcer, ligger [data-materiale](#) (et Google sheet) til det website, som du skal kode, samt en zip-fil med billeder i forskellig størrelse. Sitet skal præsentere restauranten Babushkas' udbud af retter.

Krav til løsningen

- Der skal være en side, som viser alle restaurantens retter. Her skal retterne præsenteres på en kort og overskuelig måde (loop view)
- Man skal kunne få yderligere informationer om den enkelte ret, ved at klikke på "noget" i retten (single views)
- Der skal være en navigationsmenu på siden, som giver mulighed for, at man får særskilte oversigter over forretter, hovedretter og desserter (filtrering i loop views).

Forsiden med loop viewet kan f.eks. se sådan ud: <http://mabe-kea.dk/f18/babushka/>

Aflevering

Torsdag d. 3/9 2020 kl. 23:59 på Fronter.

Du skal både aflevere et link til løsningen på dit domæne samt et link til din løsning på Github.

Feedback

Du får individuel mundtlig feedback på din opgave fredag d. 4/9 2020

Når I har afleveret, laver vi et skema over hvornår hver enkelt får feedback.