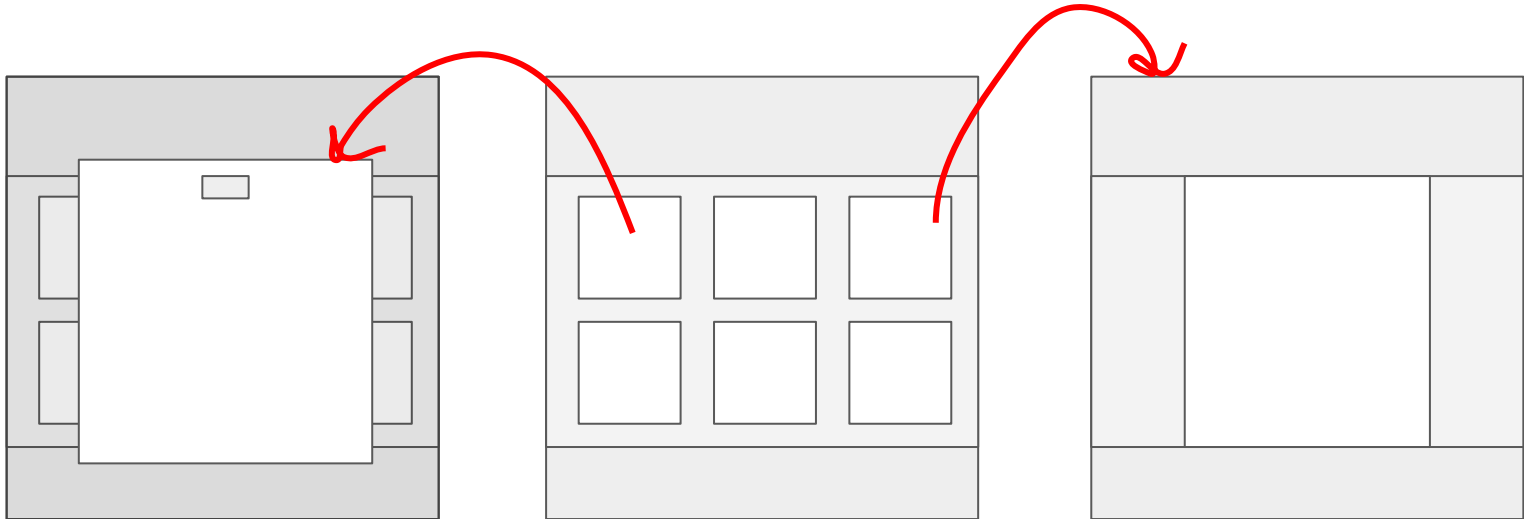


Single view



Fra mandag

Babushka - hvad **skal** der være i menuen:

I opgaven:

Der skal være en menu på siden, som giver mulighed for, at man får særskilte oversigter over forretter, hovedretter og desserter

Så andre rettyper **må** I gerne springe over!

- men det er kun fint at tage dem med.

Martins side er kun et forslag

I behøver ikke at lave en topgrafik! - men det er fint at gøre det.

Mere fra mandag

Startfilen fra igår

Vi ser på den sammen - idag er det vores grundfil.

Agenda

1. Loop view og single view?
2. Single view i popup eller på egen side?
3. Single view i popup - hvordan?
4. Single view på egen side - hvordan?

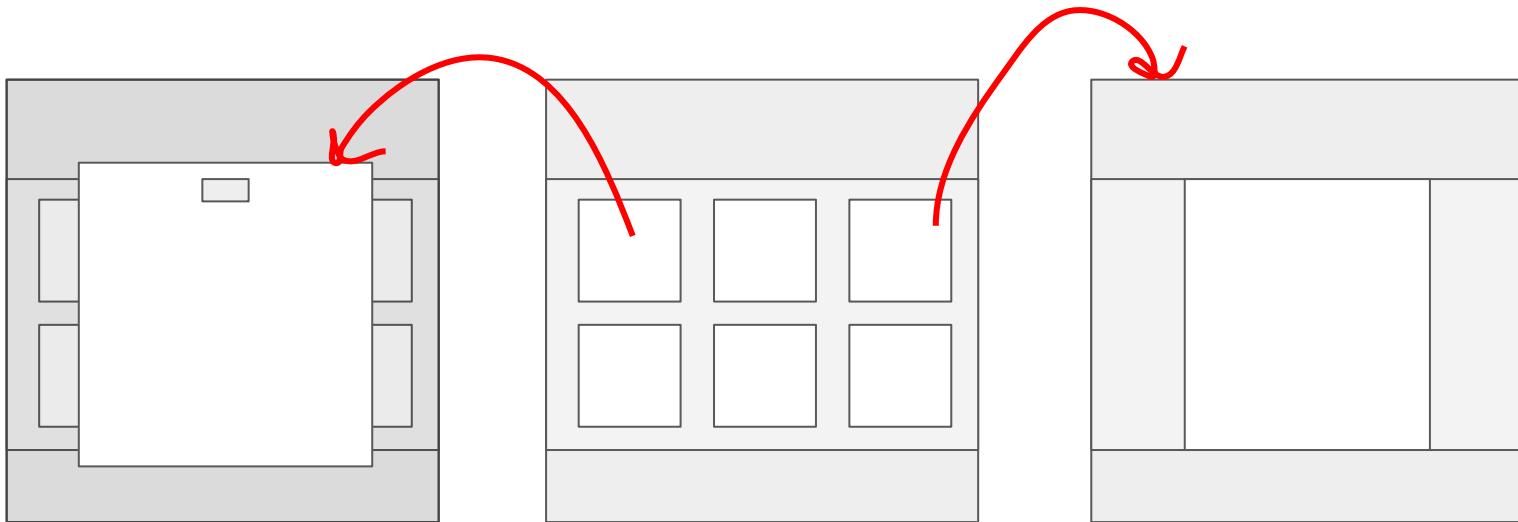
List-view kontra Detail-view

- ★ Loop-view eller list-view er en oversigts-visning af mange objekter (en liste)
- ★ Single-view eller detail-view er en detaljeret visning af ét objekt

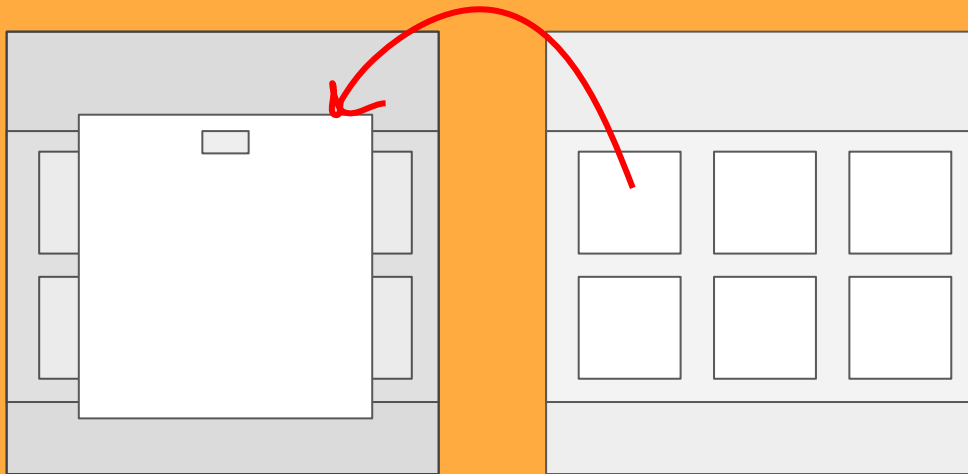
detail
single



2 måder: detail-view i popup eller på ny side



detail-view i pop-up



vis side med popup!

Fordele / ulemper

Fordele:

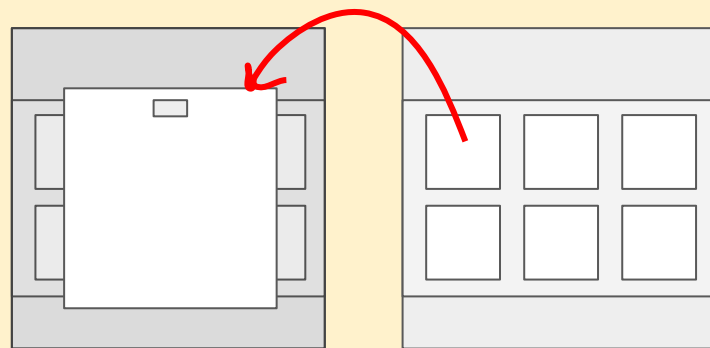
Giver en god brugeroplevelse, fordi der ikke skiftes side

Kan styles meget elegant med lightbox-effekter

Ulemper:

Single-viewet har ikke sin egen url-adresse

- kan ikke deles
- kan ikke findes af søgemaskiner



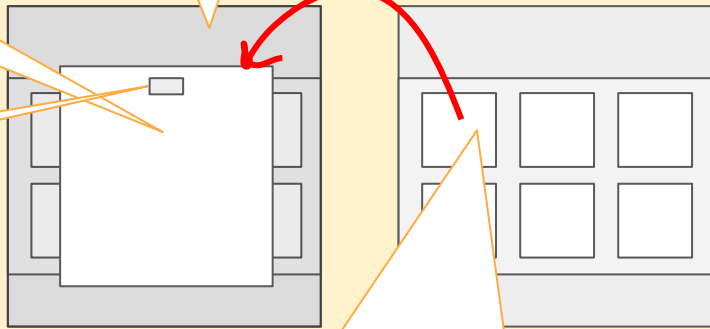
Popup-strategi

1. Lav et popupvindue, som lægger sig ovenpå det hele.

2. læg indholdselementer i popup'en

4. Luk-knap:
Når den klikkes, skal popup'en lukke

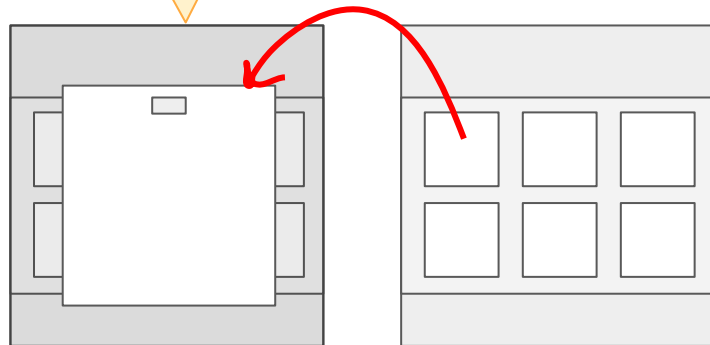
3. eventhandlers på alle elementer i loop-view.
Ved klik skal popup åbne og udfyldes med
detaljer om det element der blev klikket på.



1. Lav popup

- Man kan fx. lægge popup'en øverst eller nederst i dokumentet.
- Det er *ikke* en template, blot et almindeligt HTML-element der kan skjules og vises via css og JS, f.eks. en `<section>`
- Det skal styles, så det fylder hele vinduet / viewport (vw og vh: 100)
- Det skal styles, så det lægger sig ovenpå det eksisterende indhold (position: fixed)
- Giv den en baggrundsfarve, så den kan ses, mens vi udvikler.

1. Lav et popupvindue, som lægger sig ovenpå det hele.



```
<section id="popop">
```

```
  <article>
    <div id="luk">X</div>
    <h2></h2>
    <img src="" alt="">
    <p></p>
  </article>
</section>
```

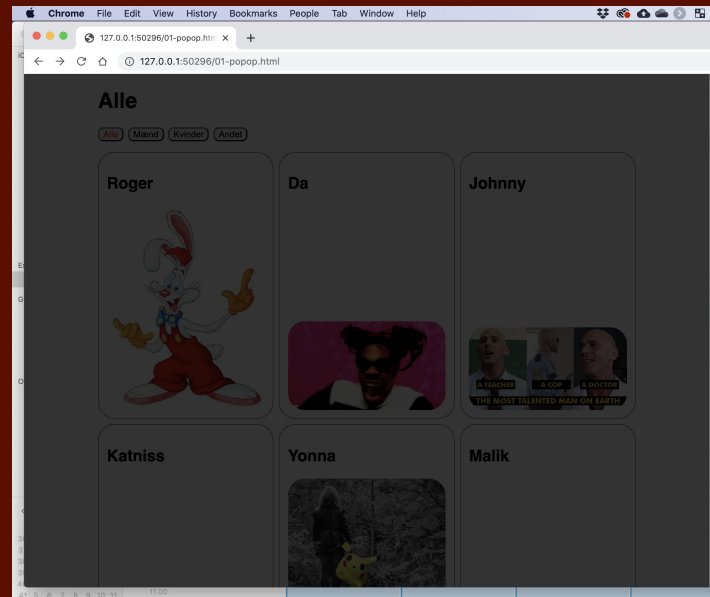
```
#popop {
  position: fixed;
  left: 0;
  top: 0;
  width: 100vw;
  height: 100vh;
  background-color: rgba(0, 0, 0, .8);
  display: none;
}
```

```
#popop article {
  width: 80vw;
  height: 80vh;
  margin: 4rem auto;
  background-color: white;
}
```

```
#luk {
  position: fixed;
  top: 0;
  right: 1em;
  background-color: white;
  color: black;
  border-radius: 5px;
  padding: .1em;
}
```

Øvelse 1 - lav en popup

- Opret en ny mappe, **single**, i **tema7**-mappen.
- Gem en kopi af din **startfil fra igår** i mappen under navnet **01-popup.html** heri.
- Læg et popup-vindue (section) øverst i HTML dokumentet.
- Style det, så det fylder hele vinduet og lægger sig ovenpå det eksisterende indhold.
- Giv også popup-vinduet en baggrundsfarve, så det er til at få øje på og dækker baggrunden.
- Test om det fungerer.

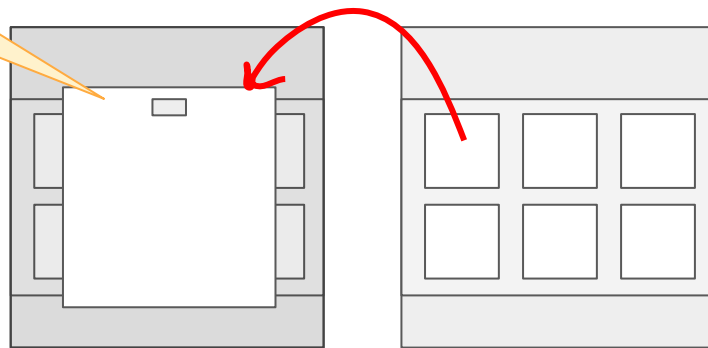


2. Indholdselementer i popup

2. læg indholdselementer i popup'en

Der skal lægges lidt indhold ind i popup-vinduet:

- En luk-knap.
- En article til indholdet.
- Nogle elementer til detaljerne, f.eks. `<h2>`, `` og `<p>`
- Styling af indholds-article.
- Styling af luk-knap.



```
<section id="popop">
  <article>
    <div id="luk">X</div>
    <h2></h2>
    <img src="" alt="">
    <p></p>
  </article>
</section>
```

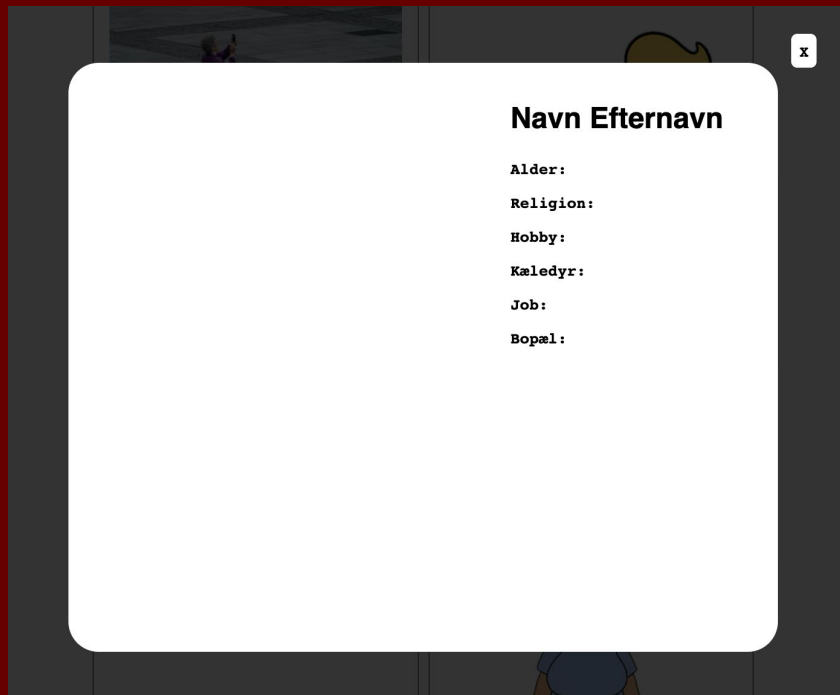
```
#popop {
  position: fixed;
  left: 0;
  top: 0;
  width: 100vw;
  height: 100vh;
  background-color: rgba(0, 0, 0, .8);
  display: none;
}
```

```
#popop article {
  width: 80vw;
  height: 80vh;
  margin: 4rem auto;
  background-color: white;
}
```

```
#luk {
  position: fixed;
  top: 0;
  right: 1em;
  background-color: white;
  color: black;
  border-radius: 5px;
  padding: .1em;
}
```

Øvelse 2

- Arbejd videre med **01-popup.html** i mappen under navnet **02-indhold.html**.
- Læg HTML elementer i popup'en til indhold og lukkeknop
- Style, så vinduet bliver funktionelt
- Style til sidst popup'en, så den er skjult (kan også gøres med js...)

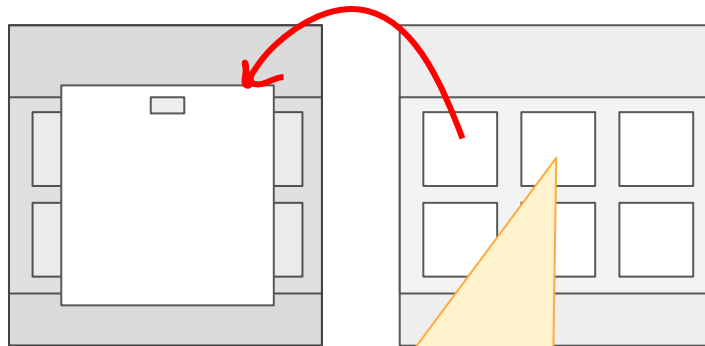


3. Eventhandlers på alle person-elementer i list-view

Ved klik på en person, åbnes og udfyldes popup

Derfor:

- **eventlisteners** på alle elementer
- **eventhandler-funktionen** skal udfylde popup-vinduet med detaljer om den valgte person.
- **Styling** af popup-vinduet indhold



3. eventhandlers på alle elementer i loop-view:
ved klik skal popup åbne og udfyldes


```
function visPersoner() {  
  const templatePointer = document.querySelector("template");  
  const listPointer = document.querySelector("#list");  
  listPointer.innerHTML = "";  
  personer.feed.entry.forEach(person => {  
    if (filter == "alle" || filter == person.gsx$køn.$t) {  
      console.log(person);  
      const minKlon = templatePointer.cloneNode(true).content;  
      minKlon.querySelector("h2").textContent = person.gsx$navn.$t;  
      minKlon.querySelector("img").src = person.gsx$billede.$t;  
  
      minKlon.querySelector("article").addEventListener("click", () => visDetaljer(person));  
  
      listPointer.appendChild(minKlon);  
    }  
  })  
}  
  
function visDetaljer(person) { ... }  
  
document.querySelector("#luk").addEventListener("click", ()=>popop.style.display="none");
```

Med en anonym function kan vi medsende en parameter (person) der indeholder al den relevante data til at vise i popop'en

I hver iteration tilføjes det nye element (article) en click-eventlistener. Når forEach er færdig, er alle artikler blevet klikbare.

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	navn	efternavn	køn	alder	religion	hobby	billede	kæledyr	job	by	land
2	1	Roger	Rabbit	m	7	Den Store Moderkanin	mælkebøtter	https://i.pinimg.com/4	mennesker	kanin	Hollywood	eventyrland

```

<section id="popop">
  <article>
    <p id="luk">X</p>
    <img src="" alt="">
    <section>
      <h2></h2>
      <p>Alder: <span class="alder"></span></p>
      <p>Religion: <span class="rel"></span></p>
      <p>Hobby: <span class="hobby"></span></p>
      <p>Kæledyr: <span class="kdyr"></span></p>
      <p>Job: <span class="job"></span></p>
      <p>Bopæl: <span class="by"></span></p>
    </section>
  </article>
</section>

```

En detaljevisning vil
oftest have et andet
og mere (html)indhold
end på en liste

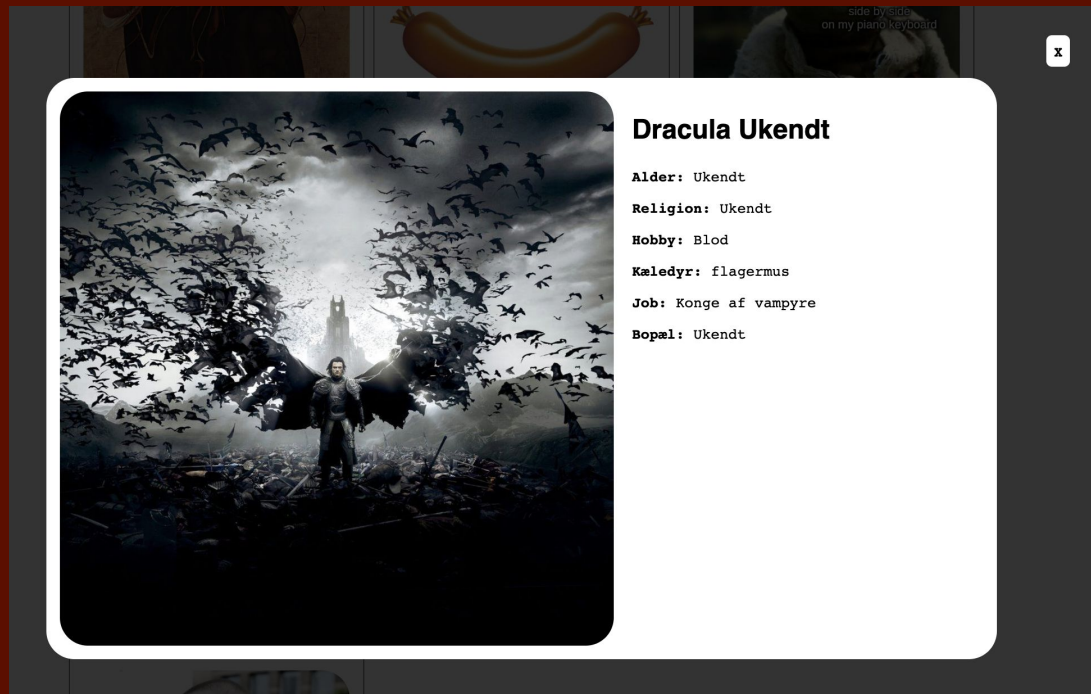
```

function visDetaljer(person) {
  popop.style.display = "block";
  popop.querySelector("h2").textContent = person.gsx$navn.$t;
  popop.querySelector("h2").textContent += " " + person.gsx$efternavn.$t;
  popop.querySelector(".alder").textContent = person.gsx$alder.$t;
  popop.querySelector(".rel").textContent = person.gsx$religion.$t;
  popop.querySelector(".hobby").textContent = person.gsx$hobby.$t;
  popop.querySelector(".kdyr").textContent = person.gsx$kæledyr.$t;
  popop.querySelector(".job").textContent = person.gsx$job.$t;
  popop.querySelector(".by").textContent = person.gsx$by.$t;
  popop.querySelector("img").src = person.gsx$billede.$t;
}

```

Øvelse 3

- Gem en kopi af din **02-indhold.htm** i mappen under navnet **03-åbn.html**.
- Når man klikker på en person i loop-view, skal popup med single-view åbnes.
- Vis artiklerne er klikbare med `css` eller `js: cursor: pointer`
- Hvis din `#popup` ikke kan scrolle: `overflow: scroll;`



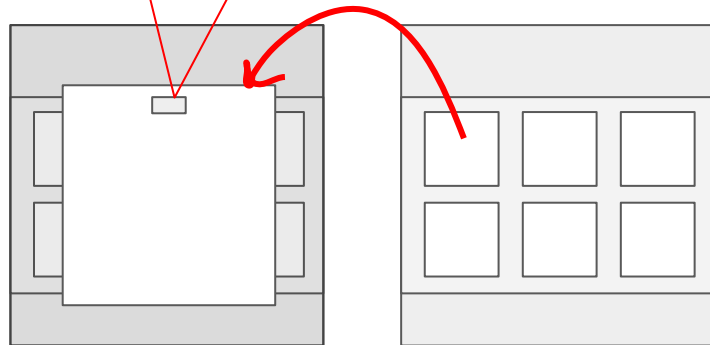
4. Luk-knappen

- Læg en lukke-knap ind i popup-vinduet.
- Lav en eventListener til knappen: når den klikkes, skal popup-vinduet lukke.

```
<section id="popop">  
  <article>  
    <p id="luk">X</p>  
    <img src="" alt="">  
  </section>
```

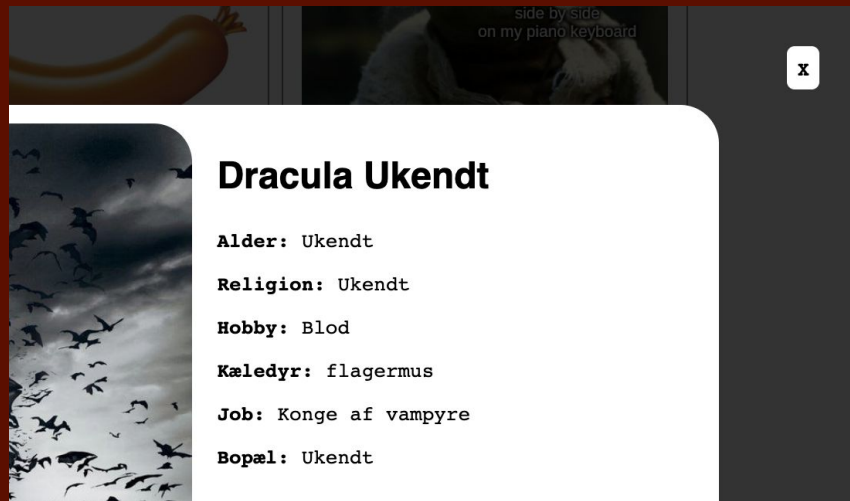
```
document.querySelector("#luk").addEventListener("click", () => popop.style.display = "none");
```

4. Luk-knap:
Når den klikkes, skal popup'en lukke

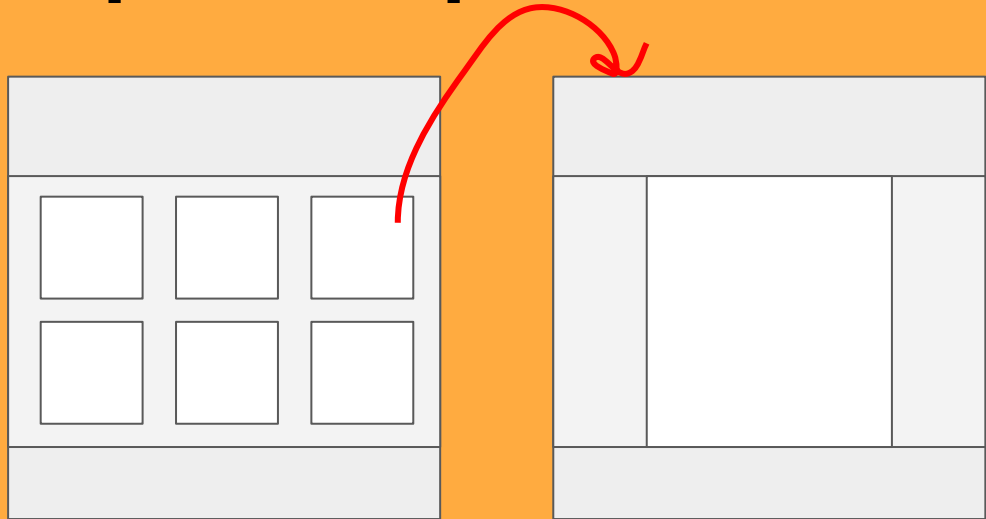


Øvelse 4

- Gem en kopi af **03-åbn.html** i mappen under navnet **04-luk.html**.
- Sæt en luk-knap ind på popup-vinduet og style, så den ser ok ud. (husk også cursor: pointer)
- Lav en eventListener på knappen, som sørger for at lukke popupvinduet igen.
- Så er popup-vinduet færdigt - bortset måske fra lidt ekstra styling ;-)



detail-view på separat side



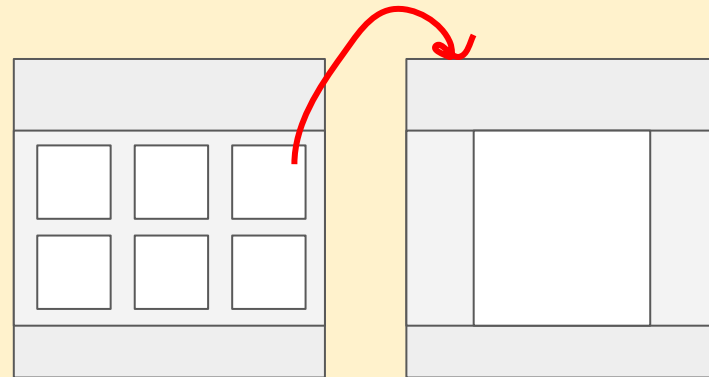
Fordele/ ulemper

Fordele:

Single-viewet får sin egen url-adresse

siden kan nemmere deles

siden kan findes af Google



Ulemper:

der skiftes side, når der klikkes på en person - en underside skal altså laves

Separat-side strategi

1. Når der klikkes på et element i liste-viewet, skal browseren hoppe over til en ny side, detalje.html

En variabel skal medbringes

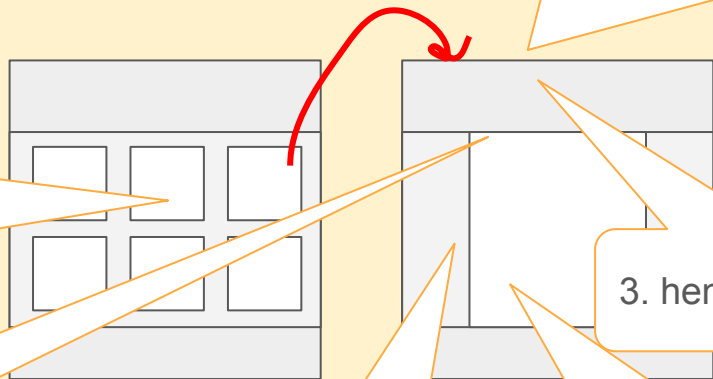
2. Lav detalje.html:
(html og styling ud fra #detalje)

3. hent url-variablen

4. hent alle personer og find den rigtige

5. udfyld #detalje (genbrug function visPersoner() med indholdet fra visDetalje())

6. luk-knappen skal laves om til en tilbageknop
Få tilbageknappen til at virke



1. kald med url-variabel

I loop-view:

Eventlisteners på alle elementer.

eventhandler:

spring over til den nye side.

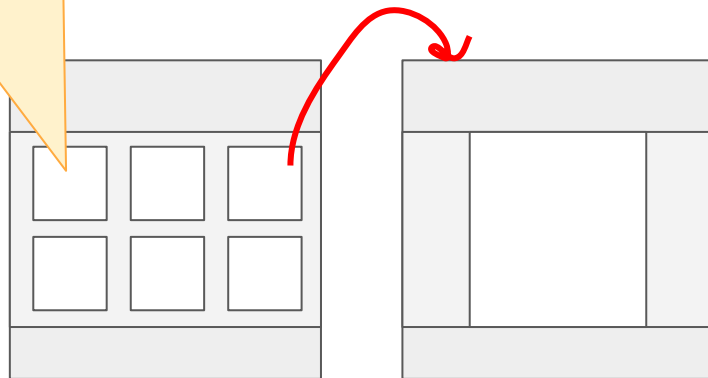
I javascript kan man hoppe til en ny adresse med **location.href**

Medbring en variabel, som kan fortælle hvilket element, der blev klikket på

Variablen kan sendes som **url-variable**

1. Når der klikkes på et element i loop-viewet, skal browseren hoppe over til en ny side, single.html

En variabel skal medbringes



window.location

I javascript har vi et indbygget objekt, `window.location`

`window.location` har en lang række egenskaber

En af dem er `href`:

```
alert(window.location.href)
```

vil vise den aktuelle sides url i en alert-popup

Mens:

```
window.location.href = "http://kea.dk"
```

får browseren til at springe over til en helt anden side: keas webside.

“window” kan udelades, så vi skriver: `location.href`

url-variabler

Når en side kaldes med en url-adresse, kan man sende variabler med til siden. Sætter jeg denne url-adresse ind i browserens adressefelt:

<https://kvikea.dk/hej.html?navn=Flemming>

vil browseren hente siden <https://kvikea.dk/hej.html>, og den vil desuden give siden en variabel, **navn**, som er lig med “**Flemming**”.

Vi kommer senere til, hvad modtagersiden gør, for at få fat i variabelen.

hvilke url-variabler skal gemmes?

Den variabel, der skal overføres til den nye side, skal sætte den nye side i stand til at slå personen op, hvis alle personer hentes herfra.

Derfor skal den variabel vi overfører være **unik**.

Vi kan **ikke** bruge en variabel, hvis to personer kan have **samme værdi**

I personlisten kunne det være **email** eller **github**-kontonavn

Vi ved, at to personer **ikke** kan have **samme github-konto** eller **samme email**. Vi kan også bruge feltet : **id**

Derfor er de hver for sig en **unik nøgle**.

144

```
145 ▼ function visPersoner() {  
146     const templatePointer = document.querySelector("template");  
147     const listPointer = document.querySelector("#list");  
148     listPointer.innerHTML = "";  
149 ▼     personer.feed.entry.forEach(person => {  
150 ▼         if (filter == "alle" || filter == person.gsx$køn.$t) {  
151             console.log(person);  
152             const minKlon = templatePointer.cloneNode(true).content;  
153             minKlon.querySelector("h2").textContent = person.gsx$navn.$t;  
154             minKlon.querySelector("img").src = person.gsx$billede.$t;  
155             minKlon.querySelector("article").addEventListener("click", () => visDetaljer(person));  
156             listPointer.appendChild(minKlon);  
157         }  
158     })  
159 }  
160  
161 ▼ function visDetaljer(person) {  
162     console.log(person);  
163  
164     location.href = `06-detalje.html?id=${person.gsx$id.$t}`;  
165  
166 }  
167
```

Anonym funktion det samme
som at skrive:
function(){visDetaljer(person)}

Bliver fx til:

location.href = "06-single.html?id=21";

Øvelse 5

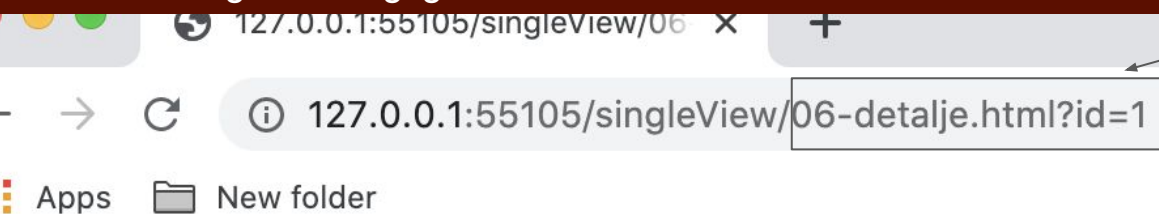
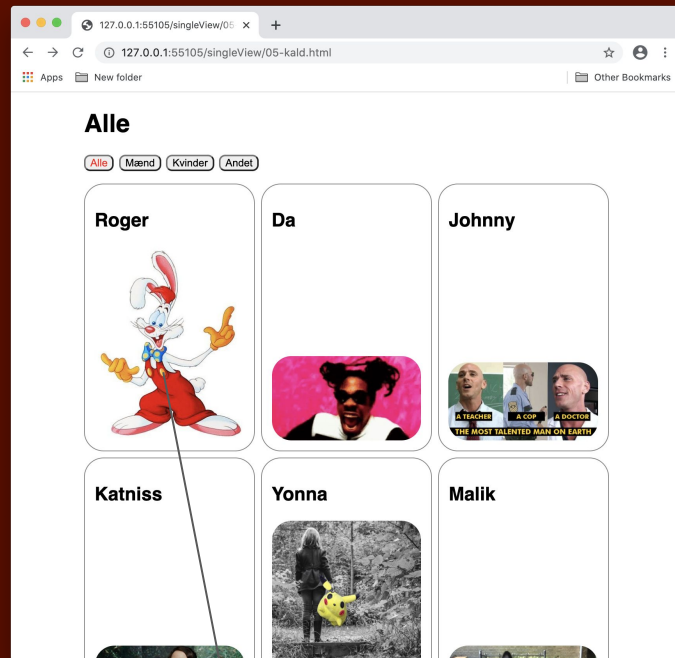
Gem en kopi af **04-luk.html** under navnet **05-kald.html**.

Fjern al html og css, som har med popup-vinduet at gøre
Fjern også i scriptet alt, hvad der har med popup'en at gøre:
variablen popup, eventlisteners på personer og på lukke-knapper.

Læg en eventlistener på alle personer, som får browseren til at hoppe til **06-detalle.html** medbringende personens id .

Test, at det virker: url'en i single.html skal have en variabel med.

vi laver **06-detalle.html** om lidt, men test at browseren prøver OG medbringer den rigtige url-variabel.



2. ny separat side: detalje.html

Lav en single-side. Tag udgangspunkt i en side med popup-vindue (fx. 04-luk.html):

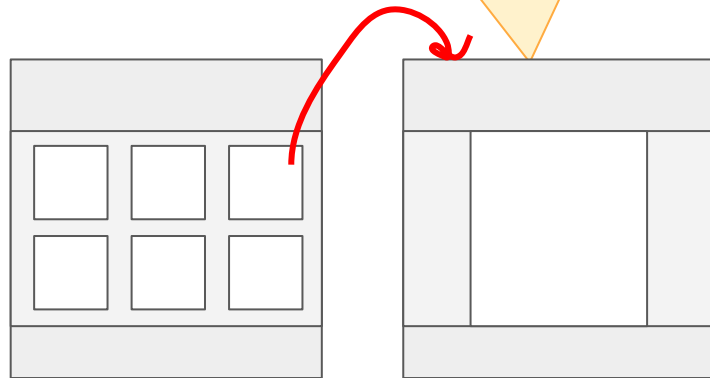
I html'en:

- væk med nav og liste
- bevar kun popup-vinduet.

I css'en:

- bevar kun styling omkring popop - dog skal intet længere være skjult eller have position: fixed (kun luk-knappen)

2. Vi laver en seperat side, **06detalje.html**.
Html og css som popup-vinduet



```

<style>
  img {
    width: 100%;
    height 60%;
    object-fit: contain;
  }

  #detalje {
    position: fixed;
    width: 100vw;
    height: 100vh;
    top: 0;
    left: 0;
    background-color: rgba(0, 0, 0, .8);
    overflow: scroll;
  }

  #detalje #luk {
    position: fixed;
    top: 0;
    left: 0;
    font-size: 4rem;
    margin: 1em;
    cursor: pointer;
  }

  #detalje .person {
    width: 60vw;
    min-height: 50vh;
    margin: auto;
    padding: 4rem;
    background: white;
  }
</style>
</head>
<body>
  <section id="detalje">
    <button id="luk"></button>
    <article class="person">
      <h2 class="navn">Navn</h2>
      <img src="" alt="" class="profil-billede">
    </article>
  </section>

```

Midlertidig hardcodet #detalje for at kunne tilpasse stylingen
-Fjernes inden de dynamiske værdier testes..

```

<!-- En midlertidig hardcodet #detalje til brug for styling
- slettes når tipasset -->
<section id="detalje">
  <article class="person">
    <button class="luk"></button>
    <h2> Mit navn </h2>
    
    <p>Github:
      <a class="githubLink" href="#">github.com/kvikea</a>
    </p>
  </article>
</section>

<!--
<section id="detalje">
  <article class="person">
    <button class="luk"></button>
    <h2></h2>
    <img src="" alt="">
    <p>Github:
      <a class="githubLink" href=""></a>
    </p>
  </article>
</section>
-->

```


Øvelse 6

Gem en kopi af **04-luk.html** under navnet **06-detalle.html**.

Fjern html og styling, som ikke har med #popop at gøre. Tilpas stylingen for #popop så den ikke længere er fixed eller skjult. (luk-knappen kan stadig være fixed og bruges som tilbageknap)

Eller - lav **06-detalle.html** fra bunden. i html : genbrug hele #detalle-sektionen og al relevant styling.

Lav evt. en midlertidig kopi af #detalle med rigtige værdier i, så du kan se hvad du styler..



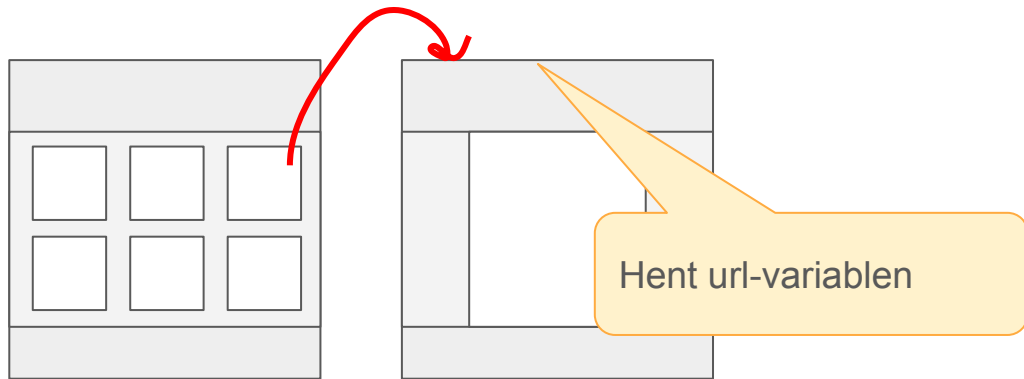
Kamilla Viktor



Github: github.com/kvikea

3. Få fat i url-variablen med URLSearchParams

Du skal skrive “ ?id=..... ” i adresselinien eller klikke fra **05-kald.html** for at teste url-variablerne



```
<script>
```

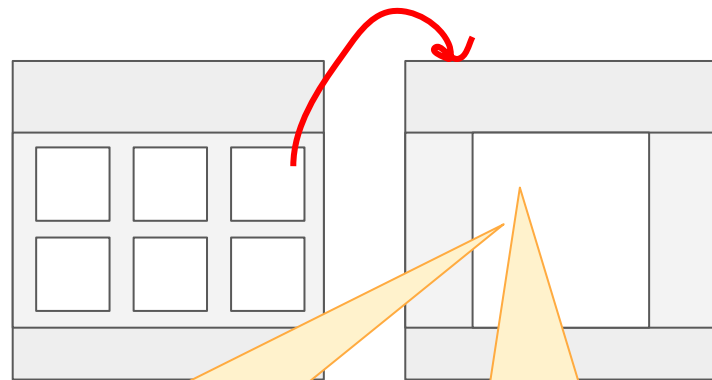
```
const urlParams = new URLSearchParams(window.location.search);  
const id = urlParams.get("id");
```

```
let personer;
```

```
const popop = document.querySelector("#popop");
```

Lav en constant der hedder id og gem url-variablen der

4. Find den rigtige person
5. Udfyld #detalje med data



I scriptet:

slet alt om filtrering, hvis der er noget endnu
i if-sætningen i loopet: - erstat med:

```
if (person.gsx$id.$t == id) {  
  //... vis indholdet fra visDetaljer-funktionen  
};
```

<script>

```
const urlParams = new URLSearchParams(window.location.search);  
const id = urlParams.get("id");
```

```
let personer;
```

```
const popop = document.querySelector("#popop");  
document.addEventListener("DOMContentLoaded", loadJSON)
```

```
async function loadJSON() {  
  const JSONData = await  
    fetch("https://spreadsheets.google.com/feeds/list/1mtHhpoEkYqVqr1TSnKQ1_sQZWI7FRVjcw3H1oatDOVQ/od6/publi  
c/values?alt=json");  
  personer = await JSONData.json();  
  // addEventListenersToButtons();  
  visPersoner();  
}
```

```
function visPersoner() {  
  const templatePointer = document.querySelector("template");  
  const listPointer = document.querySelector("#list");  
  listPointer.innerHTML = "";  
  personer.feed.entry.forEach(person => {  
    if (id == person.gsx$id.$t) {  
      visDetaljer(person)  
    }  
  })  
}
```

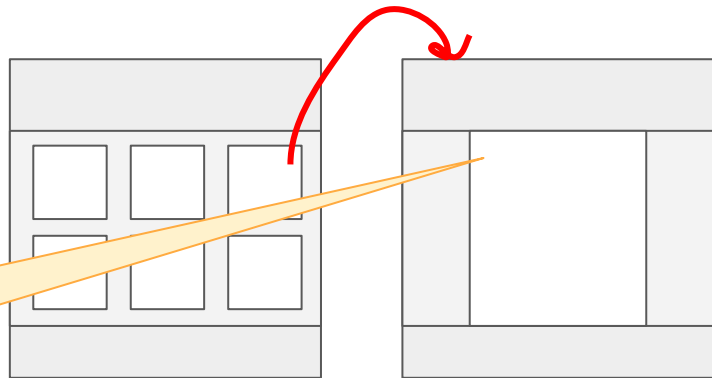
```
function visDetaljer(person) {  
  console.log(person);  
  popop.style.display = "block";  
  popop.querySelector("h2").textContent = person.gsx$navn.$t;  
  popop.querySelector("img").src = person.gsx$billede.$t;  
}
```

```
document.querySelector("#luk").addEventListener("click", () => history.back());
```

</script>

6. Eventlistener og eventhandler til Tilbage-knap

6. luk-knappen skal laves om til en tilbageknap
Få tilbageknappen til at virke



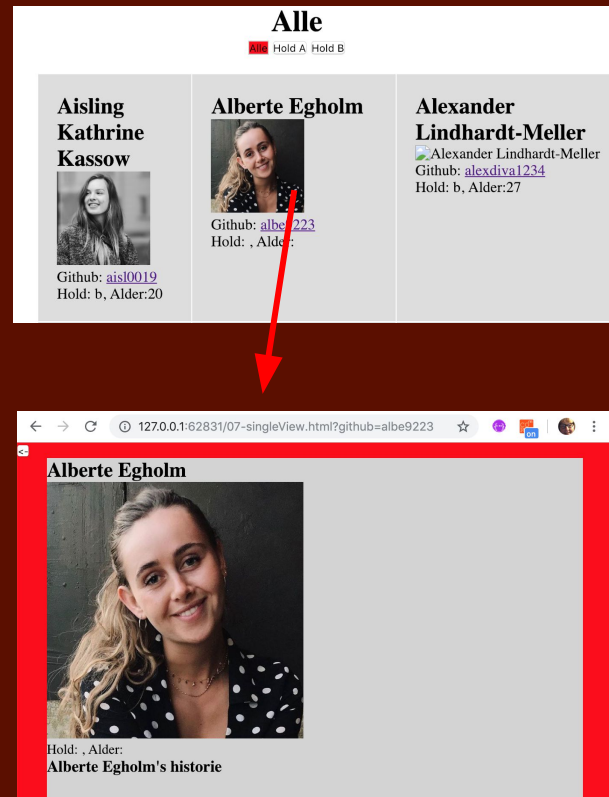
```
document.querySelector("#luk").addEventListener("click", () => history.back());
```

Øvelse 7

Arbejd videre med **06-detalle.html**

I scriptet skal du :

- hente url-variablen ved hjælp af URLSearchParams.
- fjerne alt overflødig som filtrering osv.
- tilpas visPersoner(), med ny if-sætning
- Erstat luk-knappen med en tilbage-knap, som kan springe tilbage til **05-kald.html**
- husk, når du tester, at **06-detalle.html** skal testes ved at klikke på en person i **05-kald.html**,



Babushka og singleview - hvad skal I lave

I skal lave **to løsninger**:

1. En Babushka-side med popup-detail-view
2. En Babushka-side, som åbner en ny side til detalje-visning

Lav de to løsninger i en mappe, **babushka**, og aflever et link til **mappen på Github**.

Du skal også uploade mappen til dit domæne.

Links til de to løsninger på domænet kan du lægge i **Readme-filen på dit github-repository** for mappen.