

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.

Ændring af afleveringstidspunkt

Vi syntes, at programmet blev lidt heftigt. Derfor har vi lavet disse ændringer:

Aflevering af caseopgave I: flyttet fra onsdag til torsdag

Feedback: flyttet fra torsdag til fredag

I har altså en ekstra dag at arbejde med casen: torsdag med vejledning.

De punkter, der var på fredagsprogrammet udskydes.

I får en ny dato for afleverings af tidsplan.

Single view

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?

Loop-view kontra single-view

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

loop


single

KALENDER GENRE BILLEDER INFO KONTAKT SØG

TORS DAG 14. FEB 2019 - KL 20.30 FORSALG: 80KR KØB

DILLY DALLY (CA)

silbpl/



Dilly Dally er et canadisk alt-rockband fra Toronto, Ontario, Canada. Bandet består af guitarist og vokalist Katie Monks, guitarist Liz Ball, bassist Jimmy Tony og trommeslager Benjamin Reinhartz.

Dilly Dally udgav deres debutalbum Sore den 9. oktober 2015 gennem Partisan Records og Buzz Records. Albummet var nomineret til Polaris Music Prize 2016, samt en Juno Award for 'Alternative Album of the Year' i 2017. Singlen 'Desire' vandt stort gennembrud hos fans verden over, og deres video til nummeret 'Snakehead' blev ligeledes nomineret til flere priser i 2017.

Bandet udgav deres andet album 'Heaven' i september 2018, til store roser hos anmelderne i ind og udland og til februar kan du endelig opleve den fremadstormende indie-kvartet i København når de gæster Loppén

DEL PÅ: [TWITTER](#) [FACEBOOK](#)

TJEK OGSÅ:

LINKS
DILLY DALLY FACEBOOK
TILMELD DIG FACEBOOK
EVENTET HER

vevo

vevo

 **LOPPÉN**
CHRISTIANIA

KALENDER GENRE BILLEDER INFO KONTAKT SØG

LØRDAG 02. FEB 2019 - KL 21.00 FORSALG: 75KR • KØB

BAFO DO MUNDO SAMBA DA GENTE DJ SIMON DE LA ONDA

SØNDAG 03. FEB 2019 - KL 20.30 FORSALG: 60KR • KØB

BC CAMPLIGHT (US)

ONSDAG 06. FEB 2019 - KL 20.30 FORSALG: 90KR • KØB

DEATH VALLEY GIRLS (US) BEFORE FIRE (DK)

TORS DAG 07. FEB 2019 - KL 20.30 FORSALG: 155KR • KØB

THE PAPER KITES (AU) BONNE NUIT JOHNNY

FREDAG 08. FEB 2019 - KL 21.00 FORSALG: 250KR • KØB

BOB MARLEY 74TH BIRTHDAY CELEBRATION - GOT TO HAVE KAYA 2019 FEAT: MACKA B (UK) & MUCH MORE

LØRDAG 09. FEB 2019 - KL 21.00 FORSALG: 260KR • KØB

JOHN GARCIA & THE BAND OF GOLD (US) DEAD QUIET (CA)

TIRSDAG 12. FEB 2019 - KL 20.30 FORSALG: 70KR • KØB

PILL (US)

ONSDAG 13. FEB 2019 - KL 20.30 FORSALG: 240KR • KØB

EKSTRAKONCERT DAS EFX (US) NATKAT

TORS DAG 14. FEB 2019 - KL 20.30 FORSALG: 80KR • KØB

DILLY DALLY (CA)

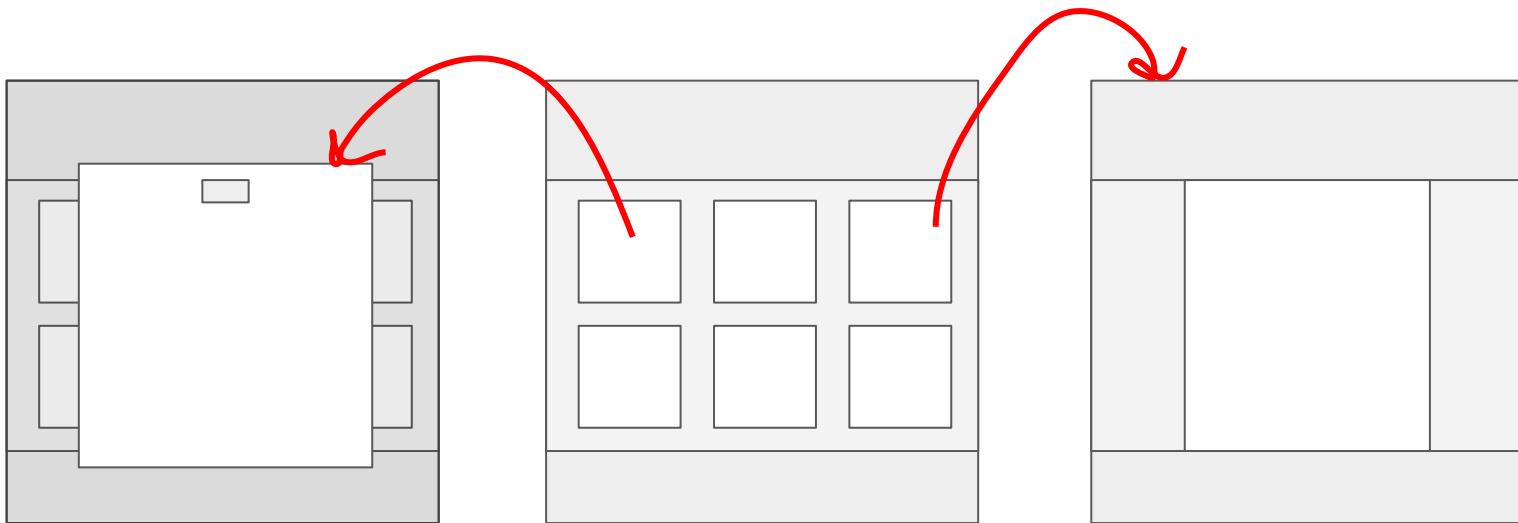
FREDAG 15. FEB 2019 - KL 21.00 UDSELGT

DAS EFX (US) NATKAT

LØRDAG 16. FEB 2019 - KL 21.00 FORSALG: 90KR • KØB

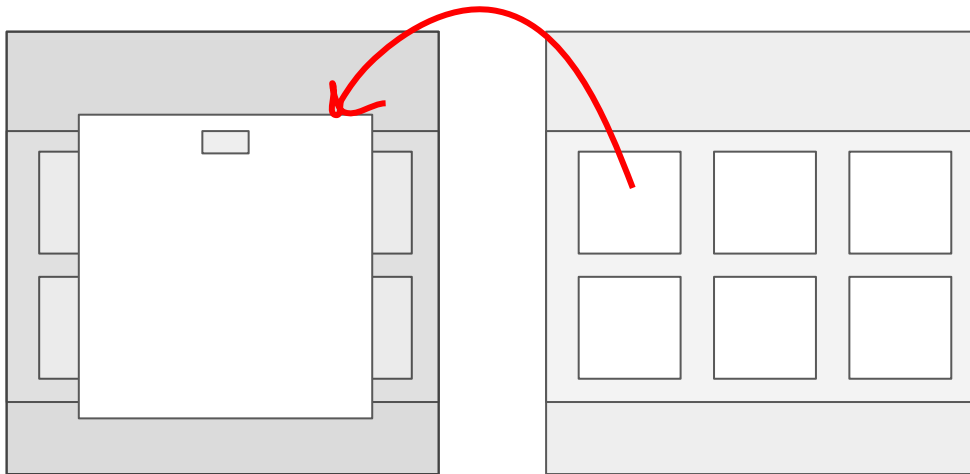
MOTORIQUE (DK)

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



Popup

single-view i popup



vis side med popup!

Fordele/ ulemper

Fordele:

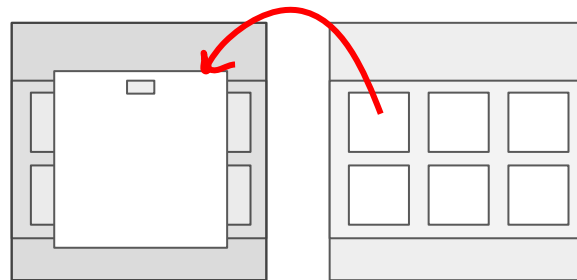
Giver en god brugeroplevelse, fordi der ikke skiftes side

Kan styles meget elegant med lightbox-effekter

Ulemper:

Single-viewet har ikke sin egen url-adresse

- kan ikke deles
- kan ikke findes af søgemaskiner



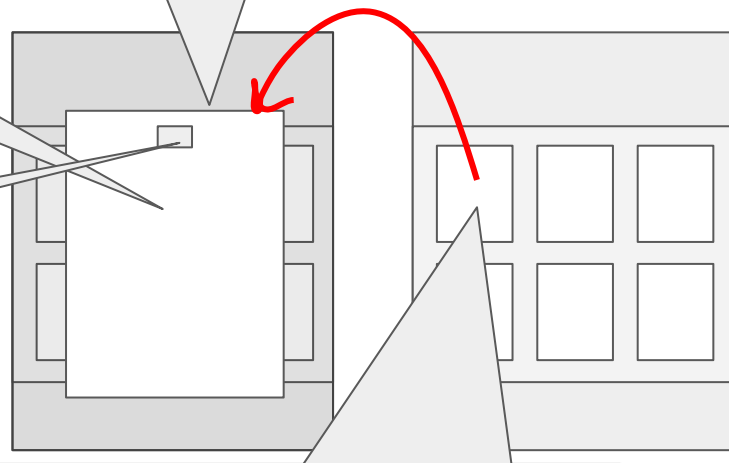
Popup-strategi

2. læg indholdselementer i popup'en

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

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



1. Lav popup

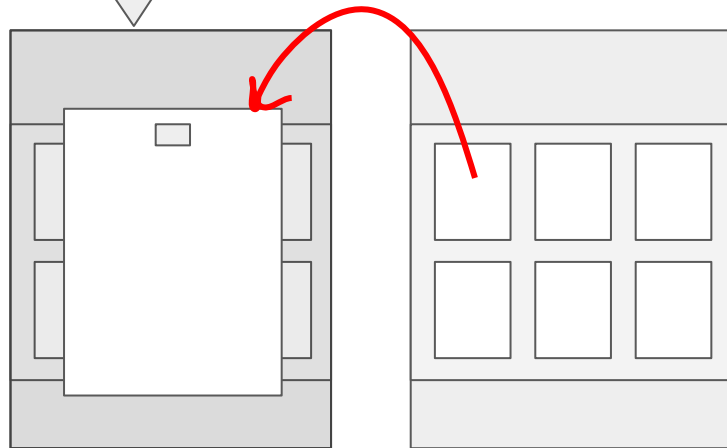
Man kan fx. lægge popup'en øverst i dokumentet

Det skal styles, så det fylder det hele (vw og vh: 100)

Det skal styles, så det lægger sig ovenpå det eksisterende indhold (position: fixed)

Giv den en baggrundsfarve, så man kan ses, mens vi udvikler.

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



```
21         .menu{text-align: center; margin-bot
22         .valgt{background-color: red;}
23         h1{text-align:center; }
24         #popup{width: 100vw;
25                 height:100vh;
26                 position: fixed;
27                 background-color: red;
28         }
29     </style>
30 </head>
31 <body>
32     <div id="popup"></div>
33     <header>
34         <h1>Alle</h1>
35         <nav class="menu">
```

Øvelse 1

Opret en ny mappe, **single**, i **tema5**-mappen.

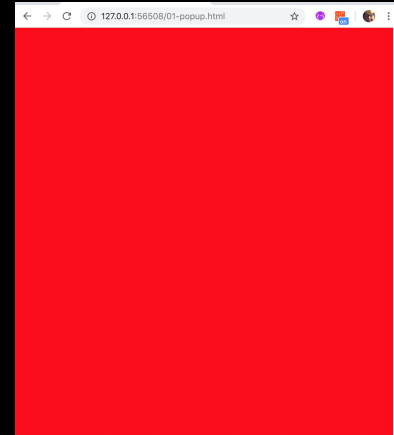
Gem en kopi af din **08-semantic.html** i mappen under navnet **01-popup.html** heri.

Læg et popup-vindue (et div-tag) øverst (eller nederst) i 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.

Test om det fungerer.

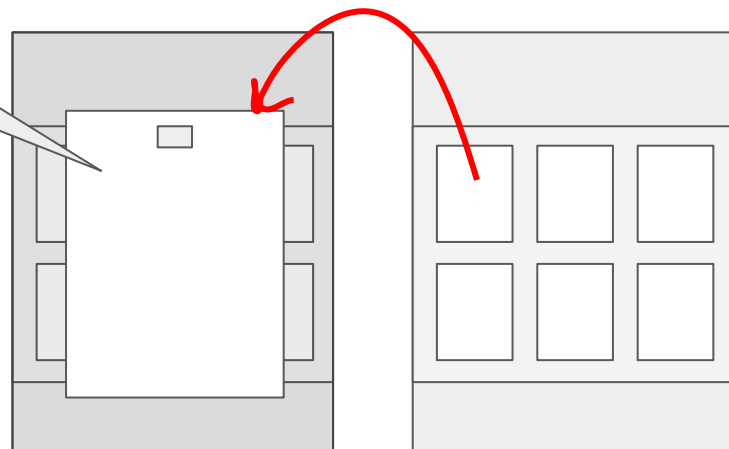


2. Indholdselementer i popup

2. læg indholdselementer i popup'en

Der skal lægges lidt indhold ind i popup-vinduet:

- En div med en luk-knap
- En div til indholdt
- Styling af indholds-div
- Style af luk-knap



```
26         width: 100vw;
27         height:100vh;
28         position: fixed;
29         background-color: red;
30     }
31     #indhold {
32         box-sizing: border-box; background-color:lightgray;
33         width: 90vw; height: 90vh; margin:auto;
34     }
35     #luk{box-sizing: border-box;text-align: right; padding-right: 1em;}
36     #luk button{font-size: 2em;}
37     html{overflow-y: hidden;}
38 </style>
39 </head>
40 <body>
41     <div id="popup">
42         <div id="luk"><button>X</button></div>
43         <div id="indhold"></div>
44     </div>
45 </body>
```

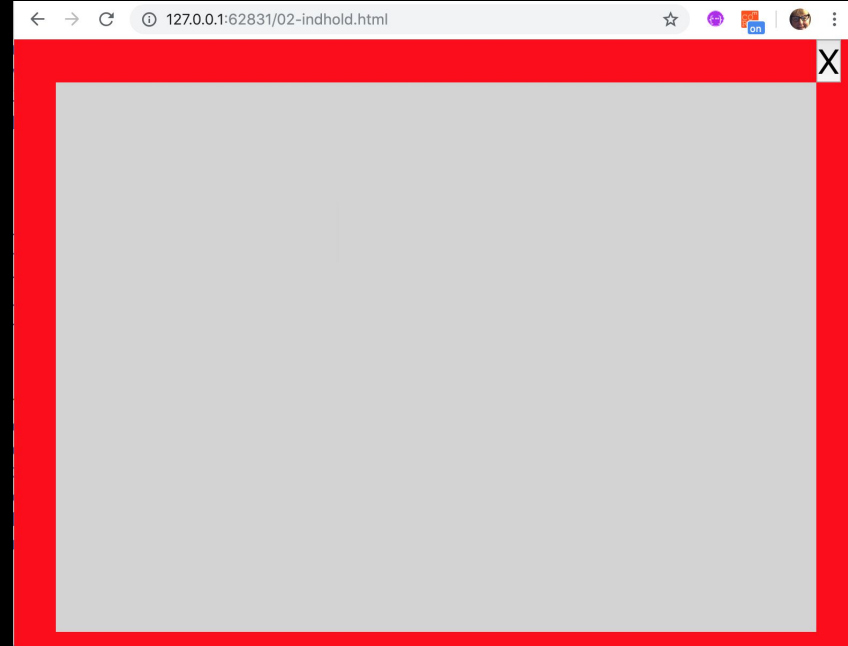
Øvelse 2

Gem en kopi af **01-popup.html** i mappen under navnet **02-indhold.html**.

Læg div'er i popup'en til indhold og lukkeknop

Style, så vinduet bliver funktionelt

Style til sidst popup'en, så den er skjult

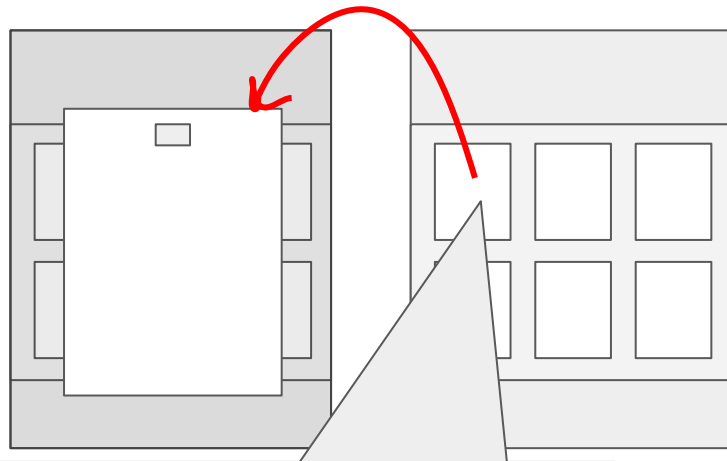


3. Eventhandlers på alle person-elementer i loop-view

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(){
  dest.innerHTML="";
  personer.forEach(person =>{
    if(filter=="alle" || filter==person.hold){
      let template= `
        <article class="person">
          <h2>${person.navn}</h2>
          
          <p>
            Github: <a href="http://github.com/${person.github}" target="_blank">
              ${person.github}
            </a>
          </p>
          <p>Hold: ${person.hold}, Alder:${person.alder} </p>
        </article>
      `;
      dest.insertAdjacentHTML("beforeend",template);
      dest.lastElementChild.addEventListener("click", åbn);
      function åbn(){//her skal udfyldes
      }
    }
  })
} //function visPersoner slut
```

insertAdjacentHTML,
ikke innerHTML

(når der er eventListeners)

dest.lastElementChild er det article-element,
som lige er blevet tilføjet til dest

insertAdjacentHTML

```
dest.insertAdjacentHTML("beforeend", template);
```

Når vi skal lægge indhold i et container-element, har vi i har tidligere brugt: `textContent`, som kun lægger tekst ind i elementet `innerHTML`, som kan bruges til at lægge html-kode ind i elementet nu er vi kommet til:

`insertAdjacentHTML`, som også kan lægge html-kode ind i et element, men som kan bruges, når der er eventListeners på noget af det eksisterende kode (bruger man `innerHTML`, forsvinder eventListeners)

<https://developer.mozilla.org/en-US/docs/Web/API/Element/insertAdjacentHTML>

```

72 ▼    function visPersoner(){
73        dest.innerHTML="";
74 ▼        personer.forEach(person =>{
75 ▼            if(filter=="alle" || filter==person.hold){
76                let template= `
77                    <article class="person">
78                        <h2>${person.navn}</h2>
79                        
80                        <p>
81                            Github: <a href="http://github.com/${person.github}" target="_blank">
82                                ${person.github}
83                            </a>
84                        </p>
85                        <p>Hold: ${person.hold}, Alder:${person.alder} </p>
86                    </article>
87                `;
88                dest.insertAdjacentHTML("beforeend",template);
89                dest.lastElementChild.addEventListener("click", åbn);
90 ▼            function åbn(){
91                document.querySelector("#indhold").innerHTML= `
92                    <article class="person">
93                        <h2>${person.navn}</h2>
94                        
95                        <p>Hold: ${person.hold}, Alder:${person.alder} </p>
96                        <h3>${person.navn}'s historie</h3>
97                        <p>${person.historie} </p>
98                    </article>
99                `;
100                document.querySelector("#popup").style.display="block";
101            }
102        }
103    })
104    } //function visPersoner slut

```

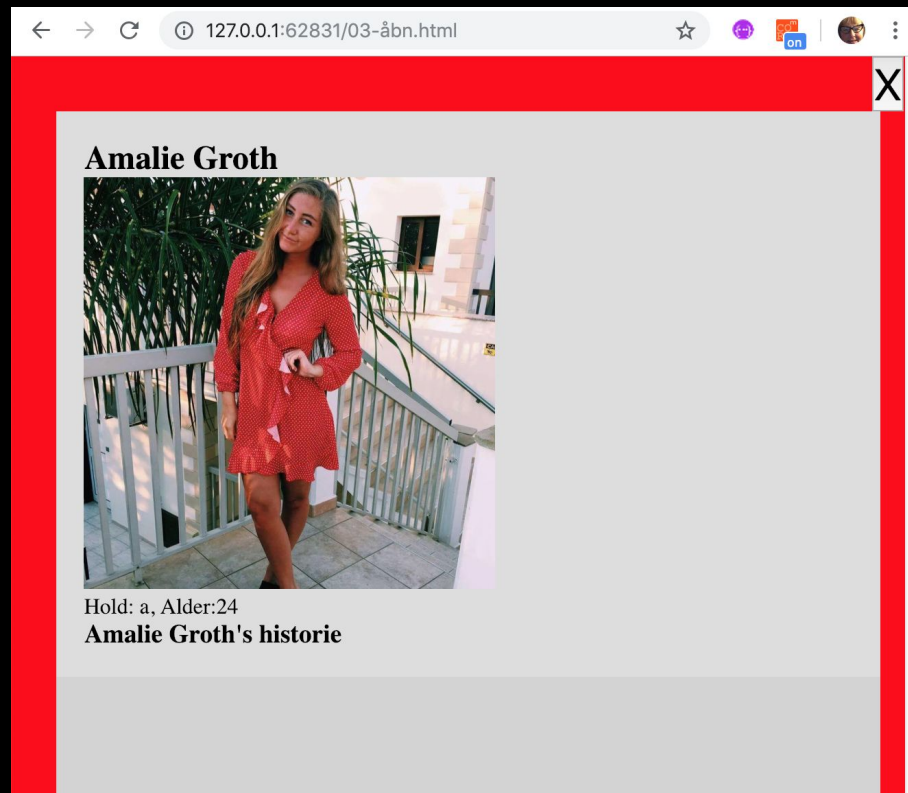
```
function åbn(){
  document.querySelector("#indhold").innerHTML= `
    <article class="person">
      <h2>${person.navn}</h2>
      
      <p>Hold: ${person.hold}, Alder:${person.alder} </p>
      <h3>${person.navn}'s historie</h3>
      <p>${person.historie} </p>
    </article>
  `;
  document.querySelector("#popup").style.display="block";
}
```

NB: vi kunne også have brugt en arrow-funktion.
Til gengæld kan function åbn() genbruges.

Øvelse 3

Gem en kopi af din **02-indhold.htm** i mappen under navnet **03-åbn.html**.

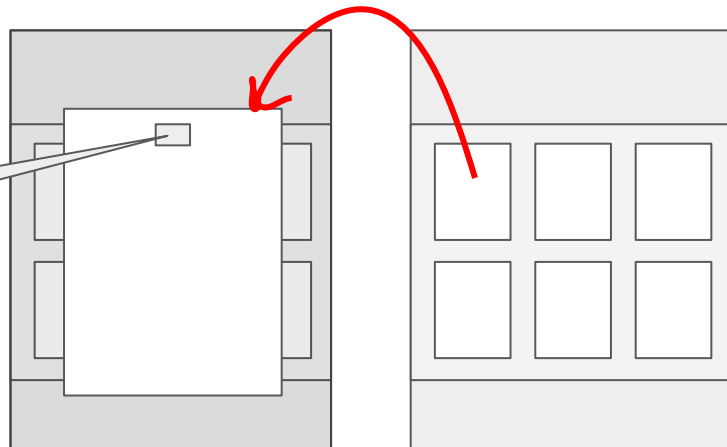
Når man klikker på en person i loop-view, skal popup med single-view åbnes.



4. Luk-knappen

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

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



```

function visPersoner () {
  dest.innerHTML="";
  personer.forEach(person =>{
    if(filter=="alle" || filter==person.hold){
      let template= `
        <article class="person">
          <h2>${person.navn}</h2>
          
          <p>
            Github: <a href="http://github.com/${person.github}" target="_blank">
              ${person.github}
            </a>
          </p>
          <p>Hold: ${person.hold}, Alder:${person.alder} </p>
        </article>
      `;
      dest.insertAdjacentHTML("beforeend",template);
      dest.lastElementChild.addEventListener("click", åbn);
      function åbn(){
        document.querySelector("#indhold").innerHTML= `
          <article class="person">
            <h2>${person.navn}</h2>
            
            <p>Hold: ${person.hold}, Alder:${person.alder} </p>
            <h3>${person.navn}'s historie</h3>
            <p>${person.historie} </p>
          </article>
        `;
        document.querySelector("#popup").style.display="block";
      }
    }
  })
} //function visPersoner slut

document.querySelector("#luk button").addEventListener("click", ()=>{
  document.querySelector("#popup").style.display="none";
})

```



```
    } // function visitor setter state  
    document.querySelector("#luk button").addEventListener("click", ()=>{  
        document.querySelector("#popup").style.display="none";  
    })
```

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)

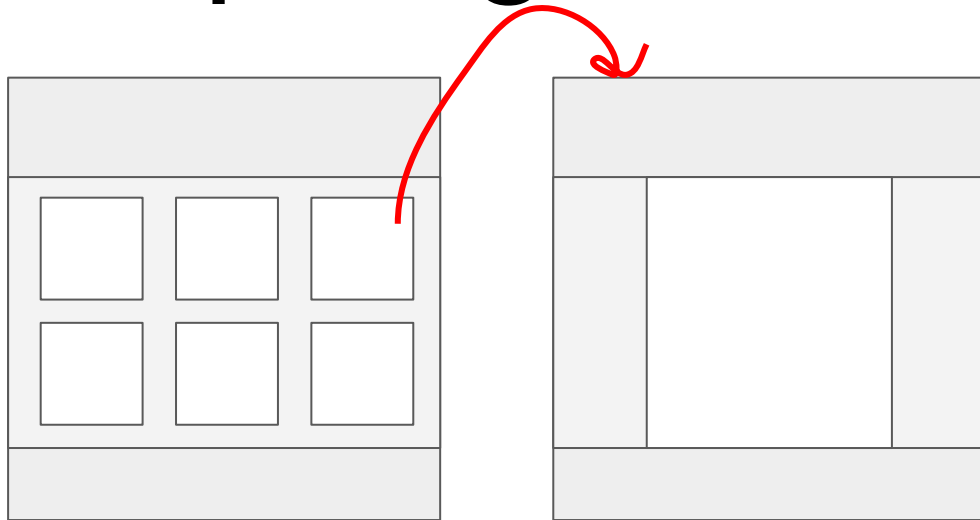
Lav en eventListener på knappen, som sørger for at lukke popupvinduet igen.

Så er popup-vinduet færdigt - bortset måske fra lidt ekstra styling - det kommer i til i næste uge



Single-view på egen side

Single-view på egen 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

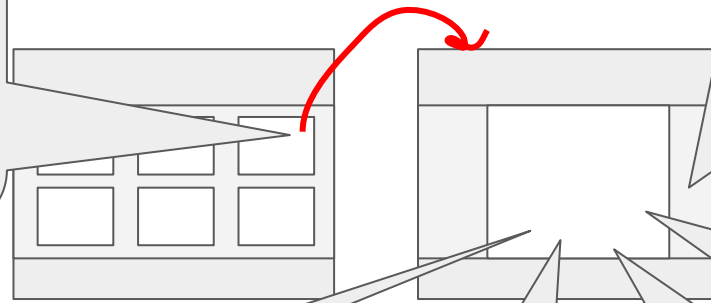
Single-side strategi

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

En variabel skal medbringes

2. Lav **single.html**:

(html og styling som #popup men med nyt navn: **#singleview**)
luk-knappen skal laves om til en **tilbageknop**



3. hent **url-variablen**

4. hent alle personer og find den rigtige

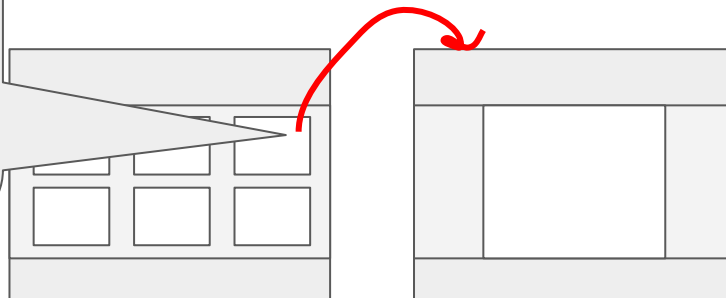
5. **udfyld #singleview**
(genbrug function **åbn()**)

6. Få tilbageknappen til at virke

1. kald med url-variabel

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

En variabel skal medbringes



1. kald med url-variabel

I loop-view:

Eventlisteners på alle elementer.

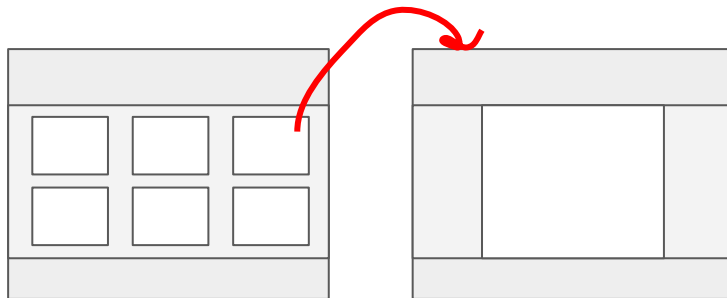
eventhandler:

spring over til den nye side.

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

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

Variablen kan sendes som **url-variable**



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:

<http://helf-kea.dk/test.html?navn=Flemming>

vil browseren hente siden <http://helf-kea.dk/test.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.

hvilken 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**.

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

```

27 > <body>
28 >   <header>
29 >     <h1>Alle</h1>
30 >     <nav class="menu"> ... </nav>
35 >   </header>
36 >   <main > ... </main>
39 >   <script>
40 >     let personer=[];
41 >     let filter="alle";
42 >     document.addEventListener("DOMContentLoaded", start);
43 >     function start(){
44 >       let dest=document.querySelector("#liste");
45 >       async function getJson(){ ... }
51 >       function visPersoner(){
52 >         dest.innerHTML="";
53 >         personer.forEach(person =>{
54 >           if(filter=="alle" || filter==person.hold){
55 >             let template= `
56 >               <article class="person">
57 >                 <h2>${person.navn}</h2>
58 >                 
59 >                 <p>
60 >                   Github: <a href="http://github.com/${person.github}" target="_blank">
61 >                     ${person.github}
62 >                 </a>
63 >                 </p>
64 >                 <p>Hold: ${person.hold}, Alder:${person.alder} </p>
65 >               </article>
66 >             `;
67 >             dest.insertAdjacentHTML("beforeend",template);
68 >             dest.lastElementChild.addEventListener("click",()=>{
69 >               location.href="06-single.html?github="+person.github;
70 >             });
71 >           }
72 >         })
73 >       } //function visPersoner slut
74 >
75 >       document.querySelectorAll(".filter").forEach(elm =>{ ... })
76 >       function filtrering(){ ... }
77 >       getJson();
78 >     }
79 >   </script>
80 > </body>

```

```
dest.lastElementChild.addEventListener("click", ()=>{  
    location.href="06-single.html?github="+person.github;  
});
```

Bliver fx til:

`location.href = "06-single.html?github=helefred"`

Øvelse 5

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

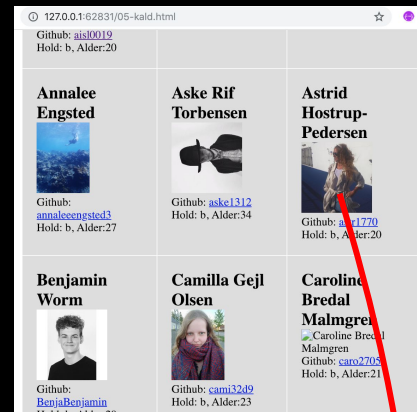
Fjern al html og css, som har med popup-vinduet at gøre.

Fjerne også i scriptet alt, hvad der har med popup'en at gøre: variablen popup, eventlisteners på personer og på lukke-knapper.

Læg i stedet en eventlistener på alle personer, som får browseren til at hoppe til **06-single.html** medbringende en variabel, id - hvilket nummer i arrayet person.

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

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



2. ny side: single.html

Lav en single-side. Tag udgangspunkt i en side med popup-vindue (fx. 03-åbn.html):

I html'en:

- væk med menu og liste
- bevar kun popup-vinduet, men kald det nu `#singleview`

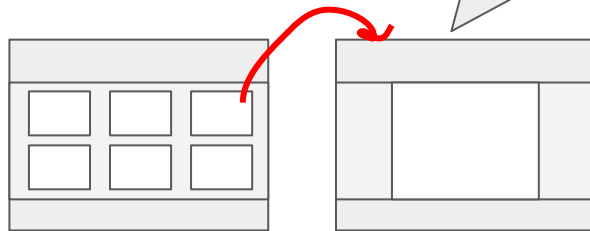
I css'en:

- bevar kun styling omkring popup'en - dog skal den ikke længere være skjult eller have `position: fixed`

I scriptet:

- bevar function `åbn()`, men slet resten af `visPersoner()`
- i `getJSON`: kald `åbn()` - ikke resten af `visPersoner()`
- slet eventhandlers til luk
- slet alt om filtrering

2. Vi laver en separat side, **single.html**.
Html og css som popup-vinduet



```

<style>
  html, body,*{margin:0; padding:0}
  #singleview{
    box-sizing: border-box;
    width: 100vw;
    height:100vh;
    position: fixed;
    background-color: red;
  }
  #indhold {
    box-sizing: border-box; background-color:lightgray;
    width: 90vw; height: 90vh; margin:auto;
  }
  #luk{box-sizing: border-box;text-align: right; padding-right: 1em;}
  #luk button{font-size: 2em;}
  #indhold img{height: 300px;}
</style>
</head>
<body>
  <div id="singleview">
    <div id="luk"><button>X</button></div>
    <div id="indhold"></div>
  </div>
  <script>
    let personer=[];
    let person;
    document.addEventListener("DOMContentLoaded", start);
    function start(){
      let dest=document.querySelector("#liste");
      async function getJson(){
        let jsonData = await fetch("https://mandalskeawebspace.dk/claude_php/clean_up_sprea
        personer=await jsonData.json();
        personer.sort((a, b) => {return a.navn.localeCompare(b.navn);})
        åbn();
      }
      function åbn(){
        document.querySelector("#indhold").innerHTML= `
        <article class="person">
          <h2>${person.navn}</h2>
          
          <p>Hold: ${person.hold}, Alder:${person.alder} </p>
          <h3>${person.navn}'s historie</h3>
          <p>${person.historie} </p>
        </article>
        `;*/
      }
      getJson();
    }
  </script>
</body>

```


Øvelse 6

Gem en kopi af **03-åbn.html** under navnet **06-single.html**.

Fjern html og styling, som ikke har med popup at gøre.

Omdøb #popup til #singleview (også i styling)

Tilpas stylingen - ikke længere fixed eller skjult.

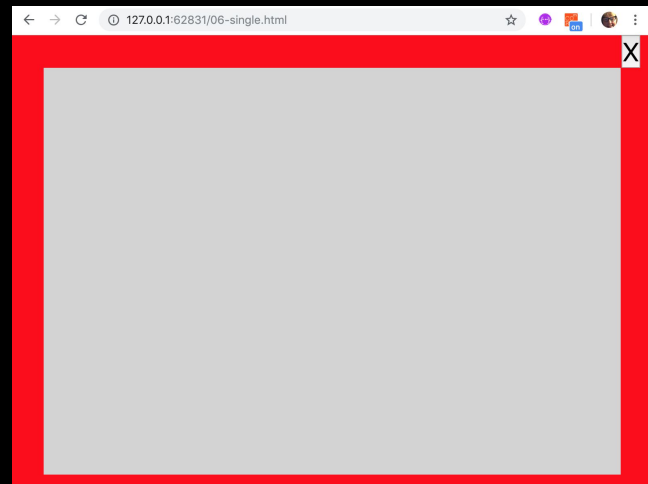
I scriptet:

fjern visPersoner(), men bevar åbn(), som ligger i den

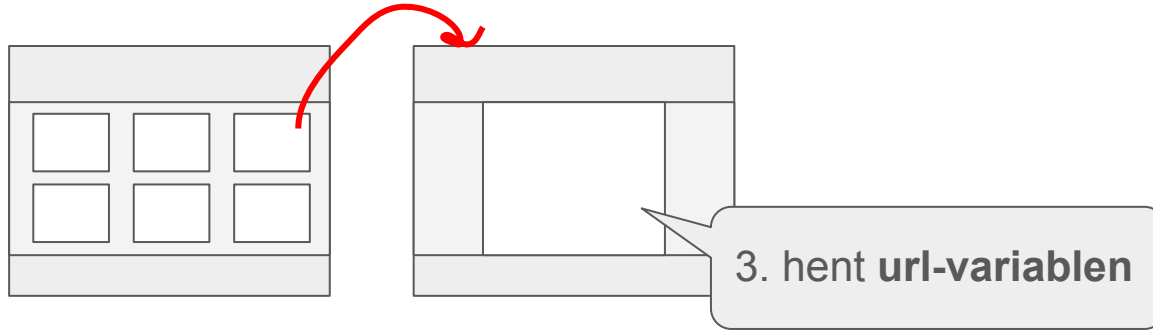
- udkommenter indhold i åbn()

fjern eventlistener/handler for luk-knap

fjern alt om filtrering



3. Hent url-variabel

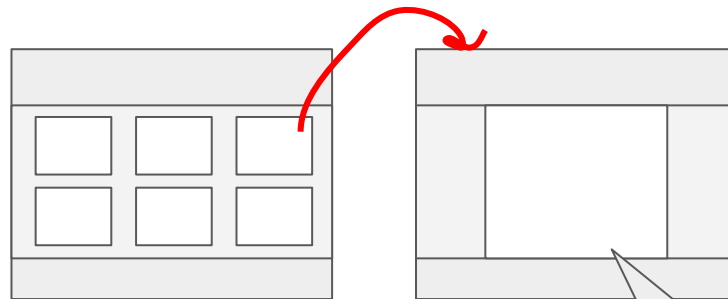


3. Få fat i url-bariablen med URLSearchParams

```
let urlParams = new URLSearchParams(window.location.search)  
let github = urlParams.get("github");
```

```
<body>
  <div id="singleview">
    <div id="tilbage"><button><-</button></div>
    <div id="indhold"></div>
  </div>
  <script>
    let urlParams = new URLSearchParams(window.location.search);
    let github = urlParams.get("github");
```

4. Find den rigtige person



4. hent alle personer
og find den rigtige

Find den rigtige person

Når alle personer er hentet ind i `getJSON`, skal vi finde den person, som har den rigtige github-konto.

Vi kan bruge en `forEach`-løkke til at løbe arrayet igennem.

Vi skal også bruge en global variabel, `person`, til at gemme det rigtige objekt i

```
let person;  
personer.forEach(obj =>{  
    if(obj.github==github){  
        person=obj;  
    }  
})
```

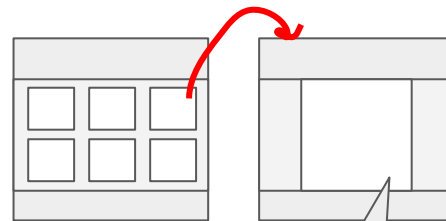
```
async function getJson(){  
  let jsonData = await fetch("https://mandalskeaweb.space.dk/claude_php/clean_up  
  personer=await jsonData.json();  
  personer.forEach(obj =>{  
    if(obj.github==github){  
      person=obj;  
    }  
  })  
  åbn();  
}
```

NB: jeg bruger variabel navnet obj i forEach-loopet,
fordi jeg nu har en global variabel, som hedder person
De to variabler kan ikke hedde det samme.

5. udfyld #singleview

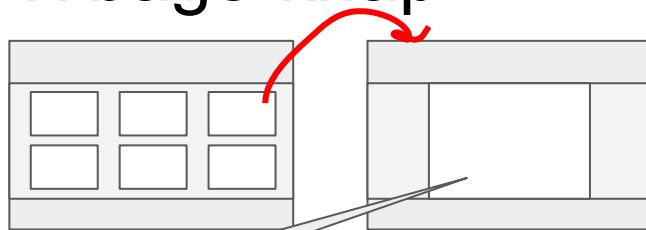
```
async function getJson(){
  let jsonData = await fetch("https://mandalskeawebspace.dk,
  personer=await jsonData.json();
  personer.forEach(obj =>{
    if(obj.github==github){
      person=obj;
    }
  })
  åbn();
}
```

```
function åbn(){
  document.querySelector("#indhold").innerHTML= `
    <article class="person">
      <h2>${person.navn}</h2>
      
      <p>Hold: ${person.hold}, Alder:${person.alder} </p>
      <h3>${person.navn}'s historie</h3>
      <p>${person.historie} </p>
    </article>
  `
}
```



5. udfyld #singleview
(udgangspunkt i åbn())

6. Eventlistener og eventhandler til Tilbage-knap



6. Få tilbageknappen
til at virke

```
document.querySelector("#tilbage button").addEventListener("click", ()=>{  
    location.href="05-kald.html"  
})
```

```
<script>
```

```
let urlParams = new URLSearchParams(window.location.search);
```

```
let github = urlParams.get("github");
```

```
let personer;
```

```
let person;
```

```
document.addEventListener("DOMContentLoaded", start);
```

```
function start(){
```

```
  async function getJson(){
```

```
    let jsonData = await fetch("https://mandalskeawebspace.dk/claude_php/clean_up_spre:
```

```
    personer=await jsonData.json();
```

```
    personer.forEach(obj =>{
```

```
      if(obj.github==github){
```

```
        person=obj;
```

```
      }
```

```
    })
```

```
    åbn();
```

```
  }
```

```
function åbn(){
```

```
  document.querySelector("#indhold").innerHTML= `
```

```
    <article class="person">
```

```
      <h2>${person.navn}</h2>
```

```
      
```

```
      <p>Hold: ${person.hold}, Alder: ${person.alder} </p>
```

```
      <h3>${person.navn}'s historie</h3>
```

```
      <p>${person.historie} </p>
```

```
    </article>
```

```
`
```

```
}
```

```
document.querySelector("#tilbage button").addEventListener("click", ()=>{
```

```
  location.href="05-kald.html"
```

```
})
```

```
getJson();
```

```
}
```

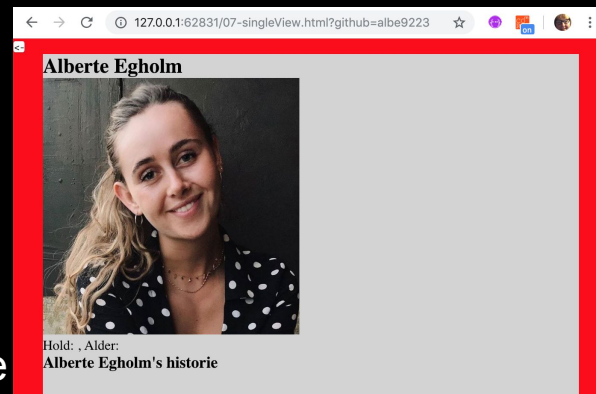
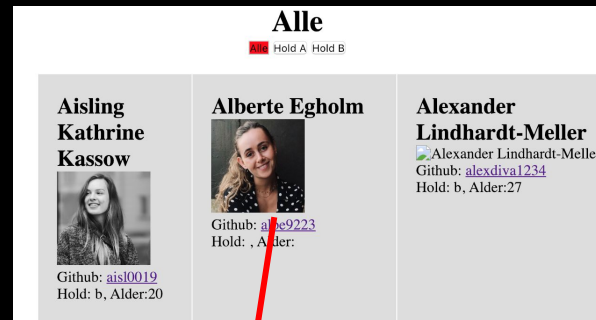
```
</script>
```

Øvelse 7

Arbejd videre med **06-single.html** i en ny kopi, **07-singleView.html**

I scriptet skal du :

- hente url-variablen ved hjælp af URLSearchParams.
- Erklære en variabel, person, globalt, og i hentJson slå den op i arrayet personer vha filter-funktionen
- Erstat luk-knappen med en tilbage-knap, som kan springe tilbage til loop-view-siden.(
location.href="05-kald.html"
- husk, når du tester, at **06-singleView.html** skal testes ved at klikke på en person i **05-kald.html**, hvor du skal huske at hoppe til **06-singleView.html** og ikke **06-single.html** ved klik på en person



Babushka og singleview - hvad skal I lave

I skal lave **to løsninger**:

1. En Babushka-side med popup-singleview
2. En Babushka-side, som åbner en ny side til single-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.