

# FILTRERING

...

# Loop-view kontra single-view

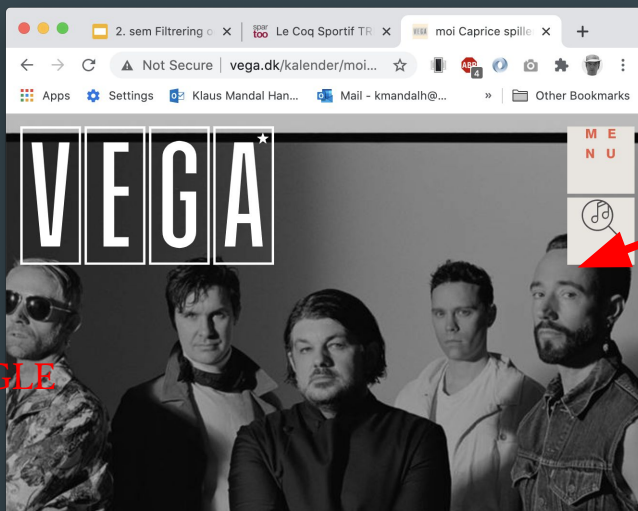
Loop-view er en oversigtsvisning af mange objekter (en liste)

Single-view er en detaljeret visning af ét objekt

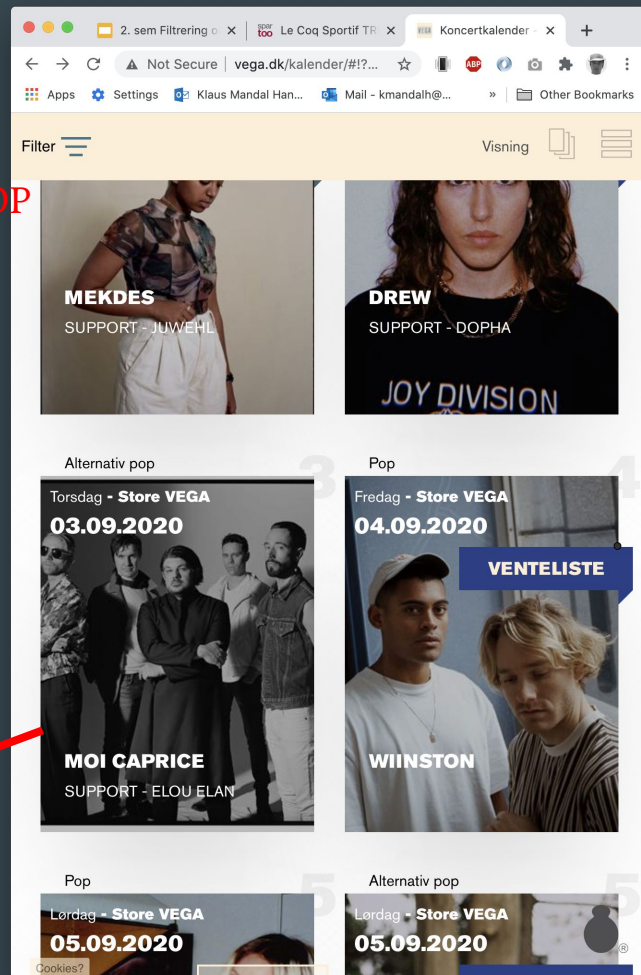
I dag arbejder vi med loop-views

I morgen med single-view

SINGLE



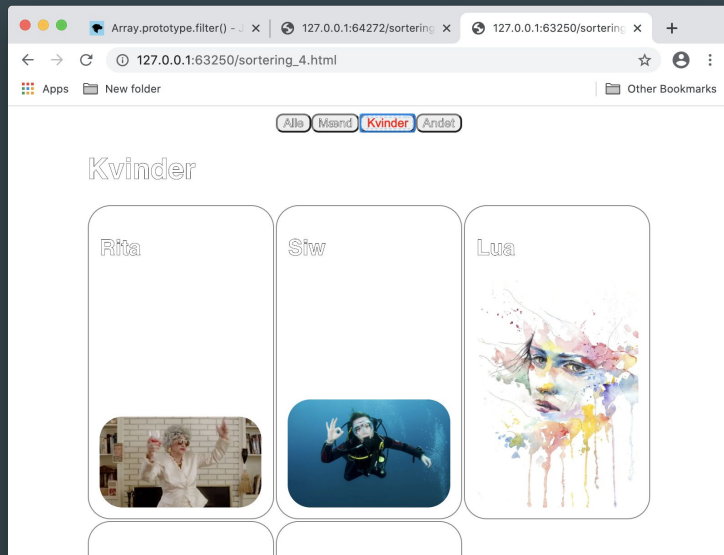
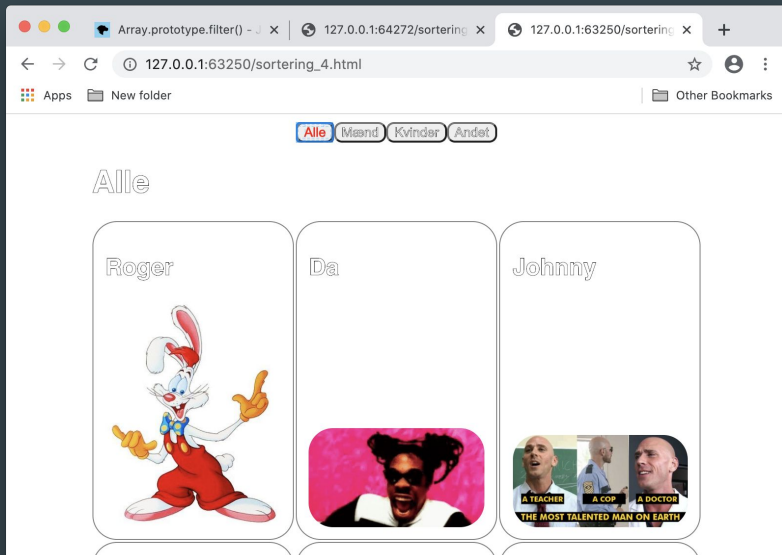
LOOP



# Loop-view med filtrering

`personliste.html` er et loop-view, som indeholder navn, billede og valgfrie kolonner fra googlesheetet.

Vi vil indføre **filtrering** på siden: mulighed for kun at se ét køn:



# Hvad er filtrering?

De enkelte objekter kan have egenskaber, hvor værdierne tilhører nogle få kategorier.

Disse kategorier kan man i loop-views **filtrere** på, så man kun ser objekter med en bestemt værdi.

Personerne i personlisten har egenskaben **sex (køn)**, som kan have værdierne **m**, **k** eller **andet**.

Derfor kan vi i loop-viewet filtrere på sex (køn).

# Hvad der skal gøres

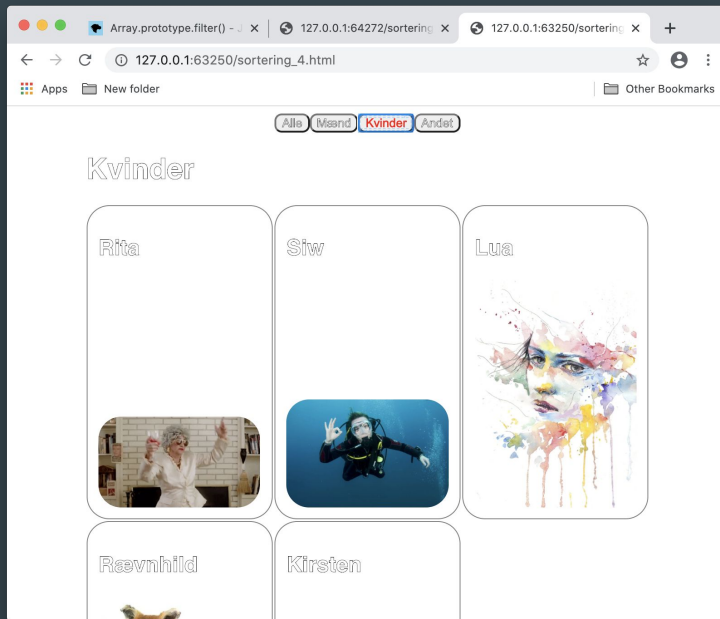
1. **Scriptet skal kunne vise et køn ad gangen.**
2. Der skal være filter-knapper til at vælge med.
3. Knapperne skal virke
4. Der skal være en overskrift, som viser det valgte
5. Den aktuelle knap skal være fremhævet.

# Scriptet skal kunne vise et køn ad gangen

Vi kan indføre en **variabel**, **filter**, som sættes til værdien af et køn, fx. "k"

I vise-funktionens vil vi kun vise objekter med denne værdi.

I forEach-loopen i vise-funktionen kan vi tjekke hvert objekt med if, om det har `køn=="k"`. Vi viser så kun kvinder.



```
<script>
```

```
let personer;  
let filter = "k";  
document.addEventListener("DOMContentLoaded", loadJSON)
```

Vi skal lave en variabel, *personer*,  
der indeholder vores JSON

```
async function loadJSON() {  
  const JSONData = await  
  fetch("https://spreadsheets.google.com/feeds/list/1mtHhpoEkYqVqr1TSnKQ1_sQZWI7FRVjcw3H1oatD0VQ/od6/public/values?alt=json");  
  personer = await JSONData.json();  
  visPersoner();  
}
```

Vi skal lave en variabel, *filter*, der  
indeholder det vi vil filtrere på.  
Vi sætter den til "k" og  
sammenligner den med værdien  
af personens køn fra google  
sheet'et

```
function visPersoner() {  
  const templatePointer = document.querySelector("template");  
  const listPointer = document.querySelector("#list");  
  personer.feed.entry.forEach(person => {  
    if (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;  
      listPointer.appendChild(minKlon);  
    }  
  })  
}
```

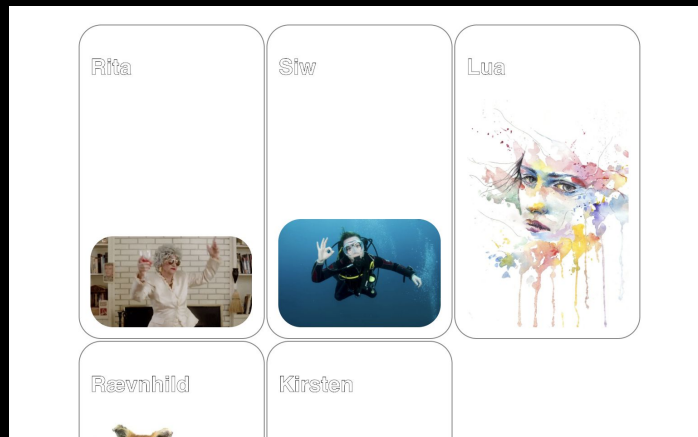
```
}
```

```
</script>
```

# Øvelse 1

Gem en kopi af [02-personliste.html](#) fra i fredags i en ny undermappe til [tema7](#)-mappen, [filtreringSortering](#). Omdøb filen til [01-kvinder.html](#).

Lav om på scriptet i filen, så der kun vises kvinder.





# Hvad der skal gøres

1. Scriptet skal kunne vise et køn ad gangen.
2. **Der skal være filter-knapper til at vælge med.**
3. Knapperne skal virke
4. Der skal være en overskrift, som viser det valgte
5. Den aktuelle knap skal være fremhævet.

## 2. Der skal være filter-knapper

Det skal være en menu med en knap for hvert køn og en knap for alle.

Det kan være **button-tags** `<button>...</button>`

Knapperne skal have tilknyttet den værdi, som filter-variablen skal sættes lig med.

Data-attributten kan bruges.

Vi skal fx have en **data-køn** attribut.

Den vil så kunne tilgås i javascript med **dataset.køn**

# Data attributten

```
<header>
  <button class="filter valgt" data-køn="alle">Alle</button>
  <button class="filter" data-køn="m">Mænd</button>
  <button class="filter" data-køn="k">Kvinder</button>
  <button class="filter" data-køn="andet">Andet</button>
</header>
```

Læg mærke til **data-køn**.

**data-** er en ny attribut-type, som kan bruges til at tilknytte noget data til et element. Værdien af attributten kan vi bruge i scriptet.

## Øvelse 2: knapper til valg

Gem **01-filtrering.html** i en ny kopi, **02-knapper.html**

Tilføj knapper til filtrering.

Lav styling af knapperne, så de kommer til at stå centreret.

Tilføj en klasse, **valgt**, til kvinde-knappen, og style', så knappen træder tydeligt frem.



# Hvad der skal gøres

1. Scriptet skal kunne vise et køn ad gangen.
2. Der skal være filter-knapper til at vælge med.
- 3. Knapperne skal virke**
4. Der skal være en overskrift, som viser det valgte
5. Den aktuelle knap skal være fremhævet.

### 3. Knapperne skal virke

1. I DOM'en er det **alle-knappen**, som skal have klassen valgt som udgangspunkt.
2. Variablen **filter** skal sættes til **“alle”**, for alle personer skal vises først.
3. Vi skal lægge **eventListeners** på alle knapper - så der udføres en funktion ved klik.
4. **Eventhandler**-funktionen skal finde den **værdi**, der ligger i knappens **data-attribut**.
5. Denne **værdi** skal **filter-variablen** sættes lig med.
6. Så skal **visPersoner**-funktionen kaldes på ny.
7. **visPersoner** skal starte med at **fjerne det eksisterende indhold fra listen i html**.

```

<script>
  let personer;
  let filter = "alle";
  document.addEventListener("DOMContentLoaded", loadJSON)

  async function loadJSON() {
    const JSONData = await
    fetch("https://spreadsheets.google.com/feeds/list/1mtHhpoEkYqVqr1TSnKQ1_sQZWI7FRVjCw3H1oafDovQ/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 (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;
        listPointer.appendChild(minKlon);
      }
    })
  }

  function addEventListenersToButtons() {
    document.querySelectorAll(".filter").forEach((btn) => {
      btn.addEventListener("click", filterBTNs);
    });
  }

  function filterBTNs() {
    filter = this.dataset.køn;
    visPersoner();
  }

```

Slet indholdet fra listen, altså det HTML element du appender template-kloner til

Eventlisteners på alle knapper husk at kalde denne funktion

Sæt filter-variablen lig med værdien af data-køn- attributten for den valgte knap. Når det er en data attribut kan den tilgås direkte i JS med dataset. *data-noget* i html bliver til *dataset.noget* i JS.

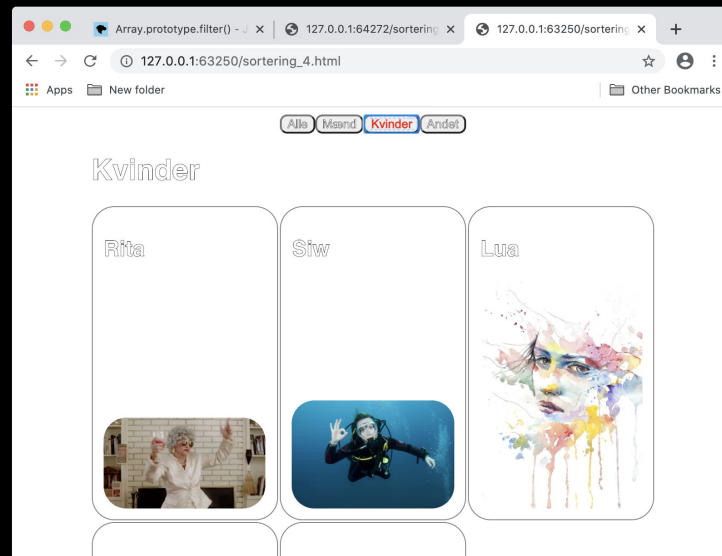
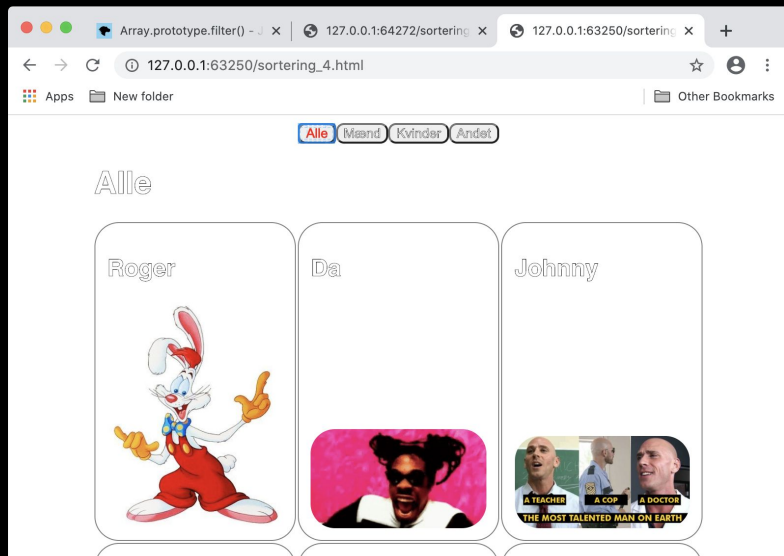
Eventhandler

Kald visPersoner

# Øvelse 3: knapper til valg

Gem **02-knapper.html** i en ny kopi, **03-virker.html**

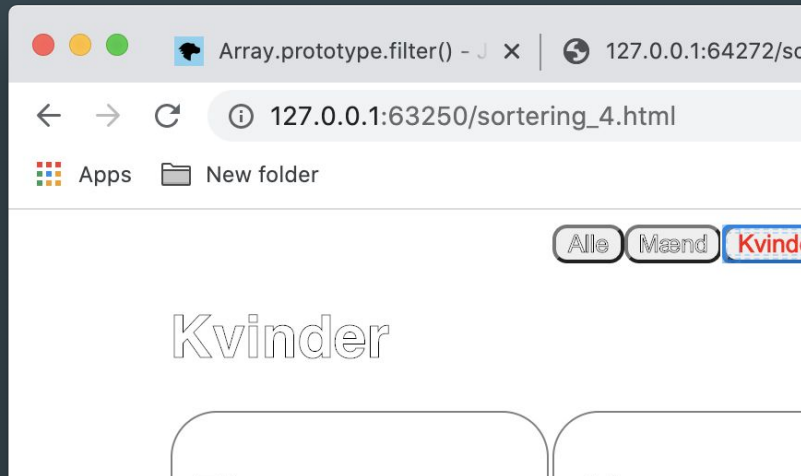
Få filtreringsknapperne til at virke!





# Hvad der skal gøres

1. Scriptet skal kunne vise et køn ad gangen.
2. Der skal være filter-knapper til at vælge med.
3. Knapperne skal virke
4. **Der skal være en overskrift, som viser det valgte**
5. Den aktuelle knap skal være fremhævet.



## 4. Det skal være en overskrift, som viser det valgte

Tilføj en **h1**-overskrift over filter-menuen med teksten “Alle”  
(jeg er i screendumpsene i denne pdf kommet til at lægge overskriften **under** filter-menuen)

Den skal styles, så den måske er centreret?

Ved knap-klik, skal **eventhandler-funktionen** sørge for, at den overskriften bliver sat til at være det samme, som der står inde i knappen.

```
<h1>Alle</h1>
```

Tilføj h1-overskrift til at vise valgt filter

```
<header>
```

```
  <button class="filter valgt" data-køn="alle">Alle</button>
```

```
  <button class="filter" data-køn="m">Mænd</button>
```

```
  <button class="filter" data-køn="k">Kvinder</button>
```

```
  <button class="filter" data-køn="andet">Andet</button>
```

```
</header>
```

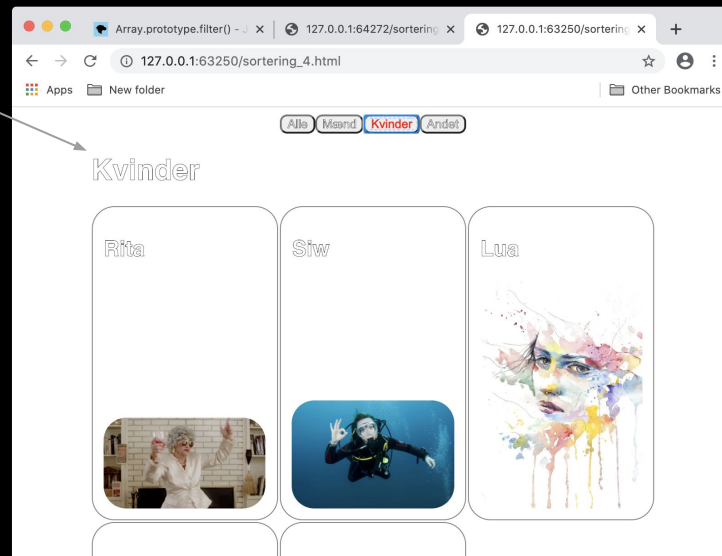
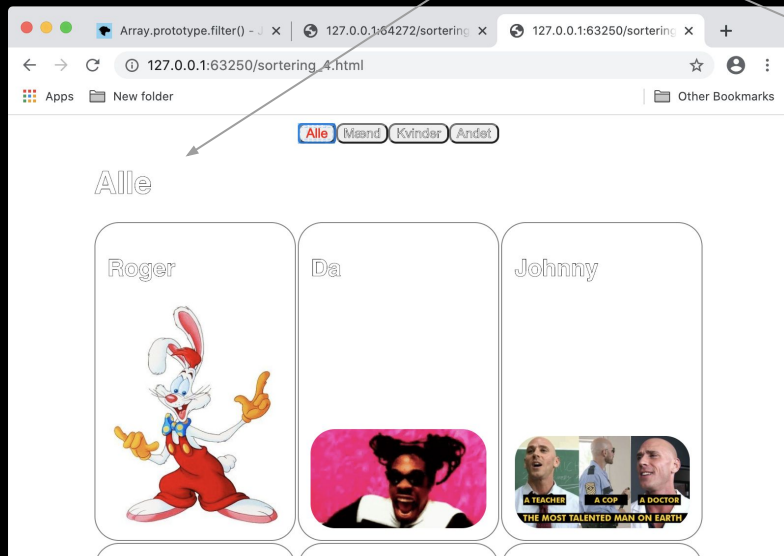
Når der klikkes på en knap, sørger eventhandler-funktionen for, at h1-overskriftens indhold opdateres.  
*this* henviser til den button der er klikket på... prøv evt. At *console.log(this)*

```
function filterBTN() {  
  filter = this.dataset.køn;  
  document.querySelector("h1").textContent = this.textContent;  
  visPersoner();  
}
```

# Øvelse 4: Overskrifter

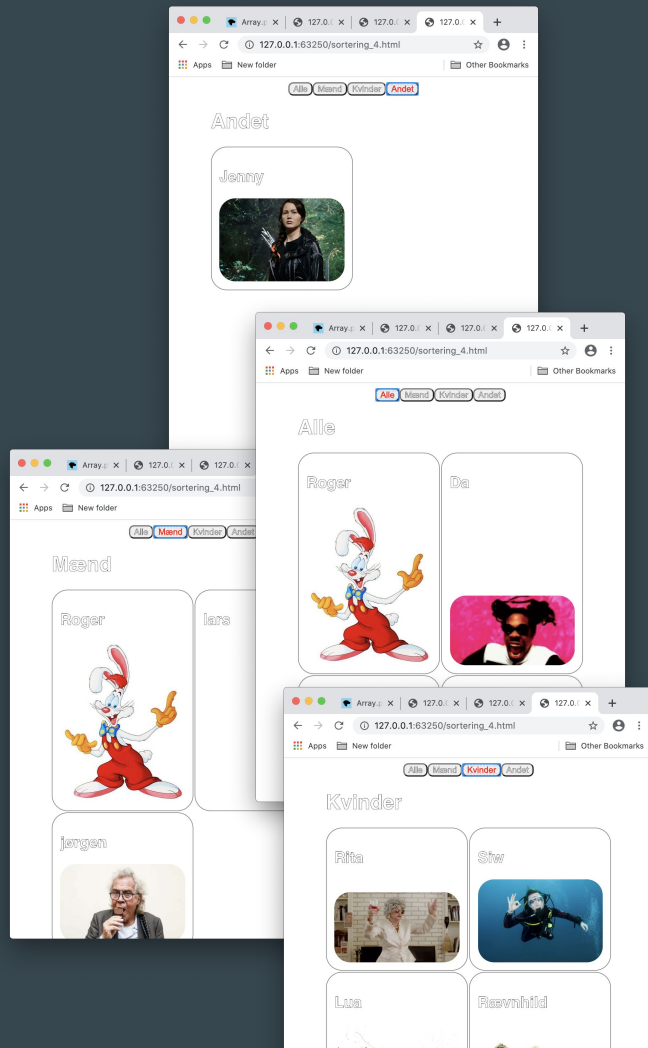
Gem **03-virker.html** i en ny kopi, **04-overskrift.html**

Få overskrifter på filtreringerne:



# Hvad der skal gøres

1. Scriptet skal kunne vise et køn ad gangen.
2. Der skal være filter-knapper til at vælge med.
3. Knapperne skal virke
4. Der skal være en overskrift, som viser det valgte
5. **Den aktuelle knap skal være fremhævet.**



# Den aktuelle knap skal være fremhævet

Ved at ændre udseende på den knap, der lige er klikket på, kan man gøre det nemmere for brugeren at se, hvilken filtrering, der er foretaget.

Når der klikkes på en knap, kan **evenhandler-funktionen**:

- **slette** *.valgt* klassen fra alle knapper `classList.remove()`
- **tilføj** *.valgt* klassen på den aktuelle knap `classList.add()`

# Et klassisk mønster

```
function filterBTNs() {  
    filter = this.dataset.køn;  
    document.querySelector("h1").textContent = this.textContent;  
    document.querySelectorAll(".filter").forEach((btn)=>{  
        btn.classList.remove("valgt");  
    })  
    this.classList.add("valgt");  
    visPersoner();  
}
```

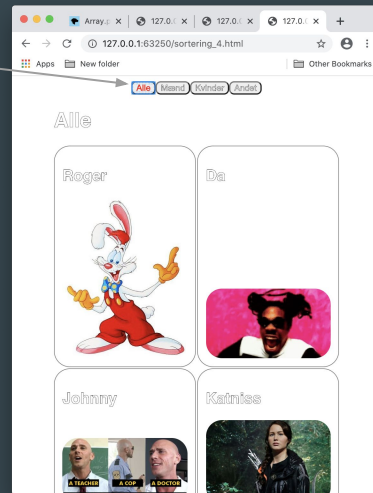
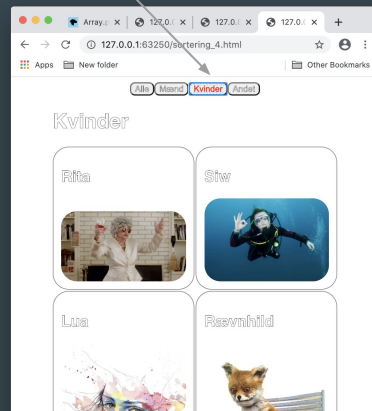
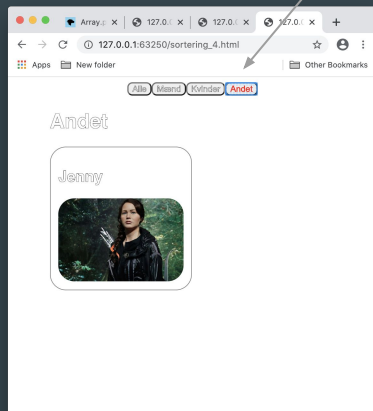
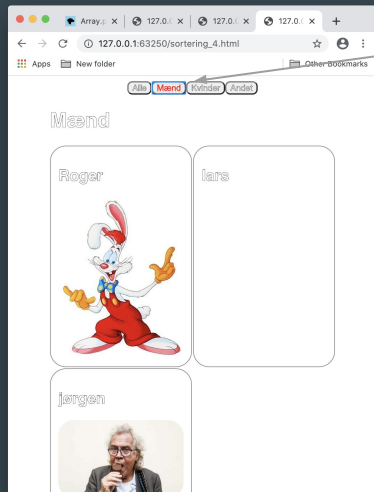
Fjern classen "valgt" fra alle knapper ...

... og tilføj classen "valgt" til den knap der er klikket på

# Øvelse 5: Fremhæv aktuel knap

Gem **04-overskrift.html** i en ny kopi, **05-fremhæv.html**

Nu skal den valgte kategori også tydeliggøres med styling af den aktuelle knap





# Ekstraøvelse

# Øvelse 5: Filtersortering

Gem **04-virker.html** i en ny kopi, **05-filtersortering.html**

Istedet for at knapperne er radiobuttons, skal de nu være sådan indrettede at vær knap slår et filter fra og til. Der er således ikke brug for *Alle*-knappen da det svarer til at have både *Mænd*-, *Kvinder*- og *Andetknapperne* trykket ind samtidigt. Der vil være brug for at oprette tre filter, en pr køn og lave en if sætning i *visPersoner* der med `||` finder ud af hvad der skal vises ....