

Kilder & Fremgangsmåte:

Kilder brukt:

ChatGPT, CoPilot, https://github.com/toremake/uin2025_coursebase

Jeg kopierte litt av arrayen fra resurser.js og starten på min kode i arbeidskravScript.js og la det til ChatGPT:

```
const resources = [
  {
    category: "HTML",
    text: "HTML står for HyperText Markup Language, og er et strukturspråk som brukes for å lage strukturer til nettside- og applikasjonsgrensesnitt.",
    sources: [
      {
        title: "W3Schools",
        url: "https://www.w3schools.com/html/"
      },
      {
        title: "HTML Living standard",
        url: "https://html.spec.whatwg.org/multipage/"
      },
      {
        title: "HTML.com Tutorials",
        url: "https://html.com/"
      }
    ]
  }
],
```



It looks like you've shared a snippet of JavaScript code with an array of resources related to HTML. Would you like assistance with extending or modifying this code, or do you have a specific question about it?

```
console.log(resources);

let resourcesHTML = "";

resources.map((resources, index) => resourcesHTML += <article id="infoSide">
  <h2>${resources.category}</h2>
  <p>${resources.text}</p>
  <nav>
    <ul>

      <li><a href="${resources.title}">${resources.sources.title}
    </a></li>
      <li><a href="#">w3Schools</a></li>
      <li><a href="#">w3Schools</a></li>
    </ul>
  </nav>
</article>);
```

```
document.getElementById("infoSide").innerHTML = resourcesHTML;
```

Det er en array inni en array på forrige spørsmål?



Yes, you're correct! The data in your `resources` array contains objects that each have a `sources` array, which itself contains objects (with `title` and `url`). When you're trying to map through `resources` and generate the HTML, there are a couple of issues that need to be addressed.

1. **Accessing `resources.title`**: The line where you're trying to access `resources.title` will not work as intended because `title` is part of each object inside the `sources` array. You need to iterate over `sources` within each `resources` item to generate the list of links correctly.
2. **HTML Output**: You are using `resources.title`, but you should be iterating over the `sources` array to get the individual titles and URLs for the anchor tags.

Here's an updated version of your code with the correct structure:

```
javascript Kopier Rediger

console.log(resources);

let resourcesHTML = "";

resources.map((resource, index) => {
  resourcesHTML += `
    <article id="infoSide">
      <h2>${resource.category}</h2>
      <p>${resource.text}</p>
      <nav>
        <ul>
          ${resource.sources.map((source) => `
            <li><a href="${source.url}" target="_blank">${source.title}</a></li>
          `).join('')}
        </ul>
      </nav>
    </article>
  `;
});

document.getElementById("infoSide").innerHTML = resourcesHTML;
```

Changes made:

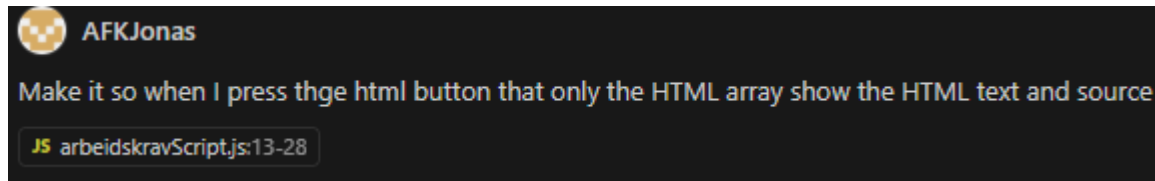
1. **Iterate over `resource.sources`**: For each resource, we map over the `sources` array to dynamically create the `` elements with the appropriate links.
↓
2. **`join('')`**: This is used to combine all the list items (``) into a single string, which is then inserted into the HTML.

Explanation:

- We loop over the `resources` array using `map`.
- Inside each `resource`, we loop over the `sources` array to create the list of links dynamically.
- The `join('')` ensures that the list of links is properly concatenated into a string without any unnecessary commas between the list items.

Let me know if you need any further clarification or adjustments!

Etterpå spurte jeg Copilot med den kopierte koden fra ChatGPT:



Forslaget var da:

Workspace

Used 4 references

Collecting workspace information

To achieve this, you need to modify the `showinfokort` function to filter the resources based on the category and update the HTML content accordingly. Here is the updated code:

```
//loggfører til konsollen for å sjekke at arrayet er korrekt:
console.log(resources);

//lager en variabel som skal holde på HTML-koden som skal vises på nettsiden:
let resourcesHTML = "";

//lager en .map() for å hente ut data fra arrayet og legge det inn i HTML-koden:
resources.map((resource, index) => resourcesHTML += `<article>
  <h2>${resource.category}</h2>
  <p>${resource.text}</p>
  <nav>
    <ul>
      ${resource.sources.map((source, index) => `<li><a href="${source.url}">${source.title}</a></li>`)}
    </ul>
  </nav>
</article>`);
//skriver ut HTML-koden til nettsiden:
document.getElementById("infoSide").innerHTML = resourcesHTML;


function showinfokort(category) {
  const infoSide = document.getElementById("infoSide");
  const filteredResources = resources.filter(resource => resource.category === category);
  let filteredHTML = "";
  filteredResources.map((resource, index) => filteredHTML += `<article>
    <h2>${resource.category}</h2>
    <p>${resource.text}</p>
    <nav>
      <ul>
        ${resource.sources.map((source, index) => `<li><a href="${source.url}">${source.title}</a></li>`)}
      </ul>
    </nav>
  </article>`);
  infoSide.innerHTML = filteredHTML;
  infoSide.style.display = "block";
}

document.addEventListener("click", showinfokort);
```

Additionally, update the `onclick` attributes in the HTML file to pass the category to the `showinfokort` function:

```
<button onclick="showinfokort('HTML')">HTML</button>
<button onclick="showinfokort('CSS')">CSS</button>
<button onclick="showinfokort('JavaScript')">JavaScript</button>
<button onclick="showinfokort('React')">React</button>
<button onclick="showinfokort('Sanity and headless CMS')">Sanity and headless CMS</button>
<button onclick="showinfokort('Sanity and headless CMS')">Sanity and headless CMS</button>
```

With these changes, clicking on a button will filter the resources based on the category and display the corresponding text and sources.

 AFKJonas

Like this?: //loggfører til konsollen for å sjekke at arrayet er korrekt: console.log(resources);

//lager en variabel som skal holde på HTML-koden som skal vises på nettsiden: let resourcesHTML = "";


//lager en .map() for å hente ut data fra arrayet og legge det inn i HTML-koden: resources.map((resources, index) => resourcesHTML +=
<article> <h2>\${resources.category}</h2> <p>\${resources.text}</p> <nav> \${resources.sources.map((source, index) =>
 \${source.title}).join('')} </nav> </article>)); //skriver ut HTML-koden til nettsiden:
document.getElementById("infoSide").innerHTML = resourcesHTML;

function showinfokort(category) { const infoSide = document.getElementById("infoSide"); const filteredResources = resources.filter(resource
=> resource.category === category); let filteredHTML = ""; filteredResources.map((resource, index) => filteredHTML += <article>
<h2>\${resource.category}</h2> <p>\${resource.text}</p> <nav> \${resource.sources.map((source, index) => \${source.title}).join('')} </nav> </article>)); infoSide.innerHTML = filteredHTML;
infoSide.style.display = "block"; }

document.addEventListener("click", showinfokort);

JS arbeidskravScript.js:1-37

Da ga den meg følgende svar:

 GitHub Copilot

> Used 1 reference

Yes, your code looks correct. However, you need to ensure that the `showinfokort` function is called with the correct category when a button is clicked. Here is the complete code with the necessary changes:

//loggfører til konsollen for å sjekke at arrayet er korrekt:
console.log(resources);

//lager en variabel som skal holde på HTML-koden som skal vises på nettsiden:
let resourcesHTML = "";

//lager en .map() for å hente ut data fra arrayet og legge det inn i HTML-koden:
resources.map((resource, index) => resourcesHTML += `<article>
 <h2>\${resource.category}</h2>
 <p>\${resource.text}</p>
 <nav>

 \${resource.sources.map((source, index) => `\${source.title}`)}

 </nav>
</article>`);

//skriver ut HTML-koden til nettsiden:
document.getElementById("infoSide").innerHTML = resourcesHTML;

function showinfokort(category) {
 const infoSide = document.getElementById("infoSide");
 const filteredResources = resources.filter(resource => resource.category === category);
 let filteredHTML = "";
 filteredResources.map((resource, index) => filteredHTML += `<article>
 <h2>\${resource.category}</h2>
 <p>\${resource.text}</p>
 <nav>

 \${resource.sources.map((source, index) => `\${source.title}`)}




 </nav>
 </article>`);
 infoSide.innerHTML = filteredHTML;
 infoSide.style.display = "block";
}

// Example buttons in your HTML file
document.getElementById("htmlButton").addEventListener("click", () => showinfokort('HTML'));
document.getElementById("cssButton").addEventListener("click", () => showinfokort('CSS'));
document.getElementById("jsButton").addEventListener("click", () => showinfokort('JavaScript'));
document.getElementById("reactButton").addEventListener("click", () => showinfokort('React'));
document.getElementById("sanityButton").addEventListener("click", () => showinfokort('Sanity and headless CMS'));

Make sure your HTML file has buttons with the corresponding IDs:

<button id="htmlButton">HTML</button>
<button id="cssButton">CSS</button>
<button id="jsButton">JavaScript</button>
<button id="reactButton">React</button>
<button id="sanityButton">Sanity and headless CMS</button>

With these changes, clicking on a button will filter the resources based on the category and display the corresponding text and sources.

Etter det, justerte jeg litt på koden selv, til den så slik ut for meg nå:

```
1 //logger til konsollen for å sjekke at arrayet er korrekt:
2 console.log(resources);
3
4 //laget en variabel som skal holde på HTML-koden som skal vises på nettsiden:
5 let resourcesHTML = "";
6
7 //lager en .map() for å hente ut data fra arrayet og legge det inn i HTML-koden:
8 resources.map((resource, index) => resourcesHTML += `<article>
9   <h2>${resource.category}</h2>
10   <p>${resource.text}</p>
11   <nav>
12     <ul>
13       ${resource.sources.map((source, index) => `<li><a href="${source.url}">${source.title}</a></li>`).join('')}
14     </ul>
15   </nav>
16 </article>`);
17
18 function showInfoKort(category) {
19   const infoSide = document.getElementById("infoSide");
20   const filteredResources = resources.filter(resource => resource.category === category);
21   const filteredHTML = filteredResources.map(resource => `
22     <article>
23       <h2>${resource.category}</h2>
24       <p>${resource.text}</p>
25       <nav>
26         <ul>
27           ${resource.sources.map(source => `<li><a href="${source.url}">${source.title}</a></li>`).join('')}
28         </ul>
29       </nav>
30     </article>
31 `).join('');
32
33   infoSide.innerHTML = filteredHTML;
34 }
35
36 //Legger bakgrunnsfarge på knappene for å vise hvilken kategori som er valgt:
37
38 //addEventListener med "click" for å kalle funksjonen showInfoKort() og dens innhold:
39 document.getElementById("htmlKnapp").addEventListener("click", () => showInfoKort('HTML'));
40 document.getElementById("cssKnapp").addEventListener("click", () => showInfoKort('CSS'));
41 document.getElementById("jsKnapp").addEventListener("click", () => showInfoKort('JavaScript'));
42 document.getElementById("reactKnapp").addEventListener("click", () => showInfoKort('React'));
43 document.getElementById("sanityKnapp").addEventListener("click", () => showInfoKort('Sanity and headless CMS'));
44
```

Det var også da jeg la til id'ene på HTML - siden(arbeidskrav_1.html)

```
15 <button id="htmlKnapp" on
16 <button id="cssKnapp" on
17 <button id="jsKnapp" onc
18 <button id="reactKnapp"
19 <button id="sanityKnapp"
```

Til nå, vil jeg gjerne si at fra linje 13 til 43 er kanskje ikke alt helt direkte kopiert fra CoPilot, men det er fortsatt store mengder.

Etter det, for at siden skal starte med 'HTML'-kortet hver gang siden lastes, la jeg til funksjonen "showInfoKort()" slik at funksjonen skulle starte.

```
44 //kaller funksjonen for å vise HTML-informasjonen når siden lastes:
45 showInfoKort('HTML');
```

Nå fungerer det slik som det skal, bare mangler at fargene på knappene skal endres etter hvilke knapp som er trykket på.

Oppsettet og inspirasjonen er basert på:

https://github.com/toremake/uin2025_coursebase

Har en Javascript- fil til array'en, og en annen JS-fil til script'et.

Jeg spurte igjen ChatGPT(Kopierte både hele HTML koden og JS koden):

```
gjør slik at når jeg klikker på en knapp så endrer kun den
bakgrunnsfargen til hvit også når jeg trykker på en annen knapp så
blir den om til hvit og den forrige går tilbake til sin forrige farge, her er
HTML koden og Javascript koden min: <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Arbeidskrav 1</title>
  <link rel="stylesheet" href="style.css">
  <script src="arbeidskravScript.js" defer></script>
  <script src="ressurser.js"></script>
</head>
<body>

<header>
  <nav>
    <button id="htmlKnapp">
onclick="showInfoKort('HTML')">HTML</button>
    <button id="cssKnapp">
onclick="showInfoKort('CSS')">CSS</button>
    <button id="jsKnapp">
onclick="showInfoKort('JavaScript')">JavaScript</button>
    <button id="reactKnapp">
onclick="showInfoKort('React')">React</button>
    <button id="sanityKnapp" onclick="showInfoKort('Sanity and
headless CMS')">Sanity and headless CMS</button>
  </nav>
</header>

<main id="infoSide">
  <!-- <article>
    <h2>HTML</h2>
    <p>HTML står for HyperText Markup Language, og er et
strukturspråk som brukes for å lage strukturer til nettside- og
applikasjongsgransnitt.</p>
    <nav>
      <ul>
        <li><a href="#">w3Schools</a></li>
        <li><a href="#">w3Schools</a></li>
        <li><a href="#">w3Schools</a></li>
      </ul>
    </nav>
  </article> -->
</main>

<footer></footer>

</body>
</html> og //Er øverst i scriptet for at infokortet med 'HTML' skal
lastes/kjøre først, (kaller funksjonen for å vise HTML-informasjonen
når siden lastes);
showInfoKort('HTML');
//loggfører til kons: ↵ for å sjekke at arrayet er korrekt:
console.log(resources);
```

```
//ressurser.js
console.log(resources);

//laget en variabel som skal holde på HTML-koden som skal vises på
nettsiden:
let resourcesHTML = "";

//lager en .map() for å hente ut data fra arrayet(resources) og legge
det inn i HTML-koden:
resources.map((resource, index) => resourcesHTML += <article>
  <h2> ${resource.category}</h2>
  <p> ${resource.text}</p>
  <nav>
    <ul>
      <!--lager en .map() for å hente ut data fra arrayet "sources" på
innsiden av arrayet "resources", og legger dataen(title og url) fra
sources inn i HTML-koden-->
        ${resource.sources.map((source, index) => <li><a
href="${source.url}">${source.title}</a></li>).join("")}
      </ul>
    </nav>
  </article>);

//funksjon her for filtrere den infoen til den kategorien som er valgt
og viser den.
function showInfoKort(category) {
  //const her for å hente ut elementet med id "infoSide" fra HTML-
filen:
  const infoSide = document.getElementById("infoSide");
  //bruker .filter() for å filtrere ut data fra arrayet(resources) basert på
hvilken kategori som er valgt:
  const filteredResources = resources.filter(resource =>
resource.category === category);
  //bruker .map() her for å hente ut dataen fra filteredResources og
legge den inn i HTML-koden:
  const filteredHTML = filteredResources.map(resource =>
  <article>
    <h2> ${resource.category}</h2>
    <p> ${resource.text}</p>
    <nav>
      <ul>
        ${resource.sources.map(source => <li><a
href="${source.url}">${source.title}</a></li>).join("")}
      </ul>
    </nav>
  </article>
  ).join('');

  //Plasserer infoen inn i der id="infoSide" er lokalisert i HTML-
filen:
  infoSide.innerHTML = filteredHTML;
}

//Legger bakgrunnsfarge på knappene for å vise hvilken kategori
som er valgt:
function changeBackgroundButtons(){
  document.getElementById("htmlKnapp").style.backgroundColor
= "#fff";
  ↵
  document.getElementById("cssKnapp").style.backgroundColor =
```

```
document.getElementById('htmlKnapp').style.backgroundColor = "#fff";
document.getElementById('cssKnapp').style.backgroundColor = "#fff";
document.getElementById('jsKnapp').style.backgroundColor = "#fff";
document.getElementById('reactKnapp').style.backgroundColor = "#fff";
document.getElementById('sanityKnapp').style.backgroundColor = "#fff";
}

//addEventListener med "click" for å kalle på funksjonen showInfoKort() og dens innhold:
document.getElementById('htmlKnapp').addEventListener("click", () => showInfoKort('HTML'));
document.getElementById('cssKnapp').addEventListener("click", () => showInfoKort('CSS'));
document.getElementById('jsKnapp').addEventListener("click", () => showInfoKort('JavaScript'));
document.getElementById('reactKnapp').addEventListener("click", () => showInfoKort('React'));
document.getElementById('sanityKnapp').addEventListener("click", () => showInfoKort('Sanity and headless CMS'));
```



For å oppnå det du ønsker (at når en knapp trykkes, endres dens bakgrunnsfarge til hvit, mens de andre knappene går tilbake til sin originale farge), må vi gjøre noen små justeringer i både HTML og JavaScript.

1. HTML Justeringer:

Du trenger å legge til en felles klasse (som f.eks. `knapp`) på alle knappene for enklere håndtering i JavaScript.

Oppdatert HTML:

```
html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Arbeidskrav 1</title>
  <link rel="stylesheet" href="style.css">
  <script src="arbeidskravScript.js" defer</script>
  <script src="ressurser.js"></script>
</head>
<body>

  <header>
    <nav>

      <button class="knapp" id="htmlKnapp" onclick="showInfoKort('HTML')">HTML</button>
      <button class="knapp" id="cssKnapp" onclick="showInfoKort('CSS')">CSS</button>
      <button class="knapp" id="jsKnapp" onclick="showInfoKort('JavaScript')">JavaScript</button>
      <button class="knapp" id="reactKnapp" onclick="showInfoKort('React')">React</button>
```

```
<button class="knapp" id="jsKnapp" onclick="showInfoKort('JavaScript')">JavaScript</button>
<button class="knapp" id="reactKnapp" onclick="showInfoKort('React')">React</button>
<button class="knapp" id="sanityKnapp" onclick="showInfoKort('Sanity and headless CMS')">Sanity and headless CMS</button>
</nav>
</header>

<main id="infoSide">
  <!-- <article>

    <h2>HTML</h2>
    <p>HTML står for HyperText Markup Language, og er et strukturspråk som brukes for å skrive nettsider.
    <nav>
      <ul>
        <li><a href="#">W3Schools</a></li>
        <li><a href="#">W3Schools</a></li>
        <li><a href="#">W3Schools</a></li>
      </ul>
    </nav>
  </article> -->
</main>

<footer></footer>

</body>
</html>
```

2. JavaScript Justeringer:

Du trenger å fjerne bakgrunnsfargen til knappene hver gang en knapp trykkes, og så sette den aktive knappen til hvit. Dette kan du gjøre i funksjonen som håndterer knappetrykkene.

Oppdatert JavaScript:

```
javascript
// Er øverst i scriptet for at infokortet med 'HTML' skal lastes/kjøres først, (kaller funksjonen showInfoKort('HTML'));

// Funksjon for å filtrere den infoen til den kategorien som er valgt og viser den:
function showInfoKort(category) {
  const infoSide = document.getElementById("infoSide");
  const filteredResources = resources.filter(resource => resource.category === category);
  const filteredHTML = filteredResources.map(resource => `
    <article>
      <h2>${resource.category}</h2>
      <p>${resource.text}</p>
      <nav>
        <ul>
          <li>${resource.sources.map(source => `<li><a href="${source.url}">${source.url}</a></li>`)}</li>
        </ul>
      </nav>
    </article>
  `).join('');
  infoSide.innerHTML = filteredHTML;
}

// Legger bakgrunnsfarge på knappene for å vise hvilken kategori som er valgt
function changeBackgroundButtons(activeId) {
  // Først setter vi alle knappene tilbake til den opprinnelige bakgrunnsfargen
  // Følgende farge er den opprinnelige bakgrunnsfargen for knappene i HTML, CSS, JavaScript, React og Sanity and headless CMS
```

```
// Først setter vi alle knappene tilbake til den opprinnelige bakgrunnsfarge
document.querySelectorAll(".knapp").forEach(knapp => {
  knapp.style.backgroundColor = ""; // Tilbakestill til ingen bakgrunnsfarge
});

// Så endrer vi bakgrunnsfargen til hvit på den aktive knappen
document.getElementById(activeButtonId).style.backgroundColor = "#fff";
}

// Legger til event listeners for hver knapp
document.getElementById("htmlKnapp").addEventListener("click", () => {
  showInfoKort('HTML');
  changeBackgroundButtons('htmlKnapp');
});
document.getElementById("cssKnapp").addEventListener("click", () => {
  showInfoKort('CSS');
  changeBackgroundButtons('cssKnapp');
});
document.getElementById("jsKnapp").addEventListener("click", () => {
  showInfoKort('JavaScript');
  changeBackgroundButtons('jsKnapp');
});
document.getElementById("reactKnapp").addEventListener("click", () => {
  showInfoKort('React');
  changeBackgroundButtons('reactKnapp');
});
document.getElementById("sanityKnapp").addEventListener("click", () => {
  showInfoKort('Sanity and headless CMS');
  changeBackgroundButtons('sanityKnapp');
});
```

Hva skjer nå?

- 1. Når en knapp trykkes, vil bakgrunnsfargen på den aktive knappen bli satt til hvit (#fff).
- 2. Bakgrunnsfargen på de andre knappene vil bli fjernet (tilbakestilles til default).
- 3. Når du trykker på en knapp, vises informasjonen for den valgte kategorien.

CSS (valgfritt):

For å sikre at knappene ser bra ut, kan du legge til grunnleggende styling i style.css :

```
css

/* Generell styling for knappene */
.knapp {
  background-color: #f0f0f0;
  border: 1px solid #ccc;
  padding: 10px 20px;
  cursor: pointer;
  margin: 5px;
}

/* Når en knapp er aktiv (får hvit bakgrunn) */
.knapp:focus {
  outline: none;
}
```

```
outline: none;
}
```

Med dette oppsettet vil knappene endre bakgrunnsfarge til hvit når de klikkes, og den aktive knappen får den ønskede effekten.

Koden min før spørsmålet:
HTML:

```
Arbeidskrav_1 > arbeidskrav_1.html > html > body > main#infoSide
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Arbeidskrav 1</title>
7   <link rel="stylesheet" href="style.css">
8   <script src="arbeidskravScript.js" defer></script>
9   <script src="ressurser.js"></script>
10 </head>
11 <body>
12
13 <header>
14   <nav>
15     <button id="htmlKnapp" onclick="showInfoKort('HTML')">HTML</button>
16     <button id="cssKnapp" onclick="showInfoKort('CSS')">CSS</button>
17     <button id="jsKnapp" onclick="showInfoKort('JavaScript')">JavaScript</button>
18     <button id="reactKnapp" onclick="showInfoKort('React')">React</button>
19     <button id="sanityKnapp" onclick="showInfoKort('Sanity and headless CMS')">Sanity and headless CMS</button>
20   </nav>
21 </header>
22
23 <main id="infoSide">
24   <!-- <article>
25     <h2>HTML</h2>
26     <p>HTML står for HyperText Markup Language, og er et strukturspråk som brukes for å lage strukturer til nettside- og applikasjonsgrensesnitt.</p>
27     <nav>
28       <ul>
29         <li><a href="#">w3Schools</a></li>
30         <li><a href="#">w3Schools</a></li>
31         <li><a href="#">w3Schools</a></li>
32       </ul>
33     </nav>
34   </article> -->
35 </main>
36
37 <footer></footer>
38
39 </body>
40 </html>
```

JS:

```
Arbeidskrav_1 > JS arbeidskravScript.js --
1 //Er evrest i scriptet for at infokortet med 'HTML' skal lastes/kjøre først, (kaller funksjonen for å vise HTML-informasjonen når siden lastes):
2 showInfoKort('HTML');
3 //loggføren til konsollen for å sjekke at arrayet er korrekt:
4 console.log(resources);
5
6 //laget en variabel som skal holde på HTML-koden som skal vises på nettsiden:
7 let resourcesHTML = "";
8
9 //lager en .map() for å hente ut data fra arrayet(resources) og legge det inn i HTML-koden:
10 resources.map((resource, index) => resourcesHTML += `<article>
11   <h2>${(resource.category)}</h2>
12   <p>${(resource.text)}</p>
13   <nav>
14     <ul>
15       <!--lager en .map() for å hente ut data fra arrayet "sources" på innsiden av arrayet "resources", og legger dataen(title og url) fra sources inn i HTML-koden:-->
16       ${resource.sources.map((source, index) => `<li><a href="${(source.url)}">${(source.title)}</a></li>`).join('')}
17     </ul>
18   </nav>
19 </article>`);
20
21 //funksjon her for filtrere den infoen til den kategorien som er valgt og viser den.
22 function showInfoKort(category) {
23   //const her for å hente ut elementet med id "infoSide" fra HTML-filen:
24   const infoSide = document.getElementById("infoSide");
25   //bruker .filter() for å filtrere ut data fra arrayet(resources) basert på hvilken kategori som er valgt:
26   const filteredResources = resources.filter(resource => resource.category === category);
27   //bruker .map() her for å hente ut dataen fra filteredResources og legge den inn i HTML-koden:
28   const filteredHTML = filteredResources.map(resource => `
29     <article>
30       <h2>${(resource.category)}</h2>
31       <p>${(resource.text)}</p>
32       <nav>
33         <ul>
34           ${resource.sources.map(source => `<li><a href="${(source.url)}">${(source.title)}</a></li>`).join('')}
35         </ul>
36       </nav>
37     </article>
38 `).join('');
39
40 //Plasserer infoen inn i den id="infoSide"/ er lokalisert i HTML-filen:
41 infoSide.innerHTML = filteredHTML;
42 }
43
44 //Legger bakgrunnsfarge på knappene for å vise hvilken kategori som er valgt:
45 function changeBackgroundButtons(){
46   document.getElementById("htmlKnapp").style.backgroundColor = "#ffff";
47   document.getElementById("cssKnapp").style.backgroundColor = "#ffff";
48   document.getElementById("jsKnapp").style.backgroundColor = "#ffff";
49   document.getElementById("reactKnapp").style.backgroundColor = "#ffff";
50   document.getElementById("sanityKnapp").style.backgroundColor = "#ffff";
51 }
52
53
54 //addEventlistener med "click" for å kalle på funksjonen showInfoKort() og dens innhold:
55 document.getElementById("htmlKnapp").addEventListener("click", () => showInfoKort('HTML'));
56 document.getElementById("cssKnapp").addEventListener("click", () => showInfoKort('CSS'));
57 document.getElementById("jsKnapp").addEventListener("click", () => showInfoKort('JavaScript'));
58 document.getElementById("reactKnapp").addEventListener("click", () => showInfoKort('React'));
59 document.getElementById("sanityKnapp").addEventListener("click", () => showInfoKort('Sanity and headless CMS'));
60
61
```

Koden min etter spørsmålet:

HTML:

Endringer:

linje:

15 til 19,

lagt til(kopiert):

class="knapp"

```
Arbeidskrav_1 > arbeidskrav_1.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Arbeidskrav 1</title>
7   <link rel="stylesheet" href="style.css">
8   <script src="arbeidskravScript.js" defer></script>
9   <script src="ressurser.js"></script>
10 </head>
11 <body>
12
13 <header>
14   <nav>
15     <button class="knapp" id="htmlKnapp" onclick="showInfoKort('HTML')">HTML</button>
16     <button class="knapp" id="cssKnapp" onclick="showInfoKort('CSS')">CSS</button>
17     <button class="knapp" id="jsKnapp" onclick="showInfoKort('JavaScript')">JavaScript</button>
18     <button class="knapp" id="reactKnapp" onclick="showInfoKort('React')">React</button>
19     <button class="knapp" id="sanityKnapp" onclick="showInfoKort('Sanity and headless CMS')">Sanity and headless CMS</button>
20   </nav>
21 </header>
22
23 <main id="infoSide">
24   <!-- <article>
25     <h2>HTML</h2>
26     <p>HTML står for HyperText Markup Language, og er et strukturspråk som brukes for å lage strukturer til nettside- og applikasjonsgrensesnitt.</p>
27     <nav>
28       <ul>
29         <li><a href="#">w3Schools</a></li>
30         <li><a href="#">w3Schools</a></li>
31         <li><a href="#">w3Schools</a></li>
32       </ul>
33     </nav>
34   </article> -->
35 </main>
36
37 <footer></footer>
38
39 </body>
40 </html>
```

JS:

Endringer:

linje:

54 til 75,

(kopiert)

```
Arbeidskrav_1 > # arbeidskravScript.js > addEventListener('click') callback
1 //Er øverst i scriptet for at infokortet med 'HTML' skal lastes/kjøres først, (kaller funksjonen for å vise HTML-informasjonen når siden lastes):
2 showInfoKort('HTML');
3 //loggerferer til konsollen for å sjekke at arrayet er korrekt:
4 console.log(resources);
5
6 //laget en variabel som skal holde på HTML-koden som skal vises på nettsiden:
7 let resourcesHTML = "";
8
9 //lager en .map() for å hente ut data fra arrayet(resources) og legge det inn i HTML-koden:
10 resources.map((resource, index) => resourcesHTML += `<article>
11   <h2>${resource.category}</h2>
12   <p>${resource.text}</p>
13   <nav>
14     <ul>
15       <!--lager en .map() for å hente ut data fra arrayet "sources" på innsiden av arrayet "resources", og legger dataen(title og url) fra sources inn i HTML-koden:-->
16       ${resource.sources.map((source, index) => `<li><a href="${source.url}">${source.title}</a></li>`).join("")}
17     </ul>
18   </nav>
19 </article>`);
20
21 //Funksjon her for filtrere den infoen til den kategorien som er valgt og viser den.
22 function showInfoKort(category) {
23   //const her for å hente ut elementet med id "infoSide" fra HTML-filen:
24   const infoSide = document.getElementById("infoSide");
25   //bruker .filter() for å filtrere ut data fra arrayet(resources) basert på hvilken kategori som er valgt:
26   const filteredResources = resources.filter(resource => resource.category === category);
27   //bruker .map() her for å hente ut dataen fra filteredResources og legge den inn i HTML-koden:
28   const filteredHTML = filteredResources.map(resource => `
29     <article>
30       <h2>${resource.category}</h2>
31       <p>${resource.text}</p>
32       <nav>
33         <ul>
34           ${resource.sources.map(source => `<li><a href="${source.url}">${source.title}</a></li>`).join("")}
35         </ul>
36       </nav>
37     </article>
38   `).join("");
39
40   //Plasserer infoen inn i der id="infoSide" er lokalisert i HTML-filen:
41   infoSide.innerHTML = filteredHTML;
42 }
43
44 // Legger bakgrunnsfarge på knappene for å vise hvilken kategori som er valgt
45 function changeBackgroundButtons(activeButtonId) {
46   // Først setter vi alle knappene tilbake til den opprinnelige bakgrunnsfargen
47   document.querySelectorAll(".knapp").forEach(knapp => {
48     knapp.style.backgroundColor = ""; // Tilbakestiller til ingen bakgrunnsfarge
49   });
50
51   // Så endrer vi bakgrunnsfargen til hvit på den aktive knappen
52   document.getElementById(activeButtonId).style.backgroundColor = "#fff";
53 }
54
55 // Legger til event listeners for hver knapp
56 document.getElementById("htmlKnapp").addEventListener("click", () => {
57   showInfoKort('HTML');
58   changeBackgroundButtons('htmlKnapp');
59 });
60 document.getElementById("cssKnapp").addEventListener("click", () => {
61   showInfoKort('CSS');
62   changeBackgroundButtons('cssKnapp');
63 });
64 document.getElementById("jsKnapp").addEventListener("click", () => {
65   showInfoKort('JavaScript');
66   changeBackgroundButtons('jsKnapp');
67 });
68 document.getElementById("reactKnapp").addEventListener("click", () => {
69   showInfoKort('React');
70   changeBackgroundButtons('reactKnapp');
71 });
72 document.getElementById("sanityKnapp").addEventListener("click", () => {
73   showInfoKort('Sanity and headless CMS');
74   changeBackgroundButtons('sanityKnapp');
75 });
76
77
78
```

Fullstendig kode, med små endringer + lagt til kommentarer:

```
Arbeidskrav_1 > ➤ arbeidskravScript.js ➤ changeBackgroundButtons
1 | //Er øverst i scriptet for at infokortet med 'HTML' skal lastes/kjøre først, (kaller funksjonen for å vise HTML-informasjonen når siden lastes):
2 | showInfoKort('HTML');
3 | changeBackgroundButtons('htmlKnapp');
4 | //loggfører til konsollen for å sjekke at arrayet er korrekt:
5 | console.log(resources);
6 |
7 | //laget en variabel som skal holde på HTML-koden som skal vises på nettsiden:
8 | let resourcesHTML = "";
9 |
10 | //lager en .map() for å hente ut data fra arrayet(resources) og legge det inn i HTML-koden:
11 | resources.map((resource, index) => resourcesHTML += `<article>
12 |     <h2>${(resource.category)}</h2>
13 |     <p>${(resource.text)}</p>
14 |     <nav>
15 |         <ul>
16 |             <!--lager en .map() for å hente ut data fra arrayet "sources" på innsiden av arrayet "resources", og legger dataen(title og url) fra sources inn i HTML-koden:-->
17 |             ${resource.sources.map((source, index) => `<li><a href="${(source.url)}">${(source.title)}</a></li>`).join('')}
18 |         </ul>
19 |     </nav>
20 | </article>`);
21 |
22 | //Funksjon her for filtrere den infoen til den kategorien som er valgt og viser den:
23 | function showInfoKort(category) {
24 |     //const her for å hente ut elementet med id "infoSide" fra HTML-filen:
25 |     const infoSide = document.getElementById("infoSide");
26 |     //bruker .filter() for å filtrere ut data fra arrayet(resources) basert på hvilken kategori som er valgt:
27 |     const filteredResources = resources.filter(resource => resource.category === category);
28 |     //bruker .map() her for å hente ut dataen fra filteredResources og legge den inn i HTML-koden:
29 |     const filteredHTML = filteredResources.map(resource => `
30 |         <article>
31 |             <h2>${(resource.category)}</h2>
32 |             <p>${(resource.text)}</p>
33 |             <nav>
34 |                 <ul>
35 |                     ${resource.sources.map(source => `<li><a href="${(source.url)}">${(source.title)}</a></li>`).join('')}
36 |                 </ul>
37 |             </nav>
38 |         </article>
39 |     `).join('');
40 |
41 |     //Plasserer infoen inn i der id="infoSide"/ er lokalisert i HTML-filen:
42 |     infoSide.innerHTML = filteredHTML;
43 | }
44 |
45 | //Funksjon som først tilbakestiller de andre knappene til standard bakgrunnsfarge,
46 | //og deretter endrer både bakgrunnsfargen og tekst-fargen til den aktive knappen:
47 | function changeBackgroundButtons(aktivKnappId) {
48 |     //querySelectorAll prøver finner alle elementene med klassen "knapp", og forEach går gjennom hvert element:
49 |     document.querySelectorAll(".knapp").forEach(knapp => {
50 |         //Tilbakestiller til standard bakgrunnsfarge:
51 |         knapp.style.backgroundColor = "";
52 |         //Tilbakestiller til standard tekstfarge:
53 |         knapp.style.color = "";
54 |     });
55 |
56 |     //Når knappen er aktiv, endres bakgrunnsfargen til hvit og teksten til svart:
57 |     document.getElementById(aktivKnappId).style.backgroundColor = "ffff";
58 |     document.getElementById(aktivKnappId).style.color = "000";
59 | }
60 |
61 | //EventListeners for å lytte etter om knappene blir trykket på, og hvis de blir trykket på,
62 | //så vil både funksjonene showInfoKort og changeBackgroundButtons kjøre:
63 | document.getElementById("htmlKnapp").addEventListener("click", () => {
64 |     showInfoKort('HTML');
65 |     changeBackgroundButtons('htmlKnapp');
66 | });
67 | document.getElementById("cssKnapp").addEventListener("click", () => {
68 |     showInfoKort('CSS');
69 |     changeBackgroundButtons('cssKnapp');
70 | });
71 | document.getElementById("jsKnapp").addEventListener("click", () => {
72 |     showInfoKort('JavaScript');
73 |     changeBackgroundButtons('jsKnapp');
74 | });
75 | document.getElementById("reactKnapp").addEventListener("click", () => {
76 |     showInfoKort('React');
77 |     changeBackgroundButtons('reactKnapp');
78 | });
79 | document.getElementById("sanityKnapp").addEventListener("click", () => {
80 |     showInfoKort('Sanity and headless CMS');
81 |     changeBackgroundButtons('sanityKnapp');
82 | });
83 | }
```