

Redigere HTML med JavaScript

Nå som vi vet hvordan vi kan hente HTML-elementer, kan vi se på hvordan vi kan *endre* eksisterende elementer med JavaScript.

La oss si at vi ønsker å endre en overskrift. Da må vi først hente elementet:

```
let overskriftEl = document.querySelector("h1");
```



Nå som vi har elementet i en variabel, kan vi bruke egenskapen `innerHTML` for å få tilgang til innholdet i `<h1>`-elementet:

```
console.log(overskriftEl.innerHTML);  
// Skriver ut: Velkommen til siden min
```



Legg merke til at `innerHTML` er en egenskap som kan gis en verdi, og som da fungerer som en variabel, og ikke en metode/funksjon. Vi skriver derfor ikke parenteser etter `innerHTML` når vi bruker den.

Vi kan bruke `innerHTML` til å gjøre om innholdet i et element. I programmet nedenfor endrer vi teksten i overskriften. Prøv selv å endre den til noe annet.

HTML CSS JS

LIVE

```
1 let overskriftEl = document.querySelector("h1");
2
3 overskriftEl.innerHTML = "Her er en ny overskrift";
```

Resources 1x

Her har vi bare brukt egenskapen `innerHTML` til å endre teksten i overskriften, men vi kan også bruke `innerHTML` til å legge til nye HTML-elementer. Vi kan derfor gjøre om hele innholdet på siden vår om vi ønsker det:

```
let bodyEl = document.querySelector("body");
bodyEl.innerHTML = "<p> Dette avsnittet erstatter alt innhold </p>";
```

Snakk!

innerHTML

Hva blir resultatet av koden ovenfor? Diskuter det først og prøv etterpå.

Med denne koden overskriver vi alt innholdet i `<body>`-elementet. Hvis vi vil legge til mye innhold, kan vi skrive det på en ryddig måte ved å bruke operatoren `+=`. Denne operatoren legger til mer innhold i stedet for å overskrive det.

```
1 let bodyEl = document.querySelector("body");
2
3 let tabell = "";
4
5 tabell += "<table>";
6 tabell += "<thead>";
7 tabell += "<tr>";
8 tabell += "<th>Overskrift 1</th>";
9 tabell += "<th>Overskrift 2</th>";
10 tabell += "</tr>";
11 tabell += "</thead>";
12 tabell += "<tbody>";
13 tabell += "<tr>";
14 tabell += "<td>Innhold 1</td>";
15 tabell += "<td>Innhold 2</td>";
16 tabell += "</tr>";
17 tabell += "</tbody>";
18 tabell += "</table>";
19
20 bodyEl.innerHTML += tabell;
```

Her lager vi en tabell ved å legge den til nederst i `<body>`-elementet. Legg merke

til at vi først lager variabelen `tabell`, som vi fyller med tabellkoden vår. Hvis vi skriver denne koden direkte til `<body>`-elementet, vil nettleseren forsøke å avslutte tabellen så snart den møter koden `<table>`. Det fører til at resten av tabellkoden havner utenfor `<table>`-elementet. Vi unngår det problemet ved å først lagre koden i en variabel, for så å legge den til i `<body>`-elementet. Denne framgangsmåten vil du få bruk for i mange sammenhenger.

Legge til HTML-elementer med attributter

I eksemplet ovenfor la vi til nye HTML-elementer (`<table>`, `<tr>`, `<th>` og `<td>`) med `innerHTML`. Vi vil ofte at nye elementer også skal ha attributter, både for å kunne hente dem med `document.querySelector()`, og fordi mange HTML-elementer må ha attributter for å fungere som de skal.

Når vi bruker `innerHTML` for å legge til attributter, vil vi få en utfordring. Hva tror du vil skje om vi prøver oss på følgende kode? (Prøv det selv.)

```
let bodyEl = document.querySelector("body");  
bodyEl.innerHTML += "<p id='mittAvsnitt'>Et nytt avsnitt</p>";
```

Her får vi en feilmelding fordi nettleseren tror at teksten bare består av `"<p id="`. For å unngå denne feilen kan vi bruke apostrofer (`'`) i stedet for anførselstegn (`"`):

```
bodyEl = document.querySelector("body");  
bodyEl.innerHTML += "<p id='mittAvsnitt'>Et gult avsnitt</p>";
```

Nå får vi et nytt avsnitt med attributtet `id="mittAvsnitt"`. Dette elementet kan vi nå jobbe videre med ved å bruke

```
document.querySelector("#mittAvsnitt");
```

Oppgaver

9 Lag et HTML-dokument med en overskrift og to avsnitt.

- 0 Bruk JavaScript til å gjøre om teksten i overskriften og begge avsnittene. Her må du finne en løsning som lar deg hente ut to like HTML-elementer (de to `<p>`-elementene).
- 1 Legg til en tabell som beskrevet ovenfor, men skriv innholdet rett til `bodyEl` uten å opprette variabelen `tabell`. Hva blir resultatet? Ved å høyreklikke i nettleseren og velge «inspiser» kan du undersøke koden som vises. Hva har skjedd med tabellkoden?
- 2 Legg til en ordnet liste nederst i `<body>`-elementet ved bare å bruke JavaScript.
- 3 Legg til en CSS-klasse i dokumentet som gir en rød kantlinje med tykkelse 3 piksler.
- 4 Legg til et nytt listepunkt nederst i listen ved å bare bruke JavaScript. Listepunktet skal bruke CSS-klassen du laget i forrige oppgave.

I dybden

innerText

I tillegg til egenskapen `innerHTML` kan du også møte på `innerText` når du ser på eksempler på internett. Forskjellen på de to metodene er at `innerText` bare lar deg legge til tekst i et element, mens `innerHTML` lar deg legge til tekst og HTML. Det er derfor greit å bruke `innerHTML` uansett hva vi ønsker å legge til, for å spare oss for unødvendig forvirring.

Forskjellen kan ha betydning når vi bare ønsker å lese ut informasjon fra en nettside. Da vil `innerText` gi oss tekstinnholdet, og hopper over eventuelle HTML-tagger.

Redigere attributter

I tillegg til å redigere innholdet i HTML vil det være behov for å endre og legge til attributter i HTML-elementer. Husk at et HTML-attributt lar oss legge til ekstra informasjon i HTML-elementer, for eksempel en klasse (`class="mittAvsnitt"`), bildefilen i et ``-element (med `src`-attributtet), eller hvor en lenke peker (med `href`-attributtet).

I koden nedenfor har vi utvidet eksemplet vårt med en ekstra CSS-klasse.

```
.rosa {  
  padding: 20px;  
  background-color: #FC6371;  
}
```



Nå kan vi hente ut `<p>`-elementet med klassen `.blaa`, og bytte ut klassen med klassen `.rosa`:

```
let avsnittEl = document.querySelector(".blaa");  
avsnittEl.setAttribute("class", "rosa");
```



Her bruker vi metoden `setAttribute()` for å angi verdien til et attributt. Vi skriver da både attributtets navn og attributtets nye verdi. NB! Legg merke til at vi ikke bruker punktum foran klassenavnet her. Det er fordi vi endrer HTML-attributtet, og der angis klassenavnene uten punktum. Vi kan også bruke `setAttribute()` til å legge til attributter der vi ikke har angitt noen fra før:

```
let overskriftEl = document.querySelector("h1");  
overskriftEl.setAttribute("class", "rosa");
```



Her får overskriften (`<h1>`-elementet) også rosa bakgrunnsfarge. Hvis vi ønsker å ta bort et attributt, kan vi bruke metoden `removeAttribute()` :

```
overskriftEl.removeAttribute("class");
```



Her angir vi bare attributtets navn fordi det ikke har noe å si hvilken verdi attributtet har, når vi bare skal fjerne det.

I dybden

Endre attributter med egenskaper

Som et alternativ til `setAttribute()` i eksemplet ovenfor kan vi bruke `className`. De fleste attributtene vi omtaler i denne boka, har sine egne egenskaper som gjør det lettere å endre dem:

```
let avsnittEl = document.querySelector(".blaa");  
avsnittEl.className = "rosa";
```



Her bruker vi altså egenskapen `className`, og fordi det er en egenskap og ikke en metode, bruker vi likhetstegn for å endre den i stedet for parenteser.

Det finnes også tilsvarende egenskaper for andre attributter, for eksempel `href`, `id` og `src`. Det vil si at vi lett kan endre hvor en lenke peker, et elements id eller bildefilen som vises i et bilde.

Oppgaver

Her skal du fortsette med HTML-dokumentet du laget i oppgavene ovenfor.

Lag en ny CSS-klasse som gir en blå kantlinje med tykkelse 3 piksler.

Gi overskriften i dokumentet ditt den nye CSS-klassen. Bruk JavaScript og egenskapen `className` for å oppnå dette.

Bruk JavaScript for å ta bort CSS-klassen på listepunktet som har fått en rød kantlinje.

Legg til kodene `` og `` i dokumentet ditt. Legg til teksten NRK i det første `<a>`-elementet og TV2 i det andre `<a>`-elementet ved å bruke JavaScript.

Bruk metoden `setAttribute()` og legg til attributter slik at NRK-lenken fører til **`http://www.nrk.no`** og åpnes i et nytt vindu.

Bruk JavaScript og egenskapene `href` og `target` og legg til attributter slik at TV2-lenken fører til **`http://www.tv2.no`** og åpnes i et nytt vindu.

Redigere CSS-egenskaper direkte

Vi har tidligere sett at vi kan bruke CSS på tre ulike måter. Én av måtene involverte å skrive CSS i `style`-attributtet direkte i HTML-tagger:

```
<p style="background-color: red;"> Et avsnitt </p>
```



Denne framgangsmåten anbefales *ikke*, fordi vi bør skille innhold (HTML) og utseende (CSS), men vi kan likevel bruke dette attributtet via JavaScript. Det gir mening når vi ønsker å gjøre enkle endringer. Vi har allerede sett hvordan vi kan få tilgang til HTML-attributter med JavaScript:

```
let avsnittEl = document.querySelector("p");  
console.log(avsnittEl.style);
```



Den siste linjen ovenfor skriver ut alle CSS-egenskapene til avsnittet i konsollen. Nå som vi har `style`-attributtet, kan vi endre CSS-egenskaper:

```
avsnittEl.style.backgroundColor = "blue";
```



Denne koden endrer egenskapen `backgroundColor` som vi finner i `style` - attributtet. Legg merke til at vi skriver `backgroundColor` og ikke `background-color` . Alle CSS-egenskaper som bruker bindestrek, blir slått sammen på denne måten.

Med denne framgangsmåten kan vi gå inn og redigere enkeltegenskaper i tillegg til å skifte ut CSS-klasser (som vi allerede har sett på).

Snakk!

Hva blir egenskapen?

Hva tror du blir egenskapene til `border-width`, `text-decoration` og `font-family` ?

Prøv å endre noen av disse i koden ovenfor.