

Et enkelt eksempel med filmer

Foreløpig har vi sett på hvordan vi kan motta informasjon fra bruker ved å la brukeren klikke på elementer. Nå skal vi se på hvordan vi kan la brukeren skrive inn informasjon selv.

Det finnes mange ulike felter vi kan bruke for registrering av informasjon. Vi skal starte med å se på et enkelt eksempel med to ulike tekstfelter, et tallfelt, en nedtrekksmeny og en knapp.

Tekstfelt

Det vanligste alternativet for å la en bruker skrive inn tekst, er et enkelt tekstfelt.

Vi lager et tekstfelt med `<input>`-elementet og attributtet `type="text"`:

```
<input type="text" id="tittel">
```



Legg merke til at `<input>`-elementet består av bare én tagg (ingen avslutningstagg). I tillegg bruker vi attributtet `id="tittel"`, slik at vi lett kan finne dette elementet med JavaScript. Vi kan også legge til tekst i selve `<input>`-elementet, for å vise brukeren et eksempel på hva som kan skrives inn:

```
<input type="text" id="tittel" placeholder="Titanic">
```



En slik tekst kalles en «placeholder»-tekst. Denne teksten forsvinner når brukeren skriver noe i feltet. Vi kunne ha brukt «Tittel» som placeholder-tekst, men det kan ofte være lurt å bruke placeholder-teksten til å vise et eksempel på hva som skal skrives inn, som i dette tilfellet er en filmtittel, for eksempel «Titanic».

Et `<input>`-element gir bare en tekstboks; vi bør derfor også legge til en tekst som forklarer hva som skal skrives i elementet:

```
<label for="tittel">Tittel</label>  
<input type="text" id="tittel">
```



label

Ovenfor plasserte vi et `<label>`-element foran `<input>`-elementet. Dette elementet inneholder teksten «Tittel», og forteller brukeren hva som skal skrives i tekstfeltet. For at `<label>`-elementet skal knyttes til et annet element, må `for`-attributtet i `<label>`-elementet ha samme verdi som `id`-attributtet i det andre elementet.

Det er to viktige grunner til at vi bruker `<label>`-elementet:

Universell utforming: Skjermlesere og andre programmer som skal lese innholdet på en nettside, vet umiddelbart at teksten «Tittel» i eksemplet ovenfor er knyttet til tekstfeltet.

Brukervennlighet: Hvis vi klikker på teksten i `<label>`-elementet, hopper vi umiddelbart til tekstfeltet. Det betyr at det er lettere å treffe tekstfeltet, noe som er spesielt verdifullt på touch-skjermer. Dette er ekstra nyttig ved bruk av avkrysningsknapper og radioknapper, som ofte er små og vanskelige å treffe.

En annen fordel er at vi får litt flere muligheter med CSS ved å bruke `<label>`-elementet.

textarea

Et tekstfelt laget med `<input>` -elementet egner seg best til korte tekster. Hvis vi ønsker å la brukeren skrive inn mye tekst, kan vi i stedet bruke `<textarea>` -elementet:

```
<label for="beskrivelse">Beskrivelse</label>
<textarea id="beskrivelse" placeholder="Skriv en kort beskrivelse">
```

I motsetning til `<input>` -elementet bruker dette elementet en avslutningstagg.

To skrivemåter for label

I eksemplet ovenfor skrev vi:

```
<label for="tittel">Tittel</label>
<input type="text" id="tittel">
```

Vi kan også bruke `<label>` slik:

```
<label>Tittel
  <input type="text" id="tittel">
</label>
```

Her «pakker vi inn» merkelappen og `<input>`-elementet i et `<label>`-element. Ved å bruke denne skrivemåten trenger vi ikke å vite hva som er id-en til `<input>`-elementet, og vi trenger ikke bruke `for`-attributtet. Nettleseren vil automatisk forstå hvilket element `<label>`-elementet tilhører.

Denne skrivemåten er ofte nyttig når vi lager `<input>`-elementer med JavaScript. Da er det ikke alltid slik at vi ønsker å bruke unike id-er for alle elementer.

Tall

For å skrive inn tallverdier kan vi bruke et `<input>`-element med attributtet `type="number"`:

```
<label for="aarstall">Årstall</label>
```



```
<input type="number" id="aarstall" placeholder="1997">
```

Dette feltet ser ut som et tekstfelt, men nettleseren vil forvente en tallverdi. På smarttelefoner skiftes ofte tastaturet ut med tall, slik at det blir lettere for brukeren.

Vi må likevel være litt på vakt, for JavaScript tolker verdiene som tekst selv om de er skrevet i et tallfelt. Vi skal snart se på et eksempel som viser hvordan vi forholder oss til det.

Nedtrekksmenyer

Hvis vi ønsker å la en bruker velge ett alternativ fra flere forhåndsbestemte alternativer, kan vi bruke en nedtrekksmeny. En nedtrekksmeny lages med et `<select>`-element, og alternativene i menyen lages med `<option>`-elementer:

```
<label for="aldersgrense">Aldersgrense</label>
<select id="aldersgrense">
  <option value="alle">Tillatt for alle</option>
  <option value="syv">7 år</option>
  <option value="elleve">11 år</option>
  <option value="femten">15 år</option>
  <option value="atten">18 år</option>
</select>
```



Her lar vi brukeren velge en aldersgrense for filmen som skal registreres. Vi kan bare velge ett av alternativene.

Knapper

Det eneste vi mangler nå for å lage et skjema, er en knapp. En knapp vil ikke alltid være nødvendig fordi vi kan bruke JavaScript-hendelser til å registrere når brukeren skriver noe i et felt. Med en slik hendelse kan vi oppdatere siden uten at brukeren klikker på knappen. Det vil likevel ofte være viktig med en knapp, for eksempel hvis brukeren må ta en avgjørelse som får konsekvenser når knappen klikkes. Her er det også viktig å tenke på brukervennlighet – det må være tydelig for brukeren hvordan nettsiden oppfører seg.

For å lage en knapp bruker vi elementet `<button>`. Knappens tekst plasserer vi inni elementet:

```
<button>Registrer</button>
```



Legg merke til at `<button>`-elementet også har en avslutningstagg.

Vi bruker ikke `<label>` -elementet for å merke et `<button>` -element. Her er det teksten i elementet som fungerer som merkelapp.

Du kan også finne knapper laget med `<input>` -elementet. Det er ingen stor forskjell mellom de to variantene, men `<button>` -elementet gir oss muligheten til å legge til mer innhold (for eksempel bilder).

Hvis vi slår sammen det vi har sett på så langt, får vi et enkelt skjema som ser slik ut:

```
1 <div id="hoved">
2   <h1> Registrer film </h1>
3
4   <label for="tittel">Tittel</label>
5   <input type="text" id="tittel" placeholder="Titanic">
6
7   <label for="beskrivelse">Beskrivelse</label>
8   <textarea id="beskrivelse" placeholder="Skriv en kort
  beskrivelse av filmen"></textarea>
9
10  <label for="aarstall">Årstall</label>
11  <input type="number" id="aarstall" placeholder="1997">
12
13  <label for="aldersgrense">Aldersgrense</label>
14  <select id="aldersgrense">
15    <option value="alle">Tillatt for alle</option>
16    <option value="syv">7 år</option>
17    <option value="elleve">11 år</option>
18    <option value="femten">15 år</option>
19    <option value="atten">18 år</option>
20  </select>
21
22  <button>Registrer</button>
23 </div>
```

Vi skal senere se på hvordan vi kan bruke CSS på disse elementene, men vi må

1x

legge til noe CSS allerede nå, ellers havner alt bortover på samme linje, og det blir vanskelig å bruke skjemaet. Vi har derfor lagt til denne CSS-koden i eksemplet ovenfor:

```
input, textarea, select {  
  display: block;  
  margin-top: 2px;  
  margin-bottom: 10px;  
}
```



Ved å angi `display: block;` for disse tre elementene, vil de plasseres fint nedenfor hverandre. Teksten vi har lagt til, legger seg fint mellom elementene. Vi har også lagt til litt `margin` ovenfor og nedenfor elementene for å få litt luft. Prøv selv å fjerne denne CSS-koden fra eksemplet ovenfor, slik at du kan se hvordan elementene oppfører seg uten noe CSS.

Oppgaver

- 1 Du skal lage en nettside som lager visittkort for elever på skolen din. Lag først et skjema med følgende innhold (legg gjerne til mer):

- et tekstfelt for registrering av personens navn
- et tallfelt for registrering av personens telefonnummer
- et tekstfelt for registrering av en kort tekst om personen
- en nedtrekksmeny for valg av trinn (Vg1, Vg2 eller Vg3)
- en knapp for generering av visittkort