

Cykelskuret

I dag skal vi lave et virtuelt cykelskur. Det er muligt at tilføje cykler til en liste og fjerne dem igen:

Cykelskuret

Navn

Antal gear

Tilføj cykel

- X Centurion - 7 gears
- X Det Gule Lyn - 1 gears
- X Havelågen - 3 gears

Opgaven er opdelt så de simpleste ting kommer først - og alle de mere spændende ting kommer som bonusopgaver for de modige.

Til at starte med skal vi have en HTML-side som danner basis for vores applikation. Den indeholder selvfølgelig et `<html>`-element med et `<head>` og et `<body>` i. I vores `<body>` placerer vi tre elementer:

- et `<form>`-element, som skal indeholde vores input-felter
- et ``-element, som skal indeholde listen af cykler (ul = **u**nordered **l**ist)
- et `<script>`-element, som peger hen på al vores JavaScript

Start med at fylde et `<input>`-felt i vores form - dette skal indeholde navnet på cyklen - og en `<button>`, som man kan bruge til at submitte vores form med. Hvis du vil gøre det lidt pænt, kan du indkapsle det med et `<fieldset>` som har en `<legend>`, så kommer det til at se flot ud som ovenover (brug Google for mere info).

Sørg for at både dit `<input>`-felt og ``-feltet har et *id* (f.eks. hhv. *navn* og *cykelListe*).

Først skal vi sørge for at oprette cyklen i vores liste. Lav en ny JavaScript-fil (kald den f.eks. *bikeshed.js*) og brug attributten *src* til at referere til den i dit `<script>`. I denne fil laver du et nyt array kaldet *bikelist* og en funktion kaldet *addBike*, som tager et *event* som parameter. Sæt attributten *onsubmit* på din `<form>` til at kalde denne funktion og test at det virker (du kan f.eks. bare have en *console.log* i funktionen og se at der bliver skrevet noget ud i dev tools).

Så langt, så godt. Vores *addBike*-funktion skal som minimum gøre følgende:

- sørge for at eventens default (at reloade siden) ikke bliver udført
- hente navnet ud fra `<input>`-feltet
- lave et *bike*-objekt med dette navn
- lægge dette objekt i vores *bikelist*
- opdatere vores `` så det reflekterer indholdet af *bikelist*

Lad os tage tingene fra en ende af.

Droppe eventens default

Vi er ikke interesseret i at siden bliver reloadet - hvilket ellers er standard når man submitter en form - da al vores JS (og indholdet af *bikelist*) så bliver nulstillet. Dette gøres heldigvis nemt med metoden *preventDefault* på vores *event*-objekt. Videre!

Hente navnet fra <input>-feltet

Vi kan bruge `document.getElementById` til at få en DOM-reference til <input>-feltet - du huskede at give feltet et *id*, ikke? Derefter kan vi bruge *value* på denne reference til at hive navnet ud.

Oprette et bike-objekt

Vi kunne godt lægge navnet direkte ind i vores liste med cykler, men vi vil gerne have muligheden for senere at lægge andre værdier ind pr. cykel, så derfor laver vi et objekt. Dette gøres ret nemt på følgende måde:

```
const bike = {  
  name: name  
};
```

hvor *name* her er værdien vi hentede tidligere.

Lægge objektet i bikelist

Nu vi har dette objekt, kan vi bruge metoden *push* på vores array til at lægge det i listen.

Opdatere UI så den viser indholdet af bikelist

Nu kommer vi til det sjove. Lav en ny funktion kaldet *updateBikes*, som kaldes fra *addBike*. Denne funktion skal nu render hele vores liste på HTML-siden.

Brug først `document.getElementById` til at få en reference til vores , og sæt dennes *innerHTML* til en tom string - på denne måde fjerner vi alle de tidligere elementer, så vi kan udfylde vores UI på ny baseret på indholdet af vores nu opdaterede *bikelist*.

Løb derefter igennem samtlige entries af vores *bikelist* (f.eks. med *forEach*) og lav et nyt med navnet på den aktuelle cykel. Brug `document.createElement`, `document.createTextNode` og `appendChild` til at hjælpe dig, og lad være med at være bange for at lave hjælpefunktioner for at splitte koden lidt op og gøre den mere læsbar. Husk at bruge `appendChild` på dit -element for at knytte dine nylavede 'ere til denne.

Mestersektionen

Det var den nemme del. Hvis du er frisk på mere, så lav en eller flere af følgende løsninger i programmet:

- lav en dropdown hvor man kan vælge antal gear på cyklen (hint: brug <select> og <option>) og sørg for at disse bliver skrevet ud sammen med cyklens navn
- lav validering af at navnet f.eks. ikke er blankt eller allerede eksisterer i listen, og skriv en fejltekst ud hvis nødvendigt (brug f.eks. et tomt -element ved siden af dit input)
- Slet værdien i input-feltet når brugeren har indtastet en cykel, og brug `focus()` på elementet for at sætte cursoren der med det samme
- Lav en slette-funktion, som gør det muligt at fjerne cykler fra listen igen (ved at trykke på et X ud for en cykel, hvormed den så bliver fjernet fra *bikelist*)