

WORKSHOP

Utvikling av interaktive nettsteder. Fredag 20. januar.

Agenda

08:15, Aud 2: Gjennomgang av oppgaver

08:45: Fordeling i klasserom, veiledet arbeid med oppgaver

11:30, Aud 2: Ta med lunsj for gjennomgang av løsningsforslag

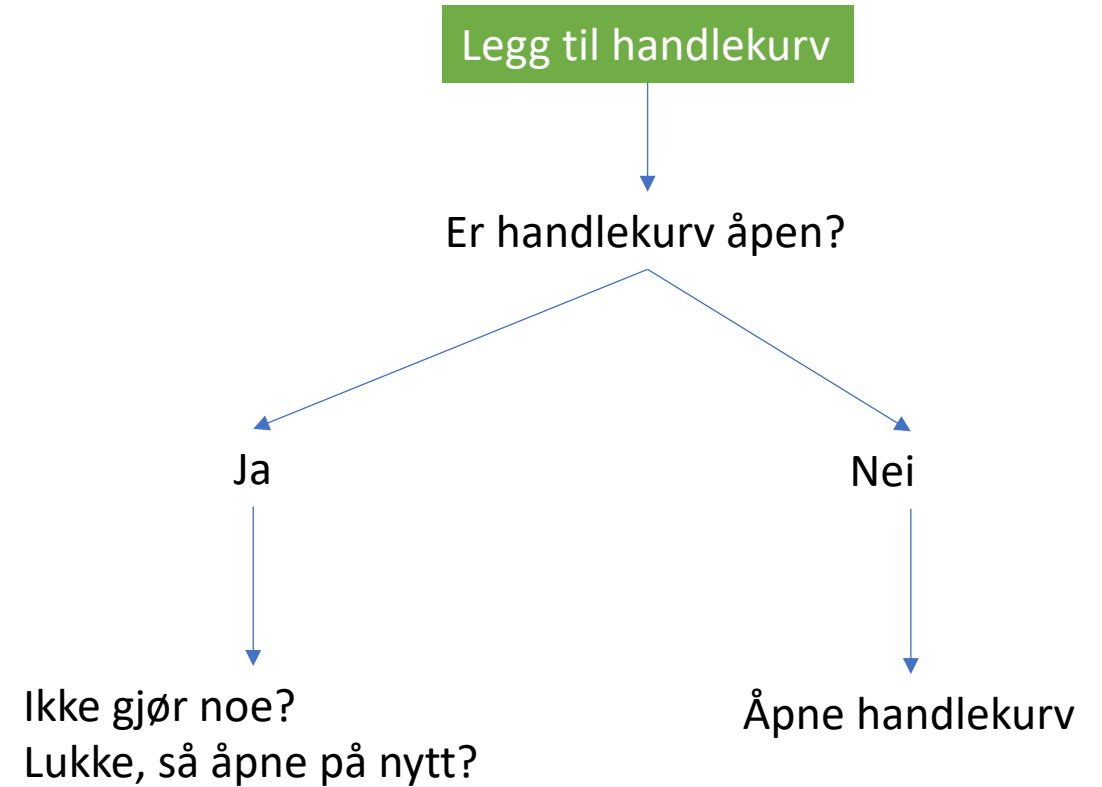
OPPGAVE 1

Åpne handlevognen når et nytt produkt legges til

Hva i koden "åpner handlevognen"?

Når må vi kalle på denne koden?

Siden vi "toggler" (slår av og på visningen), hva skjer dersom vi allerede har handlevognen åpen når vi legger til et nytt produkt?



OPPGAVE 2

Oppdater nødvendig kode for å få total pris i handlekurven.

Hvor i koden får vi tak i pris for hvert produkt?

Hvordan legger vi sammen prisene?

Når skal vi skrive ut total pris?

Hvor skal vi skrive ut total pris?

HINT:

```
let totalPrice = 0
```

*.map() henter alle produkter,
her kan vi få tak i pris...*

OPPGAVE 3

Lag slett-funksjonalitet som knyttes til X-knappen for hver produktlinje

Når en knapp trykkes, må koden vite hvilket product som skal slettes.

Det må finnes funksjonalitet for å slette produktet, men hva betyr rent praktisk "å slette"?

Når noe endres i produktvisningen, må handlevognen og antallet oppdateres.

HINT:

```
Array.splice()
```

Sende index som parameter i arrow-function i .map()?

Hva er denne parameteren?

```
Function deleteProduct(product)
```

Når må denne funksjonen kalles/kjøres?

Oppgave 3.1 (Ikke obligatorisk)

Få et produkt til å fade ut fra handlevognvisningen når man klikker slett (øvelse I å koble CSS og JavaScript)

Dette er visuelt lureri. Vi må bruke CSS for å fade ut (gi `` klassen “hidden”). Men – vi må fortsatt fjerne produktet fra `cart-arrayen`, men ikke før vi har faded det fra visningen.

HINT:

En kombinasjon av dynamisk id på ``-elementet (`<li id=“prod-0”>`, `<li id=“prod-1”>` osv) som kan brukes som selector i `deleteProduct`-funksjonen, `classList.add()`-metoden i kombinasjon med `hidden`-klassen vi allerede har laget og `setTimeout()`-funksjonen løser denne oppgaven.

PS: Du må også legge på `transition`-regler i CSS-reglene til ``.

Avanserte oppgaver

Avanserte oppgaver

- I de avanserte oppgavene skal vi gjøre innholdet enda mer dynamisk.
- Hent ned legodudes.js fra kursets GitHub-repository (ligger i legodudes_webshop-mappen)
- Kopier denne inn i ditt webshop-repository
- Bytt ut referansen til shopscripts.js i index.html-filen med legodudes.js

De videre oppgavene gjør visningen av innholdet og funksjonaliteten i nettbutikken betydelig mer dynamisk.

Oppgave 4

Lag nødvendig funksjonalitet for å liste opp produktene i products-arrayen i legodudes.js.

Hvilket HTML-element må vi skrive dem til?

Hvilke metoder må vi benytte?

Hvilke funksjonskall må opplistingen ta høyde for?

Oppgave 4: Tips

- Produktbilder: Mye av bilde-URLen er statisk. Sammenlign URL fra HTML-en for å lage riktige URL-er med informasjonen fra products-arrayen
- HTML-en for produktene skal være veldig lik HTML-en fra de forrige oppgavene, men nå skrives ut ved hjelp av JavaScript og products-arrayen.
- Du kan trenge en `<p>`-tag rundt Pris- og Antall-utskriften for å ha en `` vi kan bruke til kun utskrift av de dynamiske verdiene (tallene som anger pris og antall).
- Du kan nå trenge en flex-container i `<main>` med en ID som kan velges for produktutskriften.

Oppgave 5

- 1) Legg til antall på lager i visningen av produktet. Ta høyde for at tallet som vises skal oppdateres.
- 2) Dersom et produkt ikke finnes på lager (storage: 0), skal Legg i handlekurv-knappen ikke være tilgjengelig (disabled) og grå
Tips: sjekk CSS for disabled-status...

PS: Sett et av produktenes storage til 0 for å teste at dette fungerer.

Oppgave 6

Lag “legg til handlekurv”-funksjonalitet.

Vi ønsker nå kun en produktlinje per produkt. Det vil si: dersom et produkt med id 1 ligger i handlekurven, og vi forsøker å legge til et nytt produkt med samme id, skal antallet på denne varelinjen oppdateres.

Det skal ikke legges til en ny produktlinje med det samme produktet. Når et produkt legges til handlekurven, må antall på lager minskes i product-arrayen.

Hint: Når man legger til produkt i handlekurv, sjekk om prodid allerede finnes i cart-arrayen.

Oppgave 6: Tips og hint

- Når man legger til produkt i handlekurv, sjekk om prodid allerede finnes i cart-arrayen.
- Her kan man få god nytte av `Array.findIndex()`-metoden for å finne plassen til produktet man klikker på.
Tenk slik: `cart[index].id` vil vise til en prodid i `products-arrayen`.
- Hvis du har løst oppgave 4 ved å legge ting i en funksjon, burde oppdatering av produktlisten være veldig enkel. Dette er dog ikke noe krav, det finnes ulike måter og oppdatere antall på.

Oppgave 7

Lag slett-funksjonalitet for en produktlinje i handlekurven. Denne skal aktiveres når slett-knappen på en linje klikkes.

OBS: Her kan store deler av funksjonen fra oppgave 3 benyttes.

HUSK: Når vi sletter en produktlinje fra handlekurven, må antallet produkter på lager økes.

Oppgave 7: Tips

Vi skal ved flere anledninger oppdatere lagerantall. Kanskje en funksjon som tar imot produkt og lageroppdatering er fornuftig, så vi kan kalle denne enkelt når vi skal oppdatere lager? (se også tips for oppgave 8)

Igjen vil `Array.findIndex()`-metoden være nyttig. Vi kan bruke index for å finne hvilke rader i `cart` og/eller `products`-arrayen som må oppdateres.

CITY

NINJAGO

CASTLE AND KNIGHTS

MARINE AND PIRATES

MO

NINJAGO



Ninjago

DRAGON ZANE

Kr. 89,-

Antall på lager: 3

Legg i handlekurv



Ninjago

LIL'NINJA

Kr. 99,-

Antall på lager: 1

Legg i handlekurv



Ninjago

MASTER WU

Kr. 109,-

Antall på lager: 3

Legg i handlekurv

Din handlevogn

Dragon Zane 89,- - 2 + X

Lil'Ninja 99,- - 1 + X

Totalt: Kr. 277,-

Oppgave 8

Gjør om visningen av antall produkter i handlevognen til å vises i et input-felt.

Til venstre for input-feltet, legg en knapp med minustegn. Ved klikk på knappen, reduser antall med en. Oppdater storage-antall for produktet med +1.

Hvis antall blir null (antall er 1 , klikk på knapp gir - 1), skal produktlinjen slettes.

Til høyre for input-feltet, legg en knapp med plusstegn. Ved klikk på knappen, forhøy antall med en. Oppdater storage-antall for produktet med -1.

Oppgave 8: Tips

- Du må antagelig oppdatere både HTML og CSS for å få en ok visning av knapper og input i handlekurven.
- Husk på regel «minus og minus gir pluss» fra matematikken. Denne gjør det mulig å lage en funksjon for oppdatering av antall som tar en parameter enten det er -1 eller +1 som gjelder, og kan oppdatere både handlekurven og produktlisten.
- Hvis det er minus i handlekurven, må det bli pluss i produktlisten, og omvendt.