

Løsningsforslag Eksamen Vår 2018

Løsningen bruker et bildebehandlingsprogram i Oppgave 1 og 3, og et lydredigeringsprogram i oppgave 3. Du finner mer informasjon om bildebehandling under kapittel 6 på elevnettstedet og lydredigering under kapittel 13.

Nedenfor finner du kommentarer til hvordan vi har valgt å løse de tre oppgavene. Du finner også utfyllende kommentarer i hver av kodefilene.

Litt om stilark

Vi har samlet alle stilarkene i en mappe som heter stilark. For å få tilgang til stilarkene fra de andre mappene, så må vi skrive følgende:

```
<link rel="stylesheet" href="../stilark/hovedstil.css">
```

Legg merke til begynnelsen: «../stilark/». Prikkene gjør at vi «går opp» én mappe, slik at vi så kan gå inn i mappen stilark og finne CSS-dokumentene der.

Det er alltid en god idé å lage slike stilark i forkant av eksamen. Det vil spare mye tid.

Menyen

I tidligere eksamener har det ofte vært greit å løse oppgavene hver for seg. I årets eksamen skal løsningene knyttes sammen, og en måte å gjøre det på, er å bruke en meny. Vi har laget en enkel meny ved å bruke `flexbox`. Denne framgangsmåten er beskrevet på side 262 i læreboka.

Oppgave 1

Det er to utfordringer i denne oppgaven. Den ene er å vise fram informasjon når vi fører musepekeren over et ikon, den andre er å kunne bla gjennom de fire bildene.

For å vise fram informasjon for hvert av ikonene, har vi valgt å knytte en `mouseover`-hendelse til ikonbildene. Vi bruker et annet element til å vise fram teksten som hører til ikonet vi har ført musepekeren over. De to `<div>`-

elementene som inneholder ikoner og beskrivelse er plassert ved siden av hverandre ved hjelp av `flexbox`.

For å bla gjennom de fire bildene har vi laget to knapper: en som blar bakover (forrige bilde) og en som blar framover (neste bilde). Bildene er nummerert, så det er greit å bla fra et bilde til et annet, men vi må huske å håndtere situasjonen der vi vil gå til forrige bilde fra det første bildet (bildet `intro0.jpg` eksisterer ikke), og til neste bilde fra det siste bildet (bildet `intro5.jpg` eksisterer ikke). Vi har valgt å bruke et tall for å holde orden på hvilket bilde som vises.

Oppgave 2

Dette er en typisk «kalkulator»-oppgave. Vi må hente inn informasjon fra brukeren og utføre noen beregninger basert på hva brukeren velger.

Vi har brukt en ferdiglaget funksjon (`lagTabell()`) for å generere en tabell fra arrangements-informasjonen. Vi må ofte lage tabeller i eksamensoppgaver, så det er veldig lurt å ha en slik funksjon klar på forhånd. Akkurat denne funksjonen tar tre argumenter:

- HTML-elementet vi ønsker å plassere tabellen i
- En array med overskrifter (like mange overskrifter som det skal være kolonner i tabellen)
- En array med objekter (der hvert objekt har de samme egenskapene)

Vi har også brukt et stilark for å gi utseende til tabellen.

Videre lager vi en nedtrekksmeny som lar brukeren velge et arrangement, to tallfelter der brukeren kan oppgi antall voksne og antall barn, og en knapp.

Vi henter så inn alle verdiene brukeren har oppgitt, og beregner totalpris. Utskriften tilpasser vi etter brukerens valg.

Vi blir også bedt om å sikre at alle felt inneholder gyldige data. Vi er sikre på at nedtrekksmenyen har gyldig informasjon, fordi vi har kontroll over informasjonen. Vi må derfor sjekke at brukeren oppgir fornuftige tall i de to tallfeltene. Vi vet at det ikke vil gi mening å bestille et negativt antall billetter, så vi må sjekke at tallene brukeren oppgir er 0 eller større. Hvis vi hadde visst noe om maksimal eller ledig kapasitet på arrangementene, så hadde det også vært nyttig å sjekke at ingen bestiller for mange billetter.

Oppgave 3

Denne oppgaven ber om en pseudokode. Ved å skrive den først kan det bli lettere å løse oppgaven. En pseudokode for denne oppgaven kan se slik ut (det finnes mange måter å gjøre dette på, denne varianten er ganske muntlig):

Nettsiden viser x instrumenter og en lydspiller

Lag en array med alle instrumentnavnene

```
// Velg et tilfeldig instrument som kan spilles av med lydspilleren:  
velgTilfeldigLydklipp()
```

HENDELSE: et bilde klikkes

HVIS bildet hører til lyden som spilles av

Riktig:

antallRiktig++

Ta bort instrumentet fra arrayen over instrumenter

Skjul bildet av instrumentet (slik at det ikke kan klikkes igjen)

HVIS dette var det siste bildet

Avslutt quizen

ELLERS

velgTilfeldigLydklipp() // velg et nytt instrument

HVIS spillet er ferdig (alle instrumenter gjettet riktig)

Undersøk om vi har en ny highscore

Skriv ut passende beskjer

Tilby brukeren å spille igjen (startNyttSpill())

ELLERS

Feil:

antallFeil++

FUNKSJON velgTilfeldigLydklipp()

```
// Kode som velger et tilfeldig instrument fra arrayen over  
instrumenter
```

FUNKSJON startNyttSpill()

```
// Kode som starter det hele på nytt (uten å tømme highscore-verdien)
```

Selve programmet er laget rundt denne arrayen:

```
var instrumenterOriginal = ["fagott", "floyte", "klarinet", "obo",  
"valthorn"];
```

... og en tom array som brukes under kjøringen av programmet:

```
var instrumenter = [];
```

Denne arrayen fylles opp av de samme instrumentene som i den første arrayen. Vi må gjøre det slik fordi det skal gå an å starte opp spillet på nytt, og da må vi ha tilgang til en liste over alle instrumentene (som vi har i den første arrayen).

Vi har valgt å lage en egen funksjon: `startNyttSpill()`, som tar utgangspunkt i denne arrayen og genererer et bilde (med tekst) for hvert instrument. Tilsvarende bruker vi funksjonen `velgTilfeldigLydklipp()` for å hente ut et tilfeldig instrument fra den samme arrayen. Hver gang en bruker gjetter riktig, fjerner vi et instrument fra `instrumenter`-arrayen, slik at lyden til det instrumentet ikke kan bli valgt igjen.

Resten av løsningen er beskrevet i kommentarer i koden.

Til slutt blir vi bedt om å lage en testplan. Dette er et begrep som ikke er godt definert av læreplanen, men vi må anta at det dreier seg om å lete etter mulige feil i applikasjonen. En testplan som gjør det, kan se slik ut:

- Velg feil bilde og sjekk at alt går som forventet
- Velg riktig bilde og sjekk at alt går som forventet
- Sjekk at du ikke kan velge et bilde igjen (hvis det var riktig tidligere i spillet)
- Sjekk at et lydklipp som har vært riktig gjettet ikke spilles av igjen
- Spill gjennom hele quizen, og se at den avslutter som forventet
- Spill en gang til, med færre feil, og se at du får ny highscore
- Start spillet på nytt og se at forrige highscore fortsatt ligger inne

I tillegg er det lurt å bruke `console.log()` aktivt under testing av et program. Alle variabler som forandrer seg underveis (for eksempel `antallRiktig`, `antallFeil` og `highscore`) bør skrives til konsollen når applikasjonen testes.