

Uppgift 3

Det mesta berättas, eller kan ses, på videon.

Här kommer några specifika ytterligare instruktioner.

Inlämning

Som för tidigare uppgifter måste du lämna in din egen uppgift. Det är dock tillåtet, och uppmuntrat, att lösa uppgiften tillsammans med andra. Det är helt ok att lämna in samma kod som någon annan.

Tentamen precis efter

Precis efter deadline för uppgiften har vi lagt en tentamen. Den är individuellt och det är strikt förbjudet att kontakta någon annan (student, kompis, etc) under tentamen. Tänk er att ni sitter i sal. Ett bra sätt att förbereda sig för tentan är att jobba hårt med uppgiften. Se till att du förstår all kod du lämnar in.

Påminnelse: Dagen efter tentamen kommer vi att ha redovisning. Redovisningen är individuellt (du redovisar för Sebbe eller Erik) och vi kommer att ställa frågor för att säkerställa att du är den som har kodat svaren till tentamen.

Det lilla extra: Projektgrupper

Projekten kommer att genomföras i grupper av studenter som befinner sig på ungefär samma programmeringsnivå. Det bästa är om vi kan få input från olika håll om vilken nivå du befinner dig på. Ett av input är huruvida du hinner göra något extra i denna uppgift, och vad det är. Så, när uppgiften är avklarad:

1. Zippa och skicka in det du har så att du fixar G (glöm inte specifikationerna, se nedan)
2. Om du vill: Bygg på med något extra. Du har helt fria händer: det kan vara något kul på home-sidan, eller att du adderar någon funktionalitet på student- eller course-sidan. Denna del måste genomföras individuellt, vi förväntar oss helt olika koder (inte bara att ni ändrar på funktionsnamn eller ordningen i vilken de deklarerar och såna lame lösningar).

Specifikationer

I videon presenterar jag några funktionaliteter som appen måste ha. Den första delen av uppgiften är att skriva ner dem i en lista. Listan över specifikationer ska skrivas i en fil som heter specifikationer.txt. Denna fil måste inkluderas i zip-filen som du skickar.

Kodbas

- A. Det finns ett flertal filer redan på plats. Du får inte skapa någon ny fil.
- B. Det finns kod i dessa filer:
 - A. HTML-filer: Dessa får inte ändras överhuvudtaget. Om du vill skapa något på home-sidan ska det göras via index.js.
 - B. CSS-filer: Dessa måste fyllas i (nav.css är ifylld, använd den koden eller ändra den om du vill). Instruktioner:
 - A. style.css ska innehålla alla CSS-regler som berör element som finns på både students- och courses-sidan.

- B. students.css ska innehålla CSS-regler som endast berör element på student-sidan
- C. courses.css ska innehålla CSS-regler som endast berör element på courses-sidan
- D. index.css: Om du vill göra något med home-sidan.
- C. JS-filer: Som du ser finns det en js-fil för varje sida på webbplatsen: index.js (home), students.js och courses.js. Koden som finns på plats får inte ändras och måste användas. Sedan måste det tillkomma en del ny kod som ska fungera med den som finns. Tanken med detta är att tvinga dig tänka på tillräcklig hög abstraktionsnivå.

Tips

Börja med att skissa, på papper, strukturen som koden ska ha. Den måste ta sin utgångspunkt i kodbasen som redan finns på plats.

- A. Vilka instruktioner ska exekveras vid uppladdning? (Initialising code)
- B. Vilka funktioner ska finnas:
 - A. Vilka argument tar de emot
 - B. Vad gör dem (ändrar DOM-element på sidan, skapar DOM-element, söker information i arrayer, etc)
 - C. Vad returnerar de
 - D. Var de anropas (använd pilar för att få en klar bild av detta)
- C. Event-handlers:
 - A. Vilka element ska ha eventlisteners
 - B. Vad ska hända när events triggas
- D. CSS:
 - A. Vilka HTML-element behövs för att visa det som ska visas på sidan
 - B. Vilka CSS-regler ska användas för att formatera elementen
 - C. Vilka CSS-variabler behövs (om flera element delar samma formatering)
 - D. Vilka regler ska gå i style.css (gemensamma för alla sidor) och vilka ska gå i CSS-filerna för de specifika sidorna (courses.css, students.css)

Lycka till!