

LAB2: (M)Modell - (V)Presentasjon - (C)Kontroller

PUBLISERT: 13.09.2013

FRIST: **Torsdag 19.09.2013 kl. 23:59 i Class Fronter**

Innledning

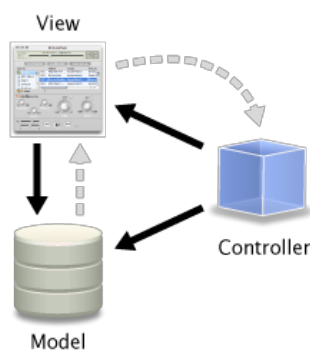
I den forrige oppgaven har dere sett på at det ofte kan være (det er ikke sagt at det alltid er) en god ide å adskille presentasjon fra innhold. Det er presentasjon som er grensesnittet mot brukeren.

Programmeringsspråk

Velg:

Du har selektert programmeringsspråket BASIC

Presentasjon blir laget av en applikasjon som mottar kodet informasjon (HTML og eventuelt CSS, JavaScript) og basert på det opptegner¹ denne applikasjonen et skjermbilde som vi også kaller for brukergrensesnitt. I Verdensveven (WWW) mottar en nettleser kodet informasjon (HTML eventuelt med CSS, JavaScript og utvidelser², på engelsk også kalt “plugins”) fra en nett-tjener (“webserver”) og gjengir det vha. grafikk-kort på en skjerm. Så kan en bruker se på innholdet eller innlede en interaktiv sesjon med innholdet gjennom brukergrensesnittet. Dvs. at brukeren med spesielle handlinger, som for eksempel, å manipulere grafiske elementer med mus eller skrive inn en tekst i et tekstfelt, kommuniserer med innholdet gjennom presentasjonen (eller **presentasjonslaget**, som man også noen ganger sier). Innholdet kaller man ofte for **modell**, dvs. det er en modell som representerer



(lagrer eller mellomlagrer) informasjon som brukeren enten legger inn (skjema) eller “tar ut” (lese og surfe). Innholdet (modell), akkurat som presentasjon er avhengig av brukerkrav (kravspesifikasjon).

Når man implementerer applikasjoner på konkrete teknologiplatformer (maskinvare + operativsystem), bruker man også et tredje lag, - **kontrolleren**. Da kaller man hele fremgangsmåten for organisering av applikasjonen for MVC (Model - View - Controller eller i et forsøk på å oversatte det til norsk, - Modell - Presentasjon - Kontroller) (se Bilde fra <http://www.cocoalab.com/?q=node/24>).

Kontrolleren “tar inn” informasjon (ofte i form av hendelser) fra brukergrensesnittet og bruker

¹ På engelsk brukes det begrepet “render”, som til norsk kan oversettes som opptegne, lage, gjengi

² F. eks. Java Applet, Adobe Flash

modellen for å returnere en presentasjon. Disse tre lagene er logiske strukturer som kan bidra til et bedre oversikt av kildekode, samt gjøre det lettere å videreutvikle applikasjonen modulært. Det er ingen fasit på hvordan man skal strukturere informasjon i en applikasjon, men MVC er mye brukt i applikasjonsutvikling som et grunnleggende prinsipp for systemarkitektur. Du skal reflektere om MVC senere i oppgaven.

Oppgave

Oppgave består av 2 deloppgaver, 1 og 2. Du kan fortsette å bruke filene fra Lab1. Gjør de endringene du trenger og gjør en *commit* i git med dine kommentarer, slik at du lager en ny versjon av filene for hver deloppgave. Det skal klart forekomme fra dine kommentarer i *git log*, hvilken versjon av deloppgaven denne *commit*-en gjelder.

Hvis du legger til nye filer i din *git repo*-en, husk også å endre din README-fil. En README fil skal alltid enkelt forklare hva er din applikasjon (i dette tilfelle en webapplikasjon for å presentere programmeringsspråk) og hvordan den er bygd opp (i dette tilfelle, hvilke filer den består av samt kort beskrivelse av hva hver fil inneholder).

Hvis du ikke har klart for deg hvordan å gjøre dette, spør hjelpelærer eller lærer om hjelp.

KRAVSPESIFIKASJON: Informasjonsside om programmeringsspråk

I forrige oppgave var det gitt følgende forsøk på et kravspesifikasjon:

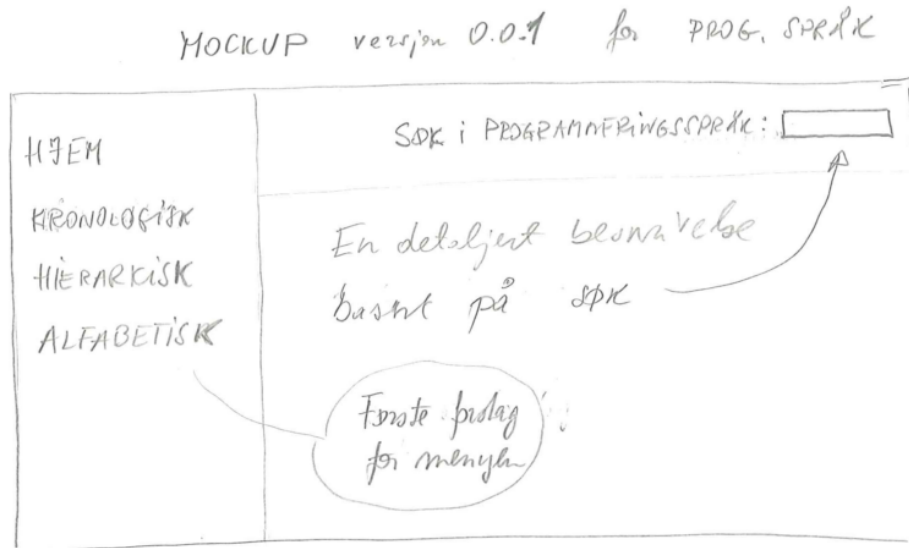
“En brukergruppe ønsker å lage en side om programmeringsspråk. Siden det finnes mange programmeringsspråk, skal en liste av alle programmeringsspråkene presenteres på en brukbart måte i første omgang.”

Anta at denne brukergruppen er lærere i en vidergående skole, som ønsker seg en nettside om programmeringsspråk. Siden skal **senere** (neste iterasjon) presentere programmeringsspråk kronologisk (etter tid de oppsto), hierarkisk (geneologisk) og alfabetisk (en liste man kan selekttere fra, eventuelt søke i).

1 [Formål: å lære om overgangsprosessen fra et designforslag til kode; å øve på HTML og CSS]

I denne deloppgaven, skal du modifisere eksempel fra <http://www.w3.org/Style/Examples/011/firstcss> (Bos, 2011) slik at den presenterer desingforslaget fra brukere (se tegning / mockup på neste side).

Modifiser din css-fil og legg til alle stildefinisjonene i denne filen. Kommenter hva hver linje gjør i denne filen (husk at kommentar i CSS defineres med `/* ...her går kommentar... */`)



Selv om du kan kopiere koden fra nettsiden og lime den inn i din tekstbehandler, anbefales det at **du skriver inn kode selv, tegn for tegn**. Da vil du huske syntaksen bedre og du vil også bli bedre til skrive på tastatur, - noe som du kan dra fordel av senere.

Vær oppmerksom på at eksempel fra W3C bruker HTML 4.01 som spesifikasjon og den krever ikke at taggene lukkes. For eksempel kan man skrive `<p>` uten å bruke `</p>` på slutten av et avsnitt. Du kan bruke HTML5 spesifikasjon i din besvarelse.

Sjekk <http://colorshemedesigner.com/> for å gjøre et godt fargevalg (fargene trenger ikke være de samme som er brukt i W3C guiden <http://www.w3.org/Style/Examples/011/firstcss>).

2 [Formål: utvikle en forståelse for hvordan JavaScript fungerer; øve på å skrive kode i JavaScript]

I denne versjonen (*commit*-en i git), bruk koden fra LAB1 for å implementere element for "Søk i programmeringsspråk" som en nedtrekksliste. Et eksempel på resultatet ser du her (du kan selv velge fargemodell og eventuelt andre layout-detaller):



Når man selekterer et programmeringsspråk fra nedtrekkslisten, skal teksten “Velkommen til en mystisk verden av programmeringsspråk” erstattes med “Du har selektert programmeringsspråket <verdien_fra_nedtrekksliste>”. Bruk JavaScript fra Lab1 og endre din JavaScript fil og gjør en *commit* i git med tilsvarende kommentar.

Hvordan kan du bruke JavaScript for å gi brukeren en respons på at sidene “Kronologisk”, “Hierarkisk”, “Alfabetisk” er under konstruksjon? Hvordan kan man få det til slik at “Hjem” går tilbake til seg selv hvis man er på hjemmesiden når man trykker på denne lenken? Implementer det i koden (ny versjon av din JavaScript fil og en ny *commit*).

Kontrollspørsmål (besvares i README-filen)

Forklar hvilken deler av din applikasjon så langt, implementerer de forskjellige lagene i MVC arkitekturen. MVC står for Model-View-Controller, som til norsk kan oversettes til Modell-Presentasjon-Kontorller.

Hva skal leveres og hvor?

Den siste versjon av alle filene skal legges inn i GitHub med tilsvarende kommentarer. README-fil skal være oppdatert.

REFERANSER (oppdatert versjon fra LAB1)

- Bokmålsordboka. Sist hentet 4.9.2012 fra <http://www.nob-ordbok.uio.no/perl/ordbok.cgi?OPP=tagg&bokmaal=+&ordbok=bokmaal>
- Bos, B. (2011). CSS Tutorial: Starting with HTML + CSS. Sist hentet 12.09.2012 fra <http://www.w3.org/Style/Examples/011/firstcss>
- Browser Shots. Sist hentet 27.08.2012 fra <http://browsershots.org/>
- Cascading Style Sheets. Sist hentet 28.08.2012 fra

http://no.wikipedia.org/wiki/Cascading_Style_Sheets

- ColorPicker 3.1. Pagetutor.com. Sist hentet 28.08.2012 fra <http://www.pagetutor.com/colorpicker/index.html>
- Color Worqx. www.worqx.com. Sist hentet 28.08.2012 fra <http://www.worqx.com/color/index.htm>
- HTML. no.wikipedia.org. Sist hentet 28.08.2012 fra <http://no.wikipedia.org/wiki/HTML>
- Introduction to CSS 2.1. W3C Recommendation. Sist hentet 28.08.2012 fra <http://www.w3.org/TR/CSS2/intro.html>
- Notepad++. Sist hentet 27.08.2012 fra <http://notepad-plus-plus.org/>
- The World Wide Web Consortium. Sist hentet 27.08.2012 fra <http://www.w3.org/>
- XHTML Basic 1.1 Cheat Sheet. Sist hentet 28.08.2012 fra <http://www.w3.org/2007/07/xhtml-basic-ref.html>
- W3C Markup Validation Service. Sist hentet 28.08.2012 fra http://validator.w3.org/#validate_by_uri
- W3Schools.com. Sist hentet 27.08.2012 fra <http://www.w3schools.com>
- WAI-ARIA. ACCESSIBLE RICH INTERNET APPLICATIONS (WAI-ARIA) CURRENT STATUS. Sist hentet 4.9.2012 fra http://www.w3.org/standards/techs/aria#w3c_all