



*Avdeling for informatikk og e-læring, Høgskolen i Sør-Trøndelag*

## ***Introduksjon til Ajax og Web 2.0***

*Svend Andreas Horgen, Øyvind Hallsteinsen og Mildrid Ljosland*

*31.01.2012*

*Lærestoffet er utviklet for faget LN393D Ajax*

### **1. Introduksjon til Ajax og Web 2.0**

*Resymé: Denne leksjonen introduserer Ajax og begrepet Web 2.0. Vi ser på formålet med teknologien og hvilken nytte den kan gi oss, først og fremst sett fra et brukerståsted. Gjennom en kort historisk gjennomgang, eksempler, og en titt på hva vi tar for oss i de øvrige leksjonene, forsøker leksjonen å gi en motivasjon for ta i bruk Ajax.*

1.1.	INTRODUKSJON – HVA ER DETTE KURSET OG HVEM ER DET FOR? .....	2
1.2.	WEBEN SOM APPLIKASJONSPLATTFORM – HISTORISK UTVIKLING .....	2
1.3.	EKSEMPLER OG MULIGHETER MED AJAX .....	3
1.4.	WEB 2.0 .....	8
1.5.	INTRODUKSJON TIL FAGSTOFFET .....	9

## 1.1. Introduksjon – hva er dette kurset og hvem er det for?

Velkommen til kurset ”Ajax”! I hovedsak er dette et kurs hvor du lærer å utvikle nettsteder med Ajax. Med Ajax kan du lage nettsteder med en bedre brukeropplevelse, likt det vi kjenner fra vanlige skrivebordsapplikasjoner som du installerer på datamaskinen din.

Ajax bygger på kjente teknologier som HTML, JavaScript og XML. Dette ligger også i navnet; forkortelsen Ajax står for *Asynchronous JavaScript and XML*. Med grunnleggende kunnskap om HTML kan du komme i gang med enkle Ajax-løsninger som gir merverdi til en side. Det er også mulig å programmere komplekse applikasjoner i Ajax.

Gjennom leksjoner, eksempler, øvinger og aktivitet i LMS-systemet vil du lære om de ulike underliggende teknologiene og hvordan de sammen kan brukes for å lage Ajax-løsninger.

Kurset passer for de som har grunnleggende kjennskap til HTML og gjerne har litt programmeringserfaring. Du trenger ikke å være noen kløpper i programmering, men bør kjenne til bruken av variabler, kontrollstrukturer (if-then-else, while, for) og funksjoner. Vi vil ikke foreta noen opplæring i grunnleggende programmering i dette kurset, men du finner lenker til relevant informasjon på de to første leksjonenes ressursider. Programmering av Ajax-applikasjoner foregår i JavaScript, og vi gir derfor en kort innføring i dette språket i leksjon 2. I tillegg blir de fleste programmeringseksempler forklart i detalj i leksjonene.

## 1.2. Weben som applikasjonsplattform – historisk utvikling

World Wide Web ble offentliggjort i 1991 som resultat av et prosjekt internt i fysikklaboratoriet CERN. Forskeren Tim Berners-Lee hadde i samarbeid med andre utviklet weben for å muliggjøre utveksling av informasjon på en enkel måte. Som forskere hadde de behov for å dele informasjon uavhengig av tid og sted, og på tvers av i utgangspunktet inkompatible datamaskinplattformer. Internett basert på TCP/IP knyttet maskinene sammen. Weben med sine hypertekstlenker og ressurser i form av tekst, bilder og filer muliggjorde enkel deling av informasjon over Internett.

Selv om weben var laget for utveksling av informasjon, ble det raskt klart at systemet også kunne brukes med en større grad av interaktivitet enn ren klikking på hyperlenker. Innføring av en rekke kontrollelementer som knapper, nedtrekksmenyer og tekstfelt ga weben mange nye bruksområder. Dette ser vi i alt fra enkle søkefelt til mer komplekse applikasjoner som flybestilling og nettbank.

Sammenlignet med tradisjonelle skrivebordsapplikasjoner er imidlertid grensesnittet på den tradisjonelle weben mangelfullt. Tenk for eksempel på alle de avanserte mulighetene for interaksjon som finnes i et bildebehandlingsprogram, en tekstbehandler eller en medieavspiller.

Foruten begrensede interaksjonsmuligheter, må vi på tradisjonelle nettsteder også laste inn en ny side hver gang vi skal få respons tilbake fra webtjeneren. I kombinasjon utgjør disse to svakhetene at brukeropplevelsen blir begrenset, noe som har redusert mulighetene for hva vi har kunnet utføre av oppgaver på web.

På Figur 1 ser vi et eksempel på en webapplikasjon som ikke bruker Ajax, phpMyAdmin. phpMyAdmin brukes for å administrere MySQL-tjenere, og på figuren ser vi skjermbildet for å legge til, slette og redigere brukerkontoer. Dersom vi ønsker å slette en av brukerkontoene

må vi klikke flere steder, hvorpå websiden til slutt lastes på nytt med oppdatert innhold. Dersom dette var en vanlig skrivebordsapplikasjon kan vi se for oss at vi bare høyreklikket på den aktuelle brukerkontoen og valgte ”Slett”, hvorpå elementet i tabellen forsvant med en gang.



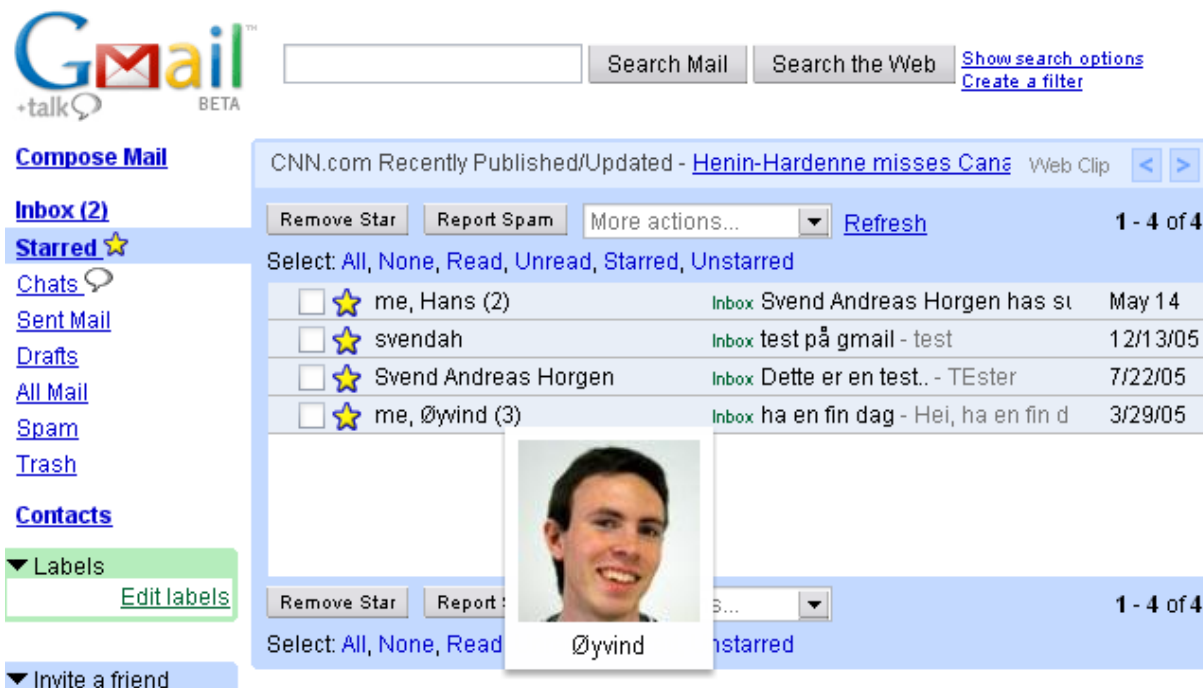
Figur 1: Webapplikasjon uten bruk av Ajax

Heldigvis har utviklingen ikke stått stille, og en rekke teknologier for å øke brukeropplevelsen i webbaserte applikasjoner har sett dagens lys. En mulig måte å øke brukeropplevelsen på, er å benytte Ajax. Med Ajax kan vi ha rike grensesnittelementer som glidekontroller, høyreklikkmenyer og dra-og-slipp-funksjonalitet. I tillegg kan det vi ser på skjermen endre seg uten at vi må laste siden på nytt, for eksempel ved at en linje i tabellen på figuren blir fjernet når en bruker blir slettet. En Ajax-applikasjon kan også kommunisere med webtjeneren i bakgrunnen, noe som for eksempel er en forutsetning for at den valgte brukeren faktisk skal bli slettet.

### 1.3. Eksempler og muligheter med Ajax

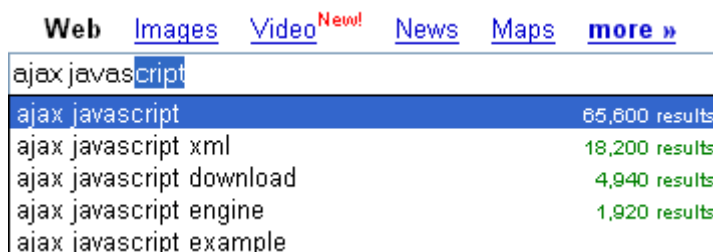
Stadig flere nettstedet legger helt eller delvis om til mer brukervennlige grensesnitt. Her følger en oversikt med en kort beskrivelse over kjente nettsteder som bruker Ajax på web.

Gmail (<http://gmail.com>) er en kjent mail-tjeneste fra Google som er fullstendig basert på Ajax. Brukeren får oppleve et intuitivt grensesnitt med funksjonalitet som minner om hva vanlige programmer kan by på. Dersom brukeren holder musepekeren over avsendernavnet på en person, vises et bilde av personen. Dette bildet hentes fra profilen til avsenderen, og er dermed et godt eksempel på asynkron, sømløs kommunikasjon med tjeneren. En hendelse hos brukeren/klienten (musepeker over et element) medfører at informasjon hentes fra tjeneren og vises til brukeren. Dette skjer uten at siden oppdateres. Bildet vises umiddelbart.



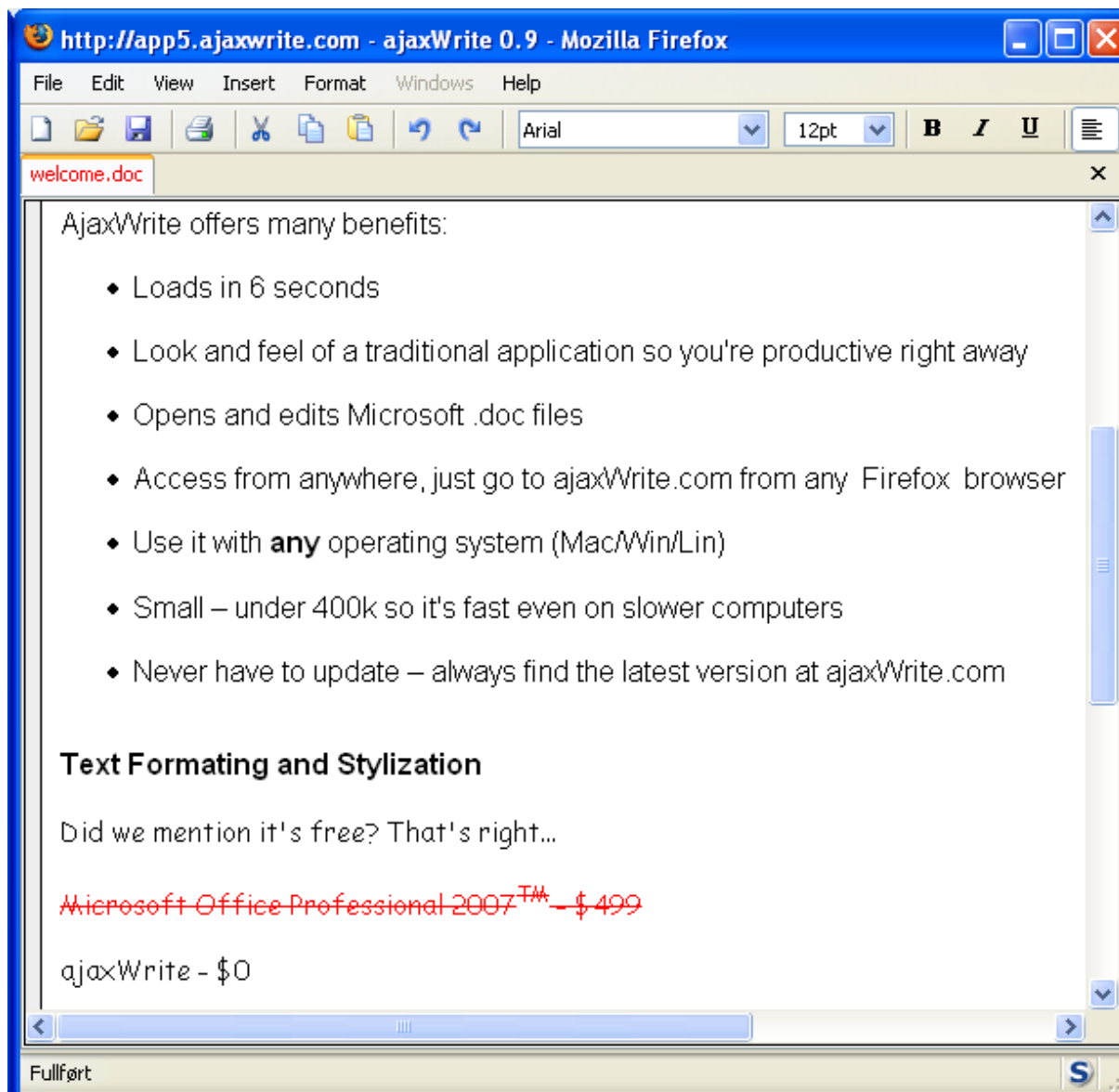
Figur 2: Dersom musepeker holdes over avsenderen (Øyvind), vil bildet som Øyvind har lagt inn i sin profil, vises umiddelbart.

Google har utviklet en rekke Ajax-baserte program som enten fungerer frittstående eller gir merverdi til veletablerte tjenester. Et eksempel på det siste er Google Suggest som lar brukeren skrive inn søkeord og deretter viser et estimert antall treff mens en skriver ("show as you type").



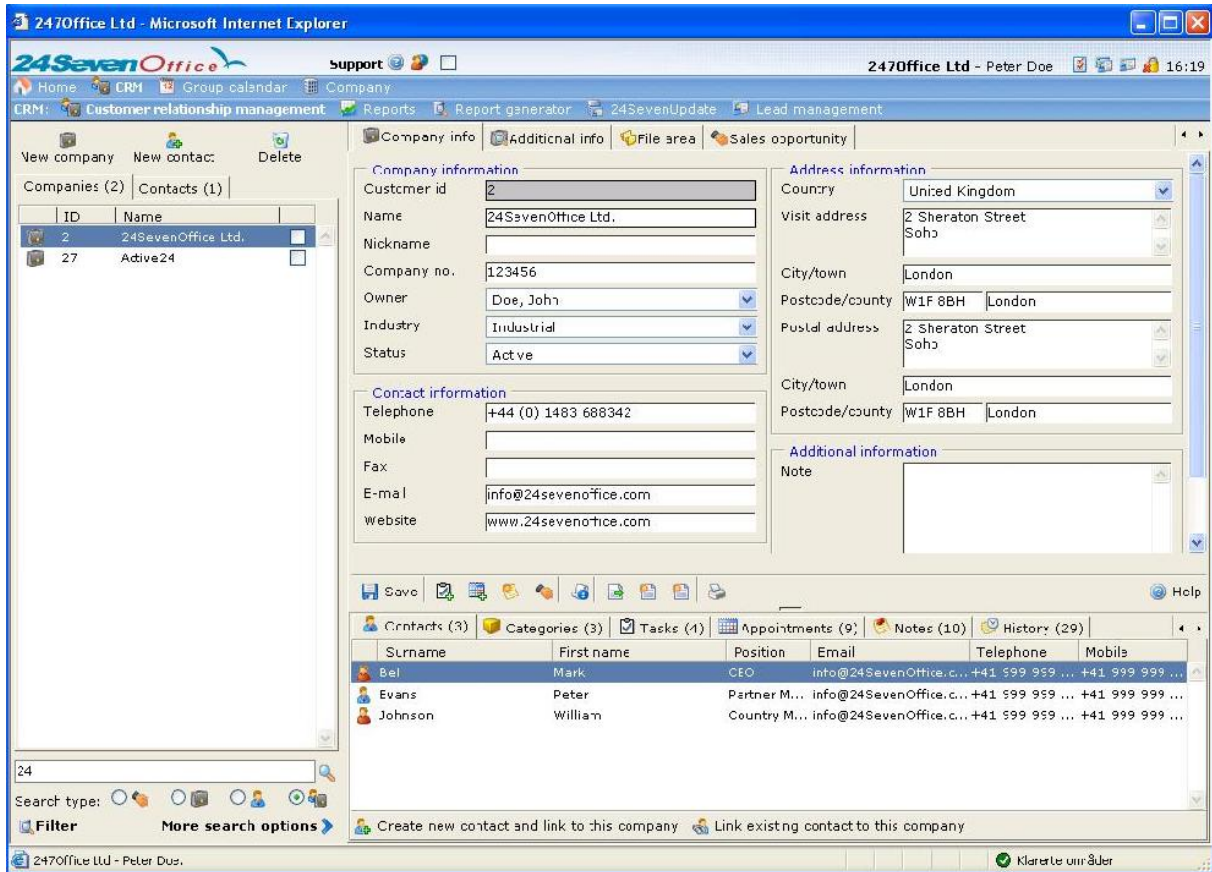
Figur 3: Google Suggest viser antall resultater mens en skriver, og foreslår en liste over vanlige søketermer.

Vi har også en rekke eksempler på fullverdige web-applikasjoner som kan erstatte tradisjonelle skrivebordsapplikasjoner. Google har med sin Google Spreadsheet lansert en gratis konkurrent til Excel, regnearkprogrammet i Microsoft Office. AjaxWrite (<http://www.ajaxlaunch.com/>) er et eksempel på en Ajax-basert, gratis Word-applikasjon som kan lagre filer i vanlig .doc-format (Figur 4). Siden alt skjer på web, er dette et alternativ for mennesker som ofte er på farten.



Figur 4: Eksempel på en fullstendig Ajax-applikasjon som gjør det mulig å jobbe med word-filer over Internett.

Det norske konsultentselskapet 24sevenOffice (<http://www.24sevenoffice.com>) selger et kundeoppfølgningssystem som i sin helhet er webbasert. Ajax er en sentral teknologi i deres løsning. Systemet har mange arkfaner og skjermbilder, og dette er helt klart å sammenlikne med en tradisjonell skrivebordsapplikasjon. Legg merke til i Figur 5 hvordan navigasjonsknappene i nettleseren er fjernet for å unngå at brukeren gjør feil.



Figur 5: En webapplikasjon for å følge opp kunder med Ajax som sentral komponent.

Norske DinSide har en kameravelger (<http://www.dinside.no/data/kameravelgeren/>) som er intuitiv i bruk for å finne ut mer om aktuelle digitalkameraer. Brukeren kan filtrere bort uaktuelle kameraer ved å kombinere ulike kriterier. Kun de kameraer som oppfyller samtlige kriterier, vises. Resten gråes ut. Dette skjer i samme øyeblikk som brukeren drar i de ulike glidekontrollene (øverst til venstre i Figur 6), og dermed får brukeren umiddelbar, visuell tilbakemelding. Dersom musepekeren holdes over et kamera, hentes derimot en oppsummering fra tilhørende kameratest og vises i en flytende boks under det aktuelle kamera. Også dette er vist i Figur 6 (tilfeldig valgt kamera).

Finn ditt nye digitalkamera:

## Kameravelgeren

Figur 6: Eksempel på noe som kunne vært Ajax, men som ikke er det. Kriteriene til venstre er med på å bestemme hvilke kameraer som vises.

Det som er interessant å merke seg her, er at denne kameravelgeren er laget i **Flash**, altså ikke i Ajax. Det er derimot fullt mulig å lage en slik kameravelger i Ajax. Ting du ser på web av rike brukergrensesnitt trenger ikke nødvendigvis å være Ajax-basert. Det er sterke og svake sider med de fleste teknologier. Stadig flere oppdager fordelene med Ajax, men det er også viktig å reflektere over utfordringene og imøtekomme disse – hvis mulig. Dersom andre teknologier egner seg bedre, bør de vurderes. En bør ikke bruke Ajax bare for teknologiens skyld (tilsvarende for alle andre teknologier).

## 1.4. Web 2.0

Web 2.0 er et begrep som ofte kommer opp sammen med Ajax. Som vi forstår av navnet henspeler Web 2.0 på noe nytt i forhold til hva vi har sett på weben tidligere. Men hva er det nye, og hva er det nytt i forhold til? Disse spørsmålene vil du kunne få ulike svar på, avhengig av hvem du spør. Her er det ingen fasitsvar, men forståelsen av begrepet bygger rundt mange trender en ser i forhold til hvilken utvikling weben tar, både på kort og lengre sikt.

Én måte å se på Web 2.0-begrepet er å betegne et nettsted med statiske websider som Web 1.0 og en web med dynamisk genererte websider som Web 1.5. Et nettsted hvor vi i tillegg har høy funksjonalitet på klientsiden, kombinert med et rikt brukergrensesnitt vil da kunne kalles Web 2.0. Hvis vi følger denne tolkningen av begrepet Web 2.0 ser vi at Ajax naturlig hører hjemme på et slikt nettsted.

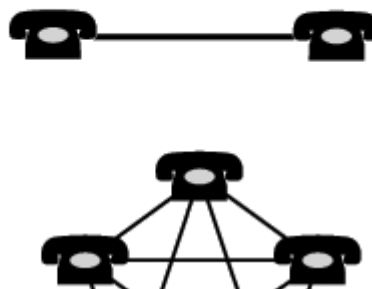
En annen måte å bruke begrepet Web 2.0 på, er å si at dette er en ny måte for å drive virksomhet på Internett. Det som er nytt er at man inviterer brukerne til å være med å delta på å skape en felles verdi, fremfor at brukerne bare er noen som benytter tjenester som ligger tilgjengelig. Vanlig netthandel med rent kjøp og salg er et eksempel på en tradisjonell virksomhet på Internett. Wikipedia, leksikonet som alle kan bidra til å fylle med innhold, er et godt eksempel på "the Web 2.0 way".

Å la brukerne kommentere produktene slik man kan på for eksempel Amazon og mange andre nettbutikker er et steg i retning Web 2.0. Hvis man drar dette enda lengre, lar man brukerne skape selve verdien på nettstedet gjennom at de selv tilbyr det som er til salgs. Dette gjelder for eksempel auksjoner som eBay og QXL, og formidlingstjenester som finn. Det virkelig interessante med dette er at verdien på tjenesten stiger med antall brukere som benytter den. En formidlingstjeneste med få brukere har liten verdi både for de som skal tilby noe og de som skal kjøpe noe.

### Metcalfe's law

From Wikipedia, the free encyclopedia

**Metcalfe's law** states that the value of a [telecommunications network](#) is proportional to the square of the number of users of the system ( $n^2$ ). First formulated by [Robert Metcalfe](#) in regard to [Ethernet](#), Metcalfe's law explains many of the [network effects](#) of communication technologies and networks such as the [Internet](#) and [World Wide Web](#).



Figur 7: Robert Metcalfe, kjent for å ha skapt Ethernet og grunnlagt selskapet 3Com, argumenterer for at verdien av et telekommunikasjonsnettverk øker tilsvarende som kvadratet av antall brukere. Kanskje kan man si det samme om tjenester på web?

Det er også viktig å merke seg at nettverkstjenester, altså tjenester der nettverket av folk er det viktige, ikke begrenser seg til tjenester hvor kjøp og salg er målet. Verdien kan være å komme i kontakt med andre, som i venne- og datingtjenester. Det kan være å oppnå en felles kunnskapsbase, som tilfelle med wikipedia. Å dele bilder på Flickr eller videosnutter på

YouTube er andre eksempler på nettverkstjenester. Mulighetene for å skape verdi gjennom å bygge nettverk er store, og vi ser at flere og flere av nettstedene som gjør suksess bygger på denne oppskriften.

Har så Ajax noe å gjøre med begrepet Web 2.0 hvis vi følger denne tolkningen? Ikke nødvendigvis, men Ajax kan være med på å utløse den fulle verdien i denne type tjenester.

For mer informasjon om Web 2.0 og de syn som eksisterer på begrepet anbefaler vi at du leser artikkelen om emnet på wikipedia, [http://wikipedia.org/wiki/Web\\_2.0](http://wikipedia.org/wiki/Web_2.0)

## 1.5. Introduksjon til fagstoffet

I dette faget vil du i hovedsak få lære om Ajax, men du vil få glimt av andre Web 2.0-teknologier underveis. Web 2.0 er en stor paraply. Et viktig element med Web 2.0 fra brukerens perspektiv, er bedre brukergrensesnitt. Det igjen gir ringvirkninger til andre Web 2.0-relaterte teknologier, som normalt har hatt en noe høyere terskel for vanlige folk å delta i. Eksempel er Wiki, sosiale nettverk, bruk av RSS-teknologi og liknende. Med et intuitivt, rikt brukergrensesnitt kan en tenke seg at Web 2.0 leveres bredere ut til folket. Vi antar at du som tar dette kurset i hovedsak er interessert i å lære programmering med Ajax.

Etter denne introduksjonsleksjonen vil vi starte med å gå kort gjennom de underliggende teknologier som er nødvendig for å få Ajax til å fungere. Ajax er kodet i JavaScript, så en basisforståelse av språket er nødvendig. Det er ikke forventet at du kan JavaScript fra før, men det er en fordel om du kjenner til grunnleggende programmeringsprinsipper generelt sett: variabler, beslutninger (if), løkker (while, for) og funksjoner.

Når du mottar en webside fra en tjener, kan sidens innhold behandles på klientsiden i ettertid uten at tjeneren har noe mer med saken å gjøre. Nettleseren registrerer alt det brukeren gjør (museklikk etc). Med vanlig DHTML er det mulig å kode slik at ting skjer som følge av disse hendelsene, og da manipuleres Document Object Model (DOM) på klientsiden. Dette har eksistert og vært i bruk lenge. JavaScript (og CSS) kan og vil ofte inngå i slik manipulasjon.

Ajax står for Asynchronous JavaScript and XML. Ajax er noe mer enn DHTML i flere henseender. Foruten en sammensetning av velkjente, velfungerende teknologier, er det den *asynkrone kommunikasjonen med tjeneren* som byr på mange nye muligheter for rike brukergrensesnitt. Det er XMLHttpRequest-objektet (i JavaScript) som byr på denne asynkrone kommunikasjonen. I flere av leksjonene vil vi se nærmere på denne kommunikasjonen, muligheter og ting å merke seg.

X-en i Ajax står for XML. Asynkron kommunikasjon fordrer at data utveksles på et eller annet format. Dette kan (men må ikke) være XML. Leksjon 4 ser nærmere på utvekslingsformater og presentasjon (Ren tekst, JSON, XML/XSLT, CSS).

Sammen med gjennomgangen av de grunnleggende teknologier, vil du få se eksempler på programkode og lage enkle løsnignger. I leksjon 5 har vi lært nok til å kunne lage en litt større, komplett Ajax-applikasjon.

En av utfordringene med Ajax er å være kreativ og se nyttige bruksområder. Det er ikke enkelt å lage gode brukergrensesnitt, og det er mange hensyn å ta. Skal Ajax brukes i det hele tatt? Gitt at dagens brukere er blitt vant til, og tenker i takt med dagens webløsninger – hvordan kan en da utforme løsninger slik at brukeren oppdager og får ta del i mulighetene i et nytt, rikere brukergrensesnitt? I Gmail kan en for eksempel holde musepekeren over navnet på en kontaktperson et lite sekund for å se bildet av vedkommende. Det kan være at brukeren

ikke oppdager slike muligheter, nettopp fordi en ikke er vant til at det kan være mulig! Balansen mellom funksjonalitet og intuitivitet er nødvendig å tenke nøye gjennom.

Mange kjenner til tjenerspråk som PHP, ASP.NET og JSP fra før. Ajax står ikke i konkurranse til disse, snarere tvert i mot. Tjener og klient må spille sammen. Ajax utfyller tjenersidespråk, siden Ajax kan tilby et bedre grensesnitt i ellers avanserte webapplikasjoner. Ajax-utvikling blir enda mer morsom når vi kan sende informasjon begge veier, og koble Ajax på allerede eksisterende webløsninger. Tenk selv på hvordan en nettbutikk som lar brukeren samle varer i en handlekurv, og til slutt registrerer kjøp – kan kombineres med Ajax slik at en kan for eksempel lage en klikk-og-dra-basert handlekurv...

Vi kommer til å introdusere ulike løsninger som kan gjøre brukergrensesnittet bedre. Rent teknisk kan det være funksjonalitet som dra-og-slipp, brytere som kan dras for å endre for eksempel størrelsen på et bilde eller antall treff fra et søk, tekstfelt som forhåndsutfylles med mulige valg, lister som reduseres i størrelse dynamisk etter hvert som en skriver inn flere bokstaver (i for eksempel søk etter ord i en ordliste) og så videre.

Ajax-løsninger vil generelt sett være mye større enn vanlige DHTML-løsninger. Det vil være mye JavaScript-kode. Fra basiskunnskap i HTML vet du sikkert at HTML-kode fort kan bli rotete, et sammensurium av kode som utvikles over tid og blir vanskelig å vedlikeholde. Du vil også se at enkelte elementer eller kodesnutter du lager går igjen i ulike løsninger. Design patterns handler om hvordan du kan sikre orden i koden, organisere på en fornuftig måte og sikre gjenbruk. Det fins også rammeverk og tredjeparts-biblioteker som kan hjelpe til slik at en ikke trenger å finne opp hjulet på nytt hver gang en skal bruke ulike kontrollere og realisere visse typer funksjonalitet i grensesnittet. Det er bevisst at disse temaene kommer mot slutten av kurset slik at de grunnleggende ferdighetene med Ajax først innøves – ting som uansett må læres.

Planlegging av løsninger og dokumentasjon er oppgaver som ofte blir neglisjerte. Det kan være mye tid å spare på å planlegge løsningene før programmeringen starter, og dokumentasjon kan bli nyttig i videre utvikling i fremtiden. Kvalitet og god planlegging/dokumentasjon hører tett sammen. Mot slutten av dette kurset kommer vi inn på noen slike tema.

Til slutt ser vi litt på sikkerhet og noen avanserte tema. Sikkerhet og webutvikling er alltid et viktig tema, også i Ajax-sammenheng.

Vi håper du gleder deg til kurset, og er klar for å lære mer om spennende webbaserte teknologier.