

# Database-baseret Web-publicering

ved

## IT-højskolen i København

### Database-baseret WEB Publicering, efterår 2001

#### Forelæsning 1, tirsdag den 28. august 2001

- Generel Kursusinformation
- Ugentlige øvelser og eksamen
- Kursusindhold og motivation
- WEB Publicering
- WEB-baserede Services
- Udvikling af Statiske Sider
- Versionskontrol
- Detaljéret indhold
- HTML på 21 minutter
- Nogle WEB-design Regler
- Eksempel på SQL
- CGI-scripts og Appletter m.m.
- Hvad er et program? - og lidt om programmeringssprog
- Eksempler på TCL

### Generel Kursusinformation

- **Kursusformål:** Lær at anvende avancerede web-teknikker til at udvikle realistiske, brugbare web-sites, der involverer udveksling af information med en database.
- **Lærer:** Niels Hallenberg (nh@it-c.dk, lokale 2.38)
- **Hjælperlærere:** Kenneth Riis (kenneth@it-c.dk) og Frank Bjergø (bjergoe@it-c.dk)
- **Kursets hjemmeside** er <http://www.it-c.dk/courses/W2/E2001/>
- **Kursusforløb,** se forelæsningsplanen: <http://www.it-c.dk/courses/W2/E2001/>
- Hjemmesiden indeholder udlånte materialer, sidste nyt, ...
- Forelæsningsplanen indeholder forelæsningsplancherne
- Forelæsningsplancherne udkommer (senest) dagen før forelæsningen
- **I skal holde øje med hjemmesiden hele semestret igennem**
- **Hvornår:** forelæsning tirsdag formiddag. Øvelser enten tirsdag eftermiddag eller onsdag formiddag.
- Stil endelig spørgsmål under forelæsningerne

### I får brug for

- **Lærebog:** <http://www.photo.net/wrt/ch ebook/>, der enten skrives ud fra nettet, bestilles på nettet eller købes i samfundslitteratur. Pris ca. 500 kroner.
- **Noter,** der udskrives fra hjemmesiden
  - *A Smooth Introduction to Tcl for Web Nerds*, Martin Elsmann.
  - *Tcl for Web Nerds*, Hal Abelson, Philip Greenspun, and Lydia Sandon.
  - *SQL for Web Nerds. Tutorial for the SQL language:* links into online Oracle docs for completeness. Philip Greenspun.
  - *SlideExtractor*, the engineering behind a tool for extracting slides from WimpyPoint, a web-based slide-structor. Niels Hallenberg.
- **Relaterede Noter**
  - *HTML overview*, Peter Sestoft.
  - *En kort introduktion til skrivning af HTML-dokumenter*, Mads Tøfte.
  - *HTML-codes for unusual characters and symbols*, MIT/W3C.
  - *Four Part Article about AOL server, tcl, and Oracle*. In *LinuxWorld*, Philip Greenspun.
  - og meget andet - se kursets hjemmeside.
- **Konto** på IT-højskolens maskiner og på kursets web-server.

## Ugentligt Øvelser

### Forudsætninger:

*Grundlæggende Web-design eller tilsvarende forudsætninger. Derudover skal du være motiveret for at lære at programmere.*

- Man lærer kun for alvor ved at prøve selv.
- Læsevejledning: Læs lærebog og noter forud for hver forelæsnings. **Løs opgaver** i god tid. Løs også de nemme ekstra opgaver hvis nødvendigt!
- Øvelser: 4 timer pr. uge pr. studerende, fra tirsdag den 4. september.
- Øvelserne foregår på IT-højskolen, Gentlevej 67
- Øvelsesidspunkter: tirsdag 12.30–16.30.
- Øvelseshold oprettes inden tirsdag den 4. september.
- Ugentlige opgaver til elektronisk aflevering. Eksempel:
  - Opgaver stilles tirsdag d. 28/8 på løbesedlen
  - I forbereder opgaverne derhjemme
  - Opgaver regnes ved øvelserne ugen efter, f.eks. tirsdag d. 4/9
  - Besvarelserne skal afleveres tirsdag d. 11/9
  - Vi giver de rettede opgaver tilbage tirsdag d. 18/9

## Kursusindhold og motivation

Web-publicering kræver generalister:

skrive, foto, publicering, system administration, databaser, bruger interface, programmering, design, . . . . .

Udgangspunkt: Statistiske sites (HTML)

Hvordan kan man konstruere mere interessante sites:

- sites som er programmer (dynamisk HTML)
  - beregning af skat
  - Bill Gates personal wealth clock
- sites som er databaser (dynamisk HTML med database tilgang)
  - RemindMe
  - IT-C's nyhedsjeneste
  - SpecWeb

Populære og brugbare sites kan bygges med få midler!

## Eksamen, Pensum og Øvelsesindhold

- Ud af semestrets 11 løbesedler vil de 8 bedste indgå med 50% af den samlede karakter.
- Skriftlig eksamen indgår med 50% af den samlede karakter.
- **Skriftlig eksamen og løbesedler skal beståes hver for sig for at bestå kurset**
- Løbesedlerne afleveres elektronisk - mere information herom på den første løbeseddel.
- Rettelserne gøres synlige i CourseGrader: <http://hug.itu.dk:8002/vu/index.tcl>
- Løbesedlerne løses individuelt:
  - Løbesedlerne dækker:
    - HTML
    - Simpel programmering med TCL
    - Simpel databaseprogrammering med SQL
    - Web-sites konstrueret med HTML, TCL og SQL
- Pensum vil være lærebog, og noterne om TCL og SQL, løbesedler, ekstra opgaver samt forelæsningsplanerne. Det forventes, at man behersker HTML. Justering og præcisering af pensum vil ske gennem semestret, og den sidste løbeseddel vil indeholde en præcis pensumliste.

## WEB Publicering

Web-publicering er ofte kendetegnet ved et magasin agtigt indhold, f.eks. produktkataloger, avisjenester m.m.

Web-services er kendetegnet ved at løse et problem for en bruger, f.eks. huskeliste, kommentarservice, bogkøb, billetbestilling m.m.

Start med at spørge hvad en bruger kan være interesseret i!

Hvad kan du gøre for at berige en bruger?

Vær parat til at interagere med brugerne, dvs. svare på kommentarer m.m., (eng. *Multiple Views*).

Undgå

- user-modeling, entry tunnel, exit tunnel
- flashy langsom grafik

Understøt flere syn på indholdet.

Tilfald brugerkommentarer med f.eks. ArsDigita's gratis services.

## Web-baserede Services

Fra desktop applikationer til web-baserede programmer

- WimpyPoint
  - RemindMe
  - Kalendersystem
  - Tekstbehandlingsversionskontrol
  - CourseGrader
  - Kursusevaluering
  - Kursusbog / tilmeldingssystem
- Fejl i web-baserede programmer kræver og tillader rettelser hurtigt!

## Versionskontrol

### Problem

- Joe henter Version A af et dokument kl. 09.00 fra web-sitet og bruger en dag på at editere det.
- Mary henter Version A kl. 12.00 og retter en typo. Mary skriver dokumentet tilbage til serveren kl. 12.10 (Version B).
- Joe skriver sit dokument tilbage til serveren kl. 17.00 (Version C).

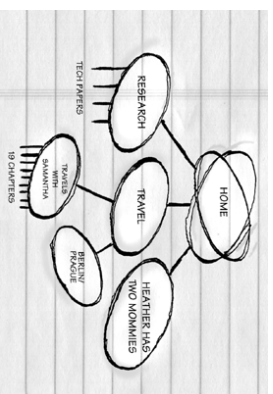
Mary's rettelser er glemt! Version C indeholder ikke rettelserne i Version B!

Løsninger: "låsnings" af filer med emails, Emacs, RCS, CVS, .....

CVS: <http://www.cvshome.org/>

## Udvikling af Statistiske Sider

- tegn en site-oversigt (eng. *site-map*)
  - saml og struktur indhold: giv navne til filer.  
Omvælg hvorledes sekundære filer såsom billeder m.m. skal organiseres.
  - konstruer et kun-tekst site
  - ansæt en grafisk designer – og instruer ham!
  - konstruer en vedligeholdelsesplan. Hvorledes opdateres indhold?
- Kan man udelegere dette ansvar til dem, der skriver indholdet, eller skal man have en central hjemmeside-afdeling.
- lav periodiske brugertests.



Billede: <http://www.photo.net/wtr/thebook/mapkin/14.1.gif>

## Detaljeret indhold

- HTML (Hyper Text Markup Language)
  - TCL (Tool Command Language)
  - SQL (Structured Query Language)
  - Hvorfor lære programmering?
    - Fordi man kun på den måde virkelig forstår muligheder og begrænsninger i udvikling af dynamiske web-sites
    - Fordi det er nødvendigt i de fleste avancerede web-sites
    - Fordi man bedre kan forestille sig nye typer af web-sites
    - Fordi man bedre forstår teknologiens muligheder og begrænsninger
- Fordi det er sjovt!
- ### Hvorfor lære TCL?
- TCL er velegnet til udvikling af dynamiske web-sites (ligesom Java, ASP, PHP3, Standard ML, ...)
  - TCL er et lille sprog med relativt få programkonstruktioner.
  - TCL er god til behandling af strenge (og derfor web-programmering)
  - `tclsh` gør det let at afprøve programmer
  - TCL er fuldt integreret med en web-server (AOL-server)

## Specielle programmeringssprog: HTML og SQL

**HTML:** Beskriv hypertextens struktur og (til dels) layout

Eksempel (hello.html):

```
<html>
<head><title>
Hello
</title></head>
<body>
Hello
</body>
</html>
```

Browsere er normalt ikke pedantiske overfor korrekt HTML – men derfor skal I skrive korrekt HTML alligevel!

**SQL:** Sender data frem og tilbage mellem et program og en database.

```
select navn, adr from navne order by navn;
insert into navne (navn, adr) values ('Iris', 'Gallevej 435');
```

Der findes et uaf af SQL-dialekter - men det er normalt trivielt at forstå forskellene.

## Generel Side Layout (lbl-ekstra.html)

Opdel gerne en side i logiske dele såsom **top**, **krop** og **bund**.

Dette vil gøre programmeringen lettere senerehen.

Du bør kende følgende tags i HTML:

- overskrifter, dvs. <h1>, </h1>, ..., <h4>, </h4>.
- skillelinier såsom <hr>
- afsnit og ny linie såsom <p>, </p> og <br> og </br>
- quotes, <blockquote> og </blockquote>.
- centring <center> og </center>
- fed <b> og </b>
- kursiv <i> og </i>
- understregning <u> og </u>
- ordnede <ol>, </ol> og uordnede <ul>, </ul> lister samt <li>.

## HTML på 21 minutter

Et legalt HTML dokument (legal1.html):

```
<html>
<head>
  <title>Mit Size</title>
</head>
<body bgcolor=white>
  <h2>Mit Size</h2>
  Se siden for
  <a href="http://www.it-c.dk/courses/W2/E2001/">Database-baseret
  Web-publicering</a>
</body>
</html>
```

Det kan være svært at sætte HTML kode pænt op – men prøv at være konsistent!

PT skriver vi HTML i hånden – senere skal vi lave programmer som genererer HTML kode som resultat.

**Skriv HTML koden selv!**

## Hyperreferencer (lbl-ekstra.html)

- hyperreferencer <a href="link">navn</a>
- lokale hyperreferencer med etiket <a name="etiket"> og reference til etiket <a href="index.html#etiket">navn</a>
- emailreferencer <a href="mailto:nh@it-c.dk">nh@it-c</a>
- tabeller med <table>, </table>, <tr>, </tr>, <th>, </th>, <td> og </td> samt de mest almindelige attributter
- billeder <img>
- farver f.eks. bgcolor

### Find fem fejl

i et HTML dokument uden konsistent indryk

```
<html><head><title>Hello</head>Hello<center>great page<p>  
this is in <b>bold<i> and italics</html>
```

eller det samme dokument med indrykning

```
<html>  
<head>  
<title>Hello</title>  
</head>  
Hello  
<center>  
great page  
<p>  
this is in <b>bold<i> and italics  
</html>
```

Netscape viser dette dokument uden fejl!

### Mere HTML dokumentation

Der er masser af henvisninger på kursets hjemmeside til små tekster om HTML:

- Peter Sestofts HTML-oversigt
- Mads Toftes Introduktion til skrivning af HTML dokumenter
- Den legendariske A Beginner's Guide to HTML
- HTML-koder for usædvanlige tegn og symboler
- Per B. Rasmussens Fravekoder i HTML

### Simpelt er Godt

Problemer med fancy-tags, såsom FONT:

- Gamle browsere ignorerer dem
- Nye browsere ignorerer dem
- Når du udskifter den grafiske designer skal du ændre 10.000 html-sider!
- Istedet: brug cascading style sheets.

Modern-brugere kan ikke lide for megen grafik!

### Flere HTML Regler

- et minimum af JavaScript
- ingen store tabeller, f.eks. et helt dokument.
- lad være med at ændre farver på links
- så lidt formatering som muligt
- sørg for at din side kan udskrives på printer. F.eks. vil Netscape udskrive hvid tekst på blå baggrund som hvid på hvid!
- tænk på din brugers skærm som en krap ressource.
- ingen "frames" – fylder på skærmen og er svære at håndtere med programmer.
- Husk at give dine HTML sider en titel, bl.a. til bogmærke.

En målsætning kunne være, at både Netscape 1.1 (eller 2.0) samt Explorer kan vise din side.

Hvad med WebTV, Palm Pilots, mobil telefoner, ...

### Flere Web-design Regler

- Informationen skal være brugergrænsefladen: ved klik på informativ ord, bringes man til uddybende information (undgå "her"-links)
- Giv brugere et bredt, flat overblik af information, istedet for et sekventielt informationsforløb
- Organiser dit site efter forventede brugerinteresser, istedet for efter den interne organisation.
- Hvorfor bruge ikoner til at navigere, når ord er mere sigende og fylder mindre?
- Længere dokumenter fremfor brede.
- Husk at linke fra dine sider med indhold tilbage til en hovedside (indeks), bl.a. p.g.a. søgemaskiner.
- Udstyr din web-site med en fuldtækt søgemaskine – og sørg for, at den holder statistik med de ord, som den ikke kan finde svar på.
- Udstyr dine sider med kontaktinformation
- En vertikal menu i siden optager mere plads på skærmen fremfor en horizontal menu

## Eksempel på SQL

Givet en tabel student med nogle studentterdata

efternavn	fornavn	studienummer
Olesen	Peter	L2143
Hansen	Erika	Jø0007
Funder	Ulrik	Hg0014

Vi kan forespørg på fornavn og efternavn eksempelvis – sorteret stigende:

```
select fornavn, efternavn from student order by efternavn;
```

Vi kan indsætte en ny række:

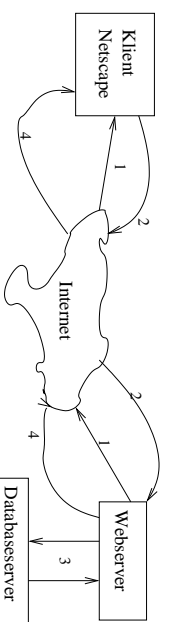
```
insert into student (efternavn, fornavn, studienummer)
values ('Fong', 'Joe', 'JF0032');
```

Vi kan slette en række:

```
delete from student where efternavn = 'Fong';
```

... og meget mere

## Brug af CGI-scripts (TCL der udvikles på web-server)



1. Klient henter HTML side fra webserver
2. Klient indtaster data i HTML side, og sender data tilbage til webserver
3. Webserver udvikler et TCL program, som behandler data fra klienten, bl.a. opdaterer en database.
4. Webserver sender en HTML side tilbage til klient.

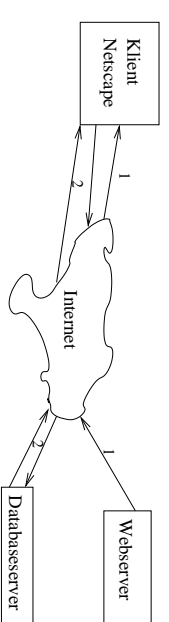
Hvis HTML koden er simpel, og bl.a. ikke indeholder for megen JavaScript, kan den vises i de fleste browsere.

Loader normalt hurtigt, men brugergrænsefladen er ikke som et normalt desktop program.

I dette kursus kører web-server (AOLserver) og database-server (Oracle) på hugg .it-c.dk.

I har alle jeres egen AOLserver og Oracle server på hugg!

## Brug af Appletter til opdatering af databaser



1. Klient henter HTML side fra webserver med en applet
  2. Klient benytter appletten som et alrn. program. Deriblandt sender forespørgsler og opdateringer direkte til en database.
- Appletter virker kun godt på nyere browsere, og tager lang tid at downloade.
- Tilgængæld er de mere interaktive.

Vi diskuterer ikke appletter og andre lignende teknologier (Flash, DoNet m.m.) i dette kursus.

## Hvad er et program?

Et program er interne instrukser til en datamaskine.

Interne instrukser kan medføre eksterne, synlige resultater.

F.eks. beregning og visning af en saldo, tegning af en graf, opdatering af en database, generering af HTML, ...

Afviklingen af et program er 100% mekanisk.

Programmer er generelle; kan løse uendelig mange problemer af en bestemt slags.

Programmer er stupide; kan ikke improvisere eller 'forstå'.

Datamaskiner er hurtige; derfor kan programmer alligevel virke smarte.

### Hvordan udføres et program på en datamaskine?

1. Program indlæses fra harddisk til det interne lager.
2. Den første/næste instruktion i programmet hentes ind i CPU'en. CPU'en udfører instruktionen.
3. CPU'en gemmer måske resultatet af en beregning i det interne lager.
4. CPU'en udskriver måske resultatet af en beregning på skærmen.
5. CPU'en returnerer måske HTML til en browser.
6. CPU'en modtager måske inddata fra en bruger (browser).
7. CPU'en modtager måske inddata fra en database.
8. ...

### Et simpelt TCL program (name.tcl)

```
set firstring Martin
set lastname Elsmann
set age 30
set email mael@itu.dk
puts "My name is $firstring $lastname and my email address is $email."
```

Programmet består af

- en variabel firstring med værdien Martin.
  - en variabel lastname med værdien Elsmann
  - en variabel age med værdien 30
  - en variabel email med værdien mael@itu.dk
  - kommandoen set anvendt fire gange, som *tildelel* en variabel en værdi
  - kommandoen puts, som udskriver en streng (hvor variabelene er nævnt) på skærmen.
- Programmet afvikles fra første linie og ned efter.

### Hvad er et programmeringssprog?

Et programmeringssprog er en notation for instrukser til datamaten, dvs. for programmer.

Man skal være meget omhyggelig med programmeringssprog.

En computer er normalt meget pændansk.

Dette er specielt til gene og frustration for begyndere.

Eksempel: Korrekt TCL program:

```
puts "I'm alive!"
```

Helt forkert TCL program:

```
Put "Im alive!"
```

Det kræver øvelse at spotte fejl!

### Redigering og udførelse af et TCL program

#### Linux

```
[23:15-S1011]# tclsh
% source name.tcl
My name is Martin Elsmann and my email address is mael@itu.dk.
%
```

Starter TCL fortolkeren tclsh og indlæser samt kører programmet med source.

Alternativt starter man fortolkeren og indtaster kommandoerne en linie af gangen.

#### Windows

I stedet for tclsh hedder TCL fortolkeren wish.

I kan installere TCL fortolkere hjemme, både under Windows og Linux.

### Grundbegreber: Variable og tildeling

En variabel/kan indeholde en værdi, f.eks. tallet 30 eller strengen "Hello"

En variabel svarer til et sted i datamaterns lager.

Du navngiver selv dine variable.

Du skaber en variabel med kommandoen set, som også *tildeler* variabelen en værdi

```
set firstname "Martin"
```

Når du skaber en variabel, så afsættes plads til variabelen i datamaterns lager.

Du kan overskrive indholdet af en variabel med en ny værdi:

```
set firstname "Hans Otto"
```

Man kan tegne datamaterns lager og se hvad der sker når et program kører.

### Et mere avanceret TCL program der genererer HTML (Loop .tcl)

Programmet, som I på nuværende tidspunkt ikke forventes at forstå:

```
proc mul_tab {n} {
  puts "<html><body><table>" ;
  puts "<tr><th>n</th><th>n*3</th><th>n*n</th></tr>" ;
  for {set i 0} {$i<$n} {incr i} {
    puts "<tr><td>$i</td><td>[expr $i * 3]</td>
      <td>[expr $i * $i]</td></tr>"
  }
  puts "</table></body></html>"
}
```

og uddata:

```
23:46-S101]# tclsh
% source loop.tcl
% mul_tab 3;
<html><body><table><tr><th>n</th><th>n*3</th><th>n*n</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>3</td><td>1</td></tr>
<tr><td>2</td><td>6</td><td>4</td></tr>
</table></body></html>
```

### Et simpelt TCL program der genererer HTML (hello.tcl)

```
puts "<html>
<head>
<title>
Hello
</title>
</head>
<body>
Hello
</body>
</html>"
```

### Denne uges Løbeseddel

- Se <http://www.it-c.dk/courses/W2/E2001/Lb/1b1.html>
- Ved problemer anvend Q&A forum.
- Aflevering - du skal benytte CourseGrader, og du skal kopiere din besvarelse til `hug.it-c.dk`.