

## Databasestøttet Webpublicering, forår 2002

### Forelæsning 4

#### Web-programmering, tilfældige tal, og tilstand

- Formulærer
- Tekstområder, indtastningsfelter og menuvalg
- Generering af tilfældige tal på web-server
- Web-programmer med tilstand
- Afsending af email fra web-server

#### Tekstområder

Brugeren kan indtaste vilkårlige tekster i et tekstområde.

Du kan angive tekstområdets størrelse, dvs. antal tegn per linje og antal linjer.

```
<textarea name=navn rows=r cols=c>
```

*tekst*

```
</textarea>
```

hvor

- *navn* er navnet på tekstområdet. Navnet anvendes i Tcl-programmet.
- *r* er antallet af rækker i tekstområdet. Hvis brugeren skriver en tekst der er større end tekstområdet, så vil browseren vise en skylder.
- *c* er antallet af tegn per række i tekstområdet. Browseren viser en skylder, hvis brugeren laver linjer på mere end *c* tegn.
- *tekst* er den tekst som vises i tekstområdet. Brugeren kan slette denne tekst.

Eksempel på et tekstområde til kommentarer:

```
<textarea name=comments rows=7 cols=50>
Skriv dine kommentarer her.
</textarea>
```

#### Formulærer

En *HTML-formular* benyttes ved indtastning af oplysninger.

Oplysningerne sendes via nettet til et Tcl program på web-serveren.

Tcl programmet behandler oplysningerne, gemmer dem i en database og returnerer en ny HTML side.

Syntaks for en simpel formular:

```
<form action=url method=post>
```

*html-kode*

```
</form>
```

hvor *url* identificerer det Tcl program som skal udføres når formen "submits" og *html-kode* er almindelig HTML kode inklusive de indtastefelter som formularen indeholder.

Der er tre grupper af indtastefelter:

- tekstområder (`<textarea>`)
- indtastningsfelter (`<input>`)
- menuvalg (`<select>`)

#### Eksempel på tekstområde (*Kursiviv.html*)

```
<html>
<head>
<title>Kursiviv</title>
</head>
<body>
<h2>Eksempel på et tekstområde.</h2>
<form action="kursiv.tcl" method=post>
Den tekst du skriver nedenfor bliver returneret i kursiv.<p>
<textarea name=tekst rows=7 cols=50>
Denne tekst bliver sat i kursiv.
</textarea><p>
<input type=submit value="Kursiver">
</form>
</body>
</html>
```

Se <http://hug.it.edu:8002/F2002/lec4/kursiviv.html>

### Eksempel på tekstområde - fortsat (kursiv i .tcl)

```
set_form_variables
ns_return 200 text/html "
<html>
<head>
<title>Kursiver Service</title>
</head>
<body>
Teksten som skal vises i kursiv:<p>
<blockquote>
<pre>
<i>$tekst</i>
</pre>
</blockquote>
</body>
</html>"
```

### Indtastningsfelter fortsat

Følgende typer kan også anvendes:

- **radio**: En eller flere knapper vises, men kun en knap kan være valgt på samme tid. Anvendes ved valg mellem gensidigt udelukkende muligheder. Attributen `checked` markerer den knap der som udgangspunkt vil være klikket.  
`<input type=radio name=sex value="male">`  
`<input type=radio name=sex value="women" checked>`
- **reset**: En knap som nulstiller alle indtastningsfelter.  
`<input type=reset value="Forfra">`
- **submit**: En knap som anvendes til at sende formularens data til Tcl programmet angivet i `action`-attributen.  
`<input type=submit value="Send">`
- **hidden**: Skulle værddier.  
`<input type=hidden name=id value="42">`

### Indtastningsfelter

Indtastningsfelter findes i flere varianter angivet ved attributen `type`:

```
<input type=type name=navn value="vælg">
```

hvor

- `navn` er navnet på indtastningsfeltet. Navnet benyttes i Tcl programmet.
- `vælg` er valgfri og angiver den værdi som feltet som udgangspunkt antager.
- `type` angiver hvorledes indtastningsfeltet vises. Du kan vælge mellem følgende typer:
  - `text`: En enkelt tekstlinje kan indtastes, f.eks.  
`<input type=text name=by>`
  - `password`: virker som `text`, men indtastningen skjules, f.eks.  
`<input type=password name=kodeord>`
  - `checkbox`: en knap der kan klikkes. Der kan være en eller flere knapper og en eller flere knapper kan vælges, f.eks.:  
`<input type=checkbox name=cheese value="cheese">`  
`<input type=checkbox name=olives value="olives">`

### Eksempler på indtastningsfelter (typesetting .html, del 1)

```
<html>
<head>
<title>Typesetting</title>
</head>
<body>
<h2>Eksempel på formattering af en tekstlinje.</h2>
<form action="typesetting.tcl" method=post>
Den tekst du skriver her
<input type=text name=tekst>
bliver formatteret således:<p>
<blockquote>
kursiv: <input type=checkbox name=font value="i"><p>
fed: <input type=checkbox name=font value="b"><p>
understrening: <input type=checkbox name=font value="u"><p>
</blockquote>
```

Du kan vælge mellem grøn og gul baggrund:  
Grøn <input type=radio name=farve value="Green">  
Gul: <input type=radio name=farve value="Yellow" checked><p>

### Eksempler på indtastingsfelter (typesetting .html, del 2)

Du skal skrive hemmeligt her <input type=password name=kodeord>, for at det virker. <p>

```
<input type=hidden name=skjult_vaerdi
value="Denne tekst var skjult i formularen">
<input type=reset value="Forfra">
<input type=submit value="Formatter tekst">
</form>
</body>
</html>
```

Se <http://hug.it.edu:8002/F2002/lec4/typesetting.html>

### Eksempler på indtastingsfelter (typesetting .tcl, del 1)

```
set_form_variables
proc home_page { title body } { ... }
```

### Procedureren set\_form\_variables

Procedureren erklærer en variabel svarende til hvert inddata-element i HTML dokumentet.

Den følger dog, hvis der ikke overføres nogen data.

Fejl kan undgås ved at give argumentet 0 til procedureren:

```
set_form_variables 0
```

### Check om en variabel findes med "info exists"

Med kommandoen info og første argument exists kan vi checke om en variabel findes:

```
% info exists x
0
% set x 42
42
% info exists x
1
```

Husk, at 0 svarer til falsk og 1 til sand.

### Eksempler på indtastingsfelter (typesetting .tcl, del 2)

```
if { [string compare $kodeord "hemmeligt"] != 0 } {
ns_return 200 text/html [home_page "Typesetting Service" "Kodeordet
$kodeord er forkert" ]
} else {
ns_return 200 text/html [home_page "Typesetting Service" "
<table bgcolor=$farve>
<tr><td>
Teksten $tekst ser således ud med formattering <font>$tekst</font><p>
Der var også en skult værdi i formularen: $skjult_vaerdi
</td></tr>
</table>
" ]
}
```

Hvad sker der hvis vi ikke vælger en formattering (kursiv, fed eller understregning)?

- Variablen font eksisterer kun, hvis brugeren vælger en formattering.
- Hvorledes kan vi checke, at brugeren har indtastet en værdi?
- Vi kan checke atvariablen, f.eks. font eksisterer efter kaldet af procedureren set\_form\_variables.

Hvad sker der hvis vi vælger mere end en formattering?

### Eksempel med med info exists (info\_exists .html og info\_exists .tcl)

```
info_exists.html:
<html>
<body> <h2>Vælg mellem A og B</h2>
<form action="info_exists.tcl" method=post>
valg A: <input type=checkbox name=valg value="Du har valgt A"><p>
valg B: <input type=checkbox name=valg value="Du har valgt B"><p>
<input type=submit>
</form>
</body>
</html>

info_exists.tcl:
set_form_variables 0
proc home_page { title body } { ... }
if { [info exists valg] } {
ns_return 200 text/html [home_page "Eksempel med Info Exists" "$valg." ]
} else {
ns_return 200 text/html [home_page "Eksempel med Info Exists" "Du
har hverken valgt A eller B." ]
}
```

Se [http://hug.it.edu:8002/F2002/lec4/info\\_exists.html](http://hug.it.edu:8002/F2002/lec4/info_exists.html)

### Eksempel med med info exists (typesetting2.tcl)

```
set_form_variables 0
proc home_page { title body } { ... }

if { [info exists font] &&
     [info exists tekst] &&
     [info exists farve] &&
     [info exists skjult_vaerd] &&
     [info exists kodeord] } {
    if { [string compare $kodeord "hemmeligt"] != 0 } {
        ns_return 200 text/html [home_page "Typesetting Service" "kodeordet
        $kodeord er forkert" ]
    } else {
        ns_return 200 text/html [home_page "Typesetting Service" "
        <table bgcolor=$farve>
        <tr><td>
        Teksten $tekst ser således ud med formattering <$font>$tekst</$font><p>
        Der var også en skult værdi i formularen: $skjult_vaerd
        </td></tr>
        </table>
        " ]
    } else {
        ns_return 200 text/html [home_page "Typesetting Service" "Du mangler
        at indtaste en eller flere værdier." ]
    }
}

Se http://hug.it.edu:8002/F2002/lec4/typesetting2.html
```

### Menuvalg

Syntaks for en simpel valgliste:

```
<select name=name multiple size=size>
<option selected value=vælg1>tek1
...
<option value=vælgn>texn
</select>
```

hvor

- *name* er navnet på inddatafeltet. Navnet benyttes i Tcl programmet.
- *multiple* er valgfri, og indikerer, at man kan vælge en eller flere elementer fra listen. Hvis *multiple* ikke er angivet kan man kun vælge et element fra listen.
- *size* angiver antal elementer som vises på skærmen.
- *option* tags angiver elementerne i valglisten
- værdien *vælg*, returneres til serveren, hvis element *i* vælges.
- for element *i* vises teksten *tek<sub>i</sub>* i valglisten.
- *selected* er valgfri og angiver, at elementet som udgangspunkt er valgt. *selected* kan være angivet for flere elementer.

### Eksempel med Menuvalg (typesetting3.html)

```
<html>
...
...
<form action="typesetting3.tcl" method=post>
Den tekst du skriver her <input type=text name=tekst>
bliver formateret således:<p>
<blockquote>
<select name=font multiple size=3>
<option selected value="i">kursiv
<option value="p">fed
<option value="u">understregetning
</select>
</blockquote>
Du kan vælge mellem grøn og gul baggrund:
<blockquote>
<select name=farve size=1>
<option value="Green">Grøn
<option value="Yellow">Gul
</select>
</blockquote>
<input type=submit value="Formater tekst">
</Form>
...
</html>
```

Se http://hug.it.edu:8002/F2002/lec4/typesetting3.html

### Eksempel med Menuvalg, fortsat (typesetting3.tcl)

```
set_form_variables 0
proc home_page { title body } { ... }

if { [info exists font] &&
     [info exists tekst] &&
     [info exists farve] } {
    ns_return 200 text/html [home_page "Typesetting Service" "
    <table bgcolor=$farve>
    <tr><td>
    Teksten $tekst ser således ud med formattering <$font>$tekst</$font><p>
    </td></tr>
    </table>
    " ]
} else {
    ns_return 200 text/html [home_page "Typesetting Service" "Du mangler
    at indtaste en eller flere værdier." ]
}
```

Hvad hvis bruger vælger flere font-indsstillinger?

Samme problem som i eksemplet typesetting.tcl.

Vi har brug for lister! Vi gennemgår lister i næste uge.

### Web-programmer med tilstand (tae1ler.tcl)

Skjulte formvaridder kan bruges til at implementere tilstand i et web-program.

```
proc home_page { title body } { ... }

proc form_up { t } {
    return "<center><h2>Web-tae1ler $t</h2>
    <form method=post action=tae1ler.tcl>
    <input type=hidden name=tal value=\"\$t\">
    <input type=submit value=UP>
    </form>
    </center>"
}

# saet formvariabel 'tal'; fejler ikke, hvis tal ikke findes.
set_form_variabels 0

if {[info exists tal]} {
    incr tal
} else {
    set tal 0
}

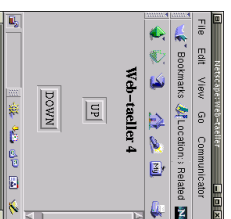
ns_return 200 text/html [home_page "Web-tae1ler" [form_up $tal]]

Se http://hug.it.edu:8002/F2002/lec4/tae1ler.tcl
```

### Eksempel tae1ler fortsat (tae1ler2.tcl)

Lad os udvide tae1ler.tcl med en knap Down som tæller den skjulte værdi ned.  
Vi har brug for:

- en ekstra knap af type submit
- at de to submit-knapper har forskellige værdier: UP og DOWN
- en ekstra if-kommando som kan finde frem til hvilken af de to knapper der er klikket.



Vi anvender string compare til at finde frem til hvilken knap der er klikket.

```
if {[info exists tal]} {
    if {[info exists submit] && [string compare $submit "UP"] == 0} {
        incr tal
    } else {
        incr tal -1
    }
} else {
    set tal 0
}
```

Udover ovenstående har vi også indsat følgende i form-koden:

```
<input type=submit name=submit value=DOWN>
```

### Generering af pseudo-tilfældige tal på web-serveren

En maskingenereret pseudo-tilfældigt talfølge synes tilfældig: 91529, 15343, 42, 68836, ...

Men faktisk er hvert tal helt bestemt af det foregående. Det første tal kaldes søkvensens frø (engelsk: seed).

Pseudo-tilfældige tal

- kan bruges til at simulere komplicerede (naturlige) processer,
- eller noget så simpelt som at kaste en terning.

#### Kommandoen randomRange

Kommandoen randomRange  $N$  genererer et ligefordelt tilfældigt tal mellem 0 og  $N - 1$ .

"Ligefordelt" betyder at alle resultater er lige sandsynlige.

Eksempler:

- randomRange 100 genererer et tilfældigt tal mellem 0 og 99
- expr [randomRange 6] + 1 genererer et tilfældigt tal mellem 1 og 6, dvs. kaster en terning.
- Aid to Evaluating Your Accomplishments  
(<http://www.photo.net/philip/careers/Four-random-people.tcl>)

### Lad os kaste nogle terninger. (kast\_terning.tcl)

Opgave: lav en procedure kast\_terning, som returnerer et tilfældigt tal mellem 1 og 6.

```
proc kast_terning { } {
    return [ ]
}
```

Opgave: lav et kald til ns\_return, således at du returnerer en side med resultatet af dit terningekast.

Du skal anvende proceduren home\_page.

```
ns_return 200 text/html [home_page "Kaste Terning Service" "
    Resultat af at kaste din terning: [ ]"]
```

Se [http://hug.it.edu:8002/F2002/lec4/kast\\_terning.tcl](http://hug.it.edu:8002/F2002/lec4/kast_terning.tcl)

## Regneservice (regneservice.tcl)

Vi har brug for følgende til vores regneservice:

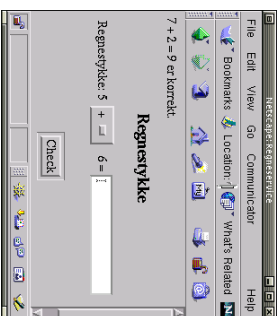
- en menu hvor man kan vælge mellem plus og minus.
- et indtastingsfelt til resultatet.
- skjulte felter som husker regnestykket.
- en knap Check, som checker ens indtastning.

Opgave: hvilke indtastefelter har vi brug for?

Opgave: hvilke formvariable har vi brug for?

Kan du udfylde skitsen til formularen med regnestykket:

```
proc form_spg { } {
    set talA [
    set talB [
    return "<form method=post action=regneservice.tcl>
    <input type=hidden name= value=\"\$
    <input type=hidden name= value=\"\$
    Regnestykke: $talA
    <select name= size=1><option value=\"Plus\">+
    <option value=\"Minus\">- </select>
    $talB = <input type=text name=
    <input type=submit name=submit value=Check>
    </form>"
}
```



## Regneservice - del 2 (regneservice.tcl)

```
proc home_page { title body } { ... }

set form_variables 0
if {[info exists talA] && [info exists talB] &&
[info exists res] && [info exists regneart] &&
[info exists submit]} {
    if {[string compare $regneart "Plus"] == 0} {
        if {[expr $talA + $talB] == $res} {
            ns_return 200 text/html [home_page
                $res er korrekt.<sp>[form_spg]]
        } else {
            ns_return 200 text/html [home_page
                "Regneservice" "$talA + $talB =
                $res er forkert.<sp>[form_spg]]"]
        }
    } else {
        if {[expr $talA - $talB] == $res} {
            ns_return 200 text/html [home_page
                $res er korrekt.<sp>[form_spg]]
        } else {
            ns_return 200 text/html [home_page
                "Regneservice" "$talA - $talB =
                $res er forkert.<sp>[form_spg]]"]
        }
    }
} else {
    ns_return 200 text/html [home_page "Regneservice" "[form_spg]"] }

Sø http://hug.it.edu:8002/F2002/lec4/regneservice.tcl
```

## Afsending af Email fra Tcl Program på Web-Server

Kommandoen ns\_sendmail bruges til at sende email fra Tcl programmer på web-serveren:

ns\_sendmail to from subject body

### Eksempel (email.tcl)

```
set manglende_afleveringer 7
set oevelse 2
ns_sendmail "mael@it.edu" "bjergoe@it.edu" "Oevelse $oevelse" "
Hej Martin,
```

Der er stadig \$manglende\_afleveringer studerende der ikke har afleveret oevelse \$oevelse.

Mvh. Frank"