



Innføring i Dynamisk HTML

Her kommer en enkel innføring i dynamisk HTML (DHTML).

DHTML er en webside (et web dokument) som kan endres etter at den er lastet inn i webleseren. Når brukeren fører musepekeren over en setning el. et ord kan bakgrunnsfargen endres. Måten HTML koden kan endres på etter at siden er lastet inn i webleseren skyldes tre teknologier:

- **Klient skript (JavaScript/VBScript)**

JavaScript er en velkjent metode for å endre en HTML side. JavaScript koden kan endre HTML elementene p.g.a.:

- **"DOM – Document Object Model"**

"Document Object Model" gjør hvert HTML element tilgjengelig for JavaScript. Alle elementene i en webside fra webleserens versjonsnummer til de enkelte HTML knagger er en del av DOM.

- **Stilsett (Cascading Style Sheets – CSS)**

Fordi CSS egenskapene er en del av DOM blir de tilgjengelige for JavaScript (og VBScript). Stilsett lar deg plassere elementer nøyaktig på siden, angi størrelse, farger osv. på HTML elementene.

Vi kan si at CSS og HTML knappene er det du endrer, DOM muliggjør endringene, og JavaScript er det som virkelig foretar endringene.

Du må ha versjon 4 el. senere av webleseren (Internet Explorer el. Netscape Communicator) for at dette skal fungere. Internet Explorer ver. 3 har en viss støtte for DHTML.

La oss se på noen eksempler:

Sette for- og bakgrunns-farge

I HTML kan du selvsagt endre fargen på teksten med å angi `TEXT="blue"` i BODY knaggen. Dersom du ønsker å ha ulike farger på ulike elementer, f.eks. svart tekst på hvit bakgrunn for et normalt avsnitt, og blå tekst med rød bord omkring for avsnitt som skal utheves, gjøres det enkelt med stilsett.

Et stilsett kan plasseres i HEAD seksjonen på et vanlig HTML dokument, eller du kan skrive stilsettet i en egen fil (tekstfil), og lenke den inn i HTML dokumentet. Her skal vi bruke den første metoden, senere skal vi vise den andre metoden.

Start en tekstbasert webeditor og skriv inn denne HTML koden:

```
<HTML>  
<HEAD>
```

Thors innføring i Dynamisk HTML

```
<TITLE>DHTML 1</TITLE>
<STYLE TYPE="text/css">
  P {color:black; background-color:white; margin-left:15px}
  DIV {color:blue;background-color:yellow;
      border-color:red;border-style:solid;border-width:2px;}
</STYLE>
</HEAD>
<BODY>
<H3>Bruke stilsett til å sette farger</H3>
<P>
  Dette er en vanlig avsnitt som skrives med svart tekst på
  hvit bakgrunn. Venstre marg er rykket inn 15 pixel.
</P>
<DIV>
  Dette er et uthevet avsnitt med blå tekst på
  gul bakgrunn, omgitt av en rød bord.
</DIV>
</BODY>
</HTML>
```

Vis denne HTML siden i Internet Explorer 4 (el. senere) eller Netscape Navigator 4 (el. senere). Det normale avsnittet bør være rykket litt inn i forhold til overskriften, og det uthevede avsnittet under. Det uthevede avsnittet bør ha gul bakgrunn med en rød bord omkring.

Prøv å endre størrelsen på webleserens vindu. Det uthevede avsnittet vil automatisk endres slik at teksten alltid er innenfor den røde bord.

La oss se på de ulike elementene i HEAD seksjonen. Linjen:

```
<STYLE TYPE="text/css">
```

angir at her begynner stilsettdefinisjonen, stilsettet avsluttes med koden

```
</STYLE>
```

Linjen:

```
P {color:black; background-color:white; margin-left:15px}
```

definerer stilreglene for avsnitt. Teksten skal være sort (black), bakgrunnen hvit, og venstre marg skal rykkes inn 15 billedpunkter (margin-left:15px). Linjen:

```
DIV {color:blue;background-color:yellow;
border-color:red;border-style:solid;border-width:2px;}
```

definerer stilreglene for DIV beholderen. All tekst som plasseres inni DIV beholderen får blå farge på gul bakgrunn. Teksten blir omgitt av en rød bord/kant (border-color:red) med en hel strek (solid) som er to billedpunkter bred.

I eksemplet over brukte vi P og DIV elementene for å lage ulike avsnitt, men hvordan få til flere ulike typer avsnitt? Det kan være at vi ønsker noen avsnitt med kursiv tekst, noen avsnitt med uthevet tekst osv.

Løsningen er ulike klasser av f.eks. P elementet. De defineres i stilsettdelen med å føye til punktum og et navn, etter navnet på HTML elementet. Her er HTML koden som viser dette:

```
<HTML>
```

```
<HEAD>
<TITLE>DHTML 2</TITLE>
<STYLE TYPE="text/css">
  BODY { font-family: verdana, arial, helvetica, sans-serif }
  P {color:black; background-color:white; margin-left:15px}
  P.kursiv {color:black; background-color:white; margin-left:15px;
           font-style:italic}
  P.uthev {color:blue; background-color:yellow; font-weight:bold;
           border-color:red; border-style:ridge; border-width:10}
</STYLE>
</HEAD>
<BODY>
<H3>Bruke stilsett til å sette farger</H3>
<P>
  Dette er et vanlig avsnitt som skrives med svart tekst på
  hvit bakgrunn. Venstre marg er rykket inn 15 pixel.
</P>
<P CLASS=kursiv>
  Dette avsnittet blir vist med kursiv tekst.
</P>
<P CLASS=uthev>
  Dette er et uthevet avsnitt med uthevet blå tekst på
  gul bakgrunn, omgitt av en rød bord.
</P>
</BODY>
</HTML>
```

Linjen:

```
P.kursiv {color:black; background-color:white; margin-left:15px;
font-style:italic}
```

definerer stilregel for klassen kursiv. I HTML teksten bruker vi denne klassen slik:

```
<P CLASS=kursiv> Dette avsnittet blir vist med kursiv tekst. </P>
```

Sette marger og innrykk

I vanlig HTML er det få muligheter til å lage ulike marger og innrykking av teksten. Med stilsettregler kan du endre margene for f.eks. et avsnitt med margin-left osv. Tekstinnrykk kan lages med stilsettregelen text-indent.

La oss utvide eksemplet over, og legge til et avsnitt (foran alle de andre) med 15% mindre høyre og venstre marger enn de andre avsnittene. Teksten utheves med fet skrift ("bold"). La oss gi dette avsnittet klassenavnet Ingress.

I tillegg lager vi et avsnitt for såkalte hengende innrykk (klassenavn Henge), der teksten i første linje er rykket til venstre i forhold til resten av avsnittet Det får vi til med å angi et negativt tekstinnrykk("text-indent: -4%").

Hent inn eksemplet over og rediger det slik:

```
<HTML>
<HEAD>
<TITLE>DHTML 2b</TITLE>
<STYLE TYPE="text/css">
  BODY { font-family: verdana, arial, helvetica, sans-serif }
```

Thors innføring i Dynamisk HTML

```
P {color:black; background-color:white; margin-left:15px}
P.Ingress {color:black; background-color:white;
           font-weight:bold; margin-left:15%; margin-right:15%;}
P.kursiv {color:black; background-color:white; margin-left:15px;
          font-style:italic}
P.uthev {color:blue; background-color:yellow; font-weight:bold;
         border-color:red; border-style:ridge; border-width:10;
         margin-left: 2px }
P.Henge {color:black; background-color:white;
         text-indent: -4%; margin-left:5%; margin-right:1%;}
</STYLE>
</HEAD>
<BODY>
<H3>Bruke stilsett til å sette farger</H3>
<P CLASS=Ingress>
  Dette avsnittet har både venstre og høyre marg
  rykket inn 15% i forhold til overskriften.
</P>
<P CLASS=Henge>
  Dette er et eksempel på et avsnitt med hengende innrykk.
  Det vil si at første linje er rykket til venstre i forhold
  til resten av avsnittet.
</P>
<P>
  Dette er et vanlig avsnitt som skrives med svart tekst på
  hvit bakgrunn. Venstre marg er rykket inn 15 pixel.
</P>
<P CLASS=kursiv>
  Dette avsnittet blir vist med kursiv tekst.
</P>
<P CLASS=uthev>
  Dette er et uthevet avsnitt med blå tekst på
  gul bakgrunn, omgitt av en rød bord.
</P>
</BODY>
</HTML>
```

Posisjonere elementer

I HTML brukes *IMG* knaggen til å vise bilder. Med *IMG* knaggen kan du angi bildets størrelse, om det skal plasseres til venstre (*ALIGN=left*), til høyre (*ALIGN=right*) osv. De som laster ned en webside kan ha en skjermopløsning som er ulik den utvikleren av siden brukte. Det er også mulig å endre vindustørrelsen til webleseren. Alt dette kan influere på hvordan bildet blir vist. Med stilsett (CSS) kan bildets plassering angis nøyaktig:

```
<HTML>
<HEAD>
<STYLE TYPE="text/css">
<!--
BODY { font-family: verdana, arial, helvetica, sans-serif }

#Absolutt {position:absolute; top:100px; left:50px;
           height:104; width:240px; z-index:0}

#Relativ {position:relative; left: 50px;
          height: 50px; width: 300px; z-index: 1}
-->
```

```
</STYLE>
</HEAD>
<BODY bgcolor="white" text="black" >
<H3>Posisjonering med stilsett</H3>
<DIV ID="Absolutt">
  <IMG SRC="logo.gif"> </DIV>
  Bildet i dette eksemplet blir plassert 100 billedpunkter
  fra øvre kant, og 50 billedpunkter fra venstre kant.
  Legg merke til at bildet overlapper teksten slik at den blir
  vanskelig å lese.
<DIV ID="Relativ">Mens denne teksten blir plassert oppå
  bildet og kan da leses.</DIV>
</BODY>
</HTML>
```

Legg merke til at vi bruker DIV elementet som en "beholder" for de elementer som vi vil plassere med stilsett koder. Z-index (pos. verdi el. 0) brukes til å angi hvilket element som skal være øverst. Linjene

```
<!-- og -->
```

angir at det som ligger mellom dem er en kommentar. Eldre weblesere vil dermed ikke bry seg om stilsett koden. Linjen

```
-->
```


avslutter kommentaren. Nyere weblesere (fra ver. 4) vil derimot ikke bry seg om

```
<!-- og -->
```

linjene, men lese linjene som kommer imellom som et stilsett. På denne måten skjuler vi stilsettkoden for eldre weblesere, mens nyere weblesere vil finne stilsettet. Linjen #Absolutt angir at HTML elementet skal posisjoneres absolutt, 100 billedpunkter fra toppen, 50 billedpunkter fra venstre kant. Så uansett hvordan du endrer webleser vinduet sin størrelse, så vil bildet alltid plasseres 100 punkter ned fra øvre kant, 50 punkter fra venstre kant.

Linjen #Relativ angir at HTML elementet skal plasseres i forhold til foregående element.

Bruke JavaScript til å endre bilde som musepeker passerer over

Mange websider bruker et bilde av en trykk knapp som peker (lenk) til en annen side, når musepekeren passerer over bildet endres bildet (f.eks. blir kantene røde).  Denne effekten lages med de to JavaScript funksjonene onMouseOver og onMouseOut.

Først har vi behov for to bilder av knappen, ett for knappens normale utseende, og ett for hvordan knappen skal se ut når musepekeren passerer over den:

Start deretter opp web redigeringsverktøyet ditt og skriv inn denne HTML koden:

```
<HTML>
<HEAD>
  <STYLE TYPE="text/css">
  <!--
  BODY { font-family: verdana, arial, helvetica, sans-serif }
  -->
</STYLE>
<SCRIPT language="javascript">
```

Thors innføring i Dynamisk HTML

```
var BtnArr=new Array(); /* Lag tabell med navn på btn.bilder*/
  BtnArr["Normal"]='btna.gif'; /* 1. element; normal knapp. */
  BtnArr["Over"]='btnb.gif'; /* 2. element; mus over knapp*/

function EndreBilde(Bilde)
{ /* I IMG knagg må du bruke: NAME="Knapp1". */
  document.Knapp1.src=BtnArr[Bilde]; /* Sett navn på knapp bilde.*/
}
</SCRIPT>
</HEAD>
<BODY bgcolor="white" text="black" >
<H3>Bruke JavaScript til å endre bilde</H3>
Her brukes JavaScript funksjonen onMouseOver og onMouseOut til
å endre bildet av trykk knappen når musepeker
befinner seg over bildet av knappen.
<DIV ID="M_Hover">
<a href="#" onMouseOver="EndreBilde('Over')"
  OnMouseOut="EndreBilde('Normal')">
  <Img src="btna.gif" NAME="Knapp1"></a>
</DIV>
</BODY>
</HTML>
```

I dette eksemplet setter vi stilsett som vi har gjort tidligere (BODY {...}), men like under finner vi knaggen <SCRIPT LANGUAGE="javascript">. Denne knaggen angir at her starter et skript i språket JavaScript.

Først opprettes en tabell – BtnArr – med *new* funksjonen. Deretter legger vi inn navnene på de to bildene av trykk knappene (btna.gif og btnb.gif) som elementer i tabellen. På denne måten kan vi i JavaScript lagre filnavnene til de knapper vi har behov for. Legg merke til den måten vi indekserer tabellen, med beskrivende ord som Normal og Over. Her vil Normal angi btna.gif – det bildet som vises når musepeker ikke er over noen del av bildet, mens Over angir btnb.gif som vises når musepeker befinner seg over bildet.

Derneft må vi ha en JavaScript funksjon som kan endre bildet når musepeker passerer over. Dette gjøres i funksjonen EndreBilde:

```
function EndreBilde(Bilde)
{ /* I IMG knagg må du bruke: NAME="Knapp1". */
  document.Knapp1.src=BtnArr[Bilde]; /*Endre navn på knappbilde*/
}
```

Funksjonen tar en parameter – Bilde – som må ha en av verdiene Normal eller Over. Objektet document.Knapp1 vil angi et objekt i webdokumentet. Her har vi gitt det navnet Knapp1, og vi endrer egenskapen src. Navnet Knapp1 må vi bruke som attributt i knaggen i webdokumentet.

Selve pekeren () plasseres inni en

```
<DIV ID="M_Hover">
```

beholder. DIV er en beholder som (sammen med SPAN) brukes mye i CSS sammenheng til å plassere elementer som skal gis en spesiell posisjon e.l.l.

Test

Dersom du også ønsker å ha et eget bilde som vises når en knapp er trykket inn, lager du et eget bilde (f.eks. som vist til høyre over, og lagrer det som btnc.gif). Deretter redigerer du JavaScript koden under variabeldeklarasjonen, der du føyer til en ny linje:

```
BtnArr["Aktiv"]='btnc.gif';
```

I tillegg må du føye til en onClick referanse i selve peker referansen nede i BODY seksjonen :

```
<DIV ID="M_Hover">
  <a href="#" onMouseOver="EndreBilde('Over')"
    onClick="EndreBilde('Aktiv')"
    onMouseOut="EndreBilde('Normal')">
    </a>
</DIV>
```

Dermed vil knappen vises inntrykket nå du klikker på bildet med venstre museknapp.

Bruke JavaScript og stilsett til å flytte bilder

Stilsett brukt sammen med JavaScript kan gi dynamiske HTML sider. For å kalle opp JavaScript funksjonen som flytter bildet kan vi enten bruke en peker (link) slik som i dette eksemplet. Eller vi kan lage et HTML skjema (Form) med trykknapper.

Dersom brukeren klikker på en peker (link) så beveger for eksempel et bilde seg på skjermen. La oss se på et eksempel som flytter dette dokumentets logo fram og tilbake på skjermen. I <HEAD> seksjonen må vi ha både et stilsett definisjon og et skript (JavaScript i vårt eksempel). I <BODY > seksjonen plasserer vi bildet som skal flyttes inni en <DIV ID="Bilde"> knagg.

Start en tekstbasert HTML editor og skriv inn denne HTML koden:

```
<HTML>
<HEAD>
<TITLE>DHTML 5</TITLE>
<STYLE TYPE="text/css">
  <!--
  BODY { background-color: white ; color: black}

  #Bilde {position:absolute; top:50px; left:250px;
          height:104; width:240px; z-index:0}
  -->
</STYLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
<!--
var StdPos = 250;    // Standard plassering av bildet.

function FlyttBilde(n) // Denne funksjonen flytter bildet.
{
  if (document.layers) // ** Dersom "layers" objektet finnes (NN)
  {
    // alert("Netscape Navigator");
    if (document.Bilde.left > 640) // Ikke flytt for langt til V.
      { document.Bilde.left = document.Bilde.left + n; }
    else { document.Bilde.left = 640; }
    if (document.Bilde.left < 0) // Dersom bildet flyttes forbi 0,
      { document.Bilde.left = StdPos; } // flytt til Std.pos.
  }
  else if (document.all) // ** Dersom "all" objektet finnes (IE)
  {
```

Thors innføring i Dynamisk HTML

```
// alert("Internet Explorer");
if(parseInt(Bilde.style.left) < 640)
  { Bilde.style.left = parseInt(Bilde.style.left) + n; }
else { Bilde.style.left = StdPos; }
  if (parseInt(Bilde.style.left) < 0)
    { Bilde.style.left = StdPos; }
}
}

function V_Klikk() // For å flytte bildet til venstre.
{
  FlyttBilde(-5);
}

function H_Klikk() // For å flytte bildet til høyre.
{
  FlyttBilde(5);
}
//-->
</SCRIPT>

</HEAD>
<BODY>
<H1>D-HTML bevege bilde</H1>
<DIV ID="Bilde">
  <IMG SRC="logo.gif">
</DIV>
<BR><BR> <BR><BR>
<P>
  Demo av stilsett med absolutt posisjonering, og bruk av JavaScript
  til å flytte bildet fram og tilbake på skjermen.

<!-- Bruk pekere til å registrere museklikk for bilde bevegelse -->
<a href="#" onClick="V_Klikk()">Flytt til venstre</a>
<a href="#" onClick="H_Klikk()">Flytt til høyre</a>
</P>
</BODY>
</HTML>
```

Her bruker vi #Bilde i stilsettdefinisjonen til å gi navn (ID) til det element (DIV) som skal inneholde bildet vi vil bevege omkring på skjermen. Vi angir at bildet skal posisjoneres absolutt, 250 billedpunkter fra venstre skjermkant, og 50 billedpunkter fra øvre kant. I tillegg angir vi bredde og høyde på elementet (her brukes bildets virkelige bredde/høyde).

I tillegg til stilsettet må vi ha JavaScript kode som reagerer på brukerens museklikk. Problemet er at Netscape Navigator 4.x og Internet Explorer 4 (el. senere) ikke bruker samme syntaks for å angi elementene i et HTML dokument!

I Netscape vil vi finne objektet "document.layers", mens i Internet Explorer vil vi finne objektet "document.all". I JavaScript koden må vi derfor bruke noen if setninger til å sjekke hvilken webleser som brukes, og ha et sett kodelinjer for hver webleser. Se <SCRIPT...> seksjonen.

I

```
<BODY>
```

seksjonen av HTML dokumentet må vi bruke elementet DIV med ID="Bilde" og så plassere IMG knaggen inni DIV elementet:

```
<DIV ID="Bilde">
  <IMG SRC="LOGO.GIF">
</DIV>
```

Til slutt bruker vi JavaScript funksjonen `onClick` i hvert peker element til å kalle opp funksjonene `V_Klikk` el. `H_Klikk` som flytter bildet til respektive venstre el. høyre med 5 billedpunkter for hvert klikk med venstre museknapp.

Lage websider for både Internet Explorer og Netscape Navigator

Det faktum at Microsoft og Netscape har valgt litt ulike måter å implementere stilsett på lager problemer dersom vi ønsker å lage websider der stilsett og JavaScript fungerer med begge weblesere.

En mulighet er å bruke *if* setninger i hver webside, slik at sidene vises korrekt med begge weblesere (slik vi gjorde i eksempelet over). En annen mulighet er å lage to ulike websider, den ene laget for Internet Explorer, og en annen (enkler?) laget for Netscape Navigator. Så lar vi *index.htm* filen inneholde en enkel test på hvilken webleser brukeren har, og så redirigeres brukeren til den rette versjonen. *Index.htm* blir da en side uten noe tekst i kroppen (BODY seksjonen), slik:

```
<HTML>
<HEAD>
  <SCRIPT language="Javascript">
    function RedirectPage()
    {
      if(document.layers) // If Netscape, load Navigator version of page
      { parent.location.href = "indexNS.htm"; }
      else if (document.all) // IE, load IE version of page
      { parent.location.href = "indexIE.htm"; } // else ???
    }
  </SCRIPT>
</HEAD>
<BODY onLoad="RedirectPage();" >
</BODY>
</HTML>
```

Du må i tillegg lage *indexNS.htm* og *indexIE.htm* sidene.

Stilsett i en ekstern fil

I de eksemplene vi har sett på til nå har stilsett reglene blitt plassert i HEAD seksjonen mellom knaggene

```
<STYLE...> og </STYLE>.
```

Når hver side har ulike stilsetregler kan dette være en grei løsning. Men i mange tilfeller ønsker vi at alle websidene skal ha et noenlunde felles utseende. F.eks. at alle overskrifter skal vises med samme skriftsnitt (font) og størrelse, marger og navigasjonselementer skal se like ut osv. Dette oppnår vi enkelt med å skrive stilsetreglene inn i en vanlig tekstfil (uten `<STYLE>` knaggene) som vi lagrer med *.CSS* som etternavn. Deretter lenker vi inn den eksterne filen i webdokumentet

vårt. Her kommer et eksempel på et eksternt stilsett (lagre som test.css):

```
/*----- Stilsett eksempel (CSS) ----- */
/* Sett standard font, nullstill marger.*/
BODY { font-family: verdana, arial, helvetica, sans-serif;
margin-left: 0pt; margin-top: 0pt; margin-right: 0pt }

/* Sett font for overskrifter, og innrykk(dvs. marg).*/
H1, H2, H3, H4, H5 { font-family: "times new roman", times, serif;
text-indent: 0.5em; }
```

Her kommer et eksempel som viser hvordan stilsettfilen over brukes i et HTML dokument:

```
<HTML>
<HEAD>
<!-- Hent inn et eksternt stilsett fra fila TEST.CSS -->
<LINK REL=STYLESHEET TYPE="text/css" HREF="test.css">
</HEAD>
<BODY>
<H2>Eksternt stilsett eksempel</H2>
Tekst her....
</BODY>
</HTML>
```

Linjen `<LINK REL=...>` henter inn stilsettet fra fila test.css. Dermed kan vi legge denne linja inn i alle websidene våre og dermed blir alle overskrifter like, og all brødteksten lik. Dersom vi senere ønsker et annet utseende på overskriftene, behøver vi kun gjøre endringene ett sted (i test.css).

Bruke JavaScript til å kontrollere feltene i et skjema

Dersom du ønsker at de som besøker hjemmesiden din skal kunne fortelle deg hva de synes, kan du enten opplyse e-post adressen din, eller du kan f.eks. lage et skjema (HTML form) som brukerne fyller ut. Dette skjemaet kan så sendes til webserveren din med CGI (se innføring i CGI4VB). Men dersom bruker ikke har fyllt ut de nødvendige feltene kan det oppstå problemer. Selvsagt kan CGI skriptet lages slik at det sjekker om feltene er fyllt ut, men det enkleste er å lage en JavaScript funksjon som sjekker om inndatafeltene er tomme el. ikke.

Et skjema kan f.eks. se slik ut:

```
<FORM ACTION="cgi-bin/mittcgiskript" NAME="MittSkjema">
  Navn : <INPUT type="text" name="Navn"><BR>
  Alder : <INPUT type="text" name="Alder"><BR>
  E-post: <INPUT type="text" name="Epost"><BR>
  <INPUT TYPE="SUBMIT" VALUE="Send">
</FORM>
```

La oss si at vi vil kreve at feltet "Navn" må fylles ut. Vi må da gjøre to endringer, den første er å bytte ut "Submit" knappen med en vanlig, og den andre er å lage en JavaScript funksjon:

Dvs. endre:

```
<INPUT TYPE="SUBMIT" VALUE="Send">
```

til:

Thors innføring i Dynamisk HTML

```
<INPUT TYPE="BUTTON" VALUE="Send" onClick="SjekkFelt();">
```

Grunnen til at vi ikke ønsker en "Submit" knapp er at vi får JavaScript til å utføre "submit" funksjonen (en "Submit" knapp ville også sende skjemaet selv om feltet Navn var tomt).

Deretter lager vi en SCRIPT seksjon i HTML koden, og skriver inn funksjonen *SjekkFelt()*:

```
<SCRIPT language="JavaScript">
function SjekkFelt()
{
  if (document.MittSkjema.Navn.value == "")
  {
    alert("Du må skrive inn et navn!");
    return;
  }
  document.MittSkjema.submit();
}
</SCRIPT>
```

Funksjonen starter med å kontrollere om verdien (value) til feltet Navn er tomt. Dersom feltet er tomt vises en beskjed (alert) på skjermen, og funksjonen returnerer. Dvs. at bruker får en ny sjanse til å skrive inn noe i feltet Navn. Dersom Navn er fylt ut, sendes skjemaet med kallet:

```
document.MittSkjema.submit();
```

Slik fortsetter vi med å kontrollere de andre feltene. Først sjekker om feltet Alder inneholder fornuftige verdier. Funksjonen *isNaN* sjekker at verdien til Alder ikke er et tall ("isNotaNumber"), dermed elimineres tekst og et tomt felt. I e-post feltet sjekker vi om bruker har tastet inn krøllalfa (@) ved hjelp av *indexOf* metoden.

Fullstendig utlisting av HTML koden:

```
<HTML>
<HEAD>
<TITLE>Bruke JavaScript til å kontrollere innfelt</TITLE>
<META name="description" content="JavaScript til å kontrollere innfelt">
<META name="keywords" content="JavaScript">
<META name="Author" content="TH">
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#800080">
<H1>Bruke JavaScript til å kontrollere innfelt</H1>
<FORM ACTION="cgi-bin/mittcgiskript" NAME="MittSkjema">
  Navn : <INPUT type="text" name="Navn"><BR>
  Alder : <INPUT type="text" name="Alder"><BR>
  E-post: <INPUT type="text" name="Epost"><BR>
  <INPUT TYPE="BUTTON" VALUE="Send" onClick="SjekkFelt();">
</FORM>
<SCRIPT language="JavaScript">
<!--
function SjekkFelt()
{
  if (document.MittSkjema.Navn.value == "")
  {
    alert("Du må skrive inn et navn!");
    return;
  }

  var Alder = parseInt(document.MittSkjema.Alder.value);
  if (isNaN(Alder) || Alder < 1 || Alder > 150)
```

Thors innføring i Dynamisk HTML

```
{
  alert("Ikke gyldig alder (1-150)");
  return;
}

if (MittSkjema.Epost.value.indexOf('@',1) == -1)
{
  alert("Ikke korrekt e-post adresse!");
  return;
}

document.MittSkjema.submit();
}
-->
</SCRIPT>
</BODY>
</HTML>
```