



## Tilpasning til skærm og browser

**Du har lavet en kode, som fungerer fortrinligt på din egen computer. Glad og tilfreds lægger du din hjemmeside på nettet. Dagen efter får du at vide af dine bekendte, at hjemmesiden ser helt forkert ud hjemme hos dem.**

Skrevet den **03. Feb 2009** af **per1291** i kategorien **Programmering / JavaScript** | ★★★★★

Selv om en kodelump virker på din egen computer, så kan du ikke være sikker på at den virker på samme måde på andre computere. Der skal tages højde for, at andre brugere har andre skærme - og andre browsere.

Lad os først kigge på, hvordan siden tilpasses andre skærme.

FASTSÆT STØRRELSEN AF DET DU VISER

Din side ser pæn ud på din egen skærm. Man kan se hele teksten på én gang, altså uden at bruge scroll-pilene. Du vil gerne have siden til at se cirka ligesådan ud på andre skærme.

Hvis brugeren har en skærm med en meget lav opløsning eller et meget smalt vindue, eller hvis brugeren har aktiveret "Vis oversigt" i sin browser, så kan det blive nødvendigt at din tekst fylder hele den tilgængelige del af vinduet - og måske er dette ikke engang nok til, at hele teksten kan vises på én gang (selv om det ville være det mest elegante). Så må du finde dig i, at teksten vises med scrollbars. Hvis brugeren har en skærm med meget høj opløsning, og samtidig et meget bredt vindue på sin skærm, så skal din tekst ikke fylde hele vinduet - for så bliver linjerne så lange at de er svære at læse.

Siden fylder 600 pixels i bredden. Opgaven består derfor i at definere et tekstområde, som skal være præcis 600 pixels bredt.

Metode:

1. Teksten anbringes i en div "BodyDiv", som bringes til at fylde hele den tilgængelige del af vinduet.
2. Bodybredde og Bodyhøjde aflæses, for at se hvor meget SYNLIG plads der i virkeligheden er adgang til.
3. Divbredden sættes til 600 pixels, og Divhøjden sættes automatisk til den nødvendige størrelse.

OBS: I punkt 1 har jeg givet "BodyDiv" en lysegrå baggrundsfarve, sådan at du tydeligt kan se hvor meget den fylder.

OBS: Aflæsningerne i punkt 2 er i virkeligheden overflødige, men foretages for at tydeliggøre hvad der sker.

OBS: I punkt 3 fremkommer en vandret scrollbar, såfremt Bodybredden er smallere end 600 pixels.

I bunden af artiklen ses koden, nemlig i dokumentets første javascript.

Prøv at køre koden både med og uden "Vis oversigt", og med forskellige vinduesstørrelser, for at se forskellen.

(OBS: For at opnå et pænt resultat kunne det være fristende, i stedet at ændre brugerens VINDUESSTØRRELSE efter dine behov. I min første version af denne artikel var det præcis dét jeg gjorde. Men jeg ønsker især at fremme "god programmeringsskik", og ekspertbrugeren roenving gjorde venligt opmærksom på, at det ikke er hensigtsmæssigt at pille ved den vinduesstørrelse, som brugeren har valgt som sin foretrukne. Jeg har derfor i denne version kun vist, hvordan man ændrer på størrelsen af **den del af det givne vindue, som skal bruges.**)

Resten af denne artikel handler om tilpasning til browseren.

## DET TRADITIONELLE BROWSERCHECK

I 1998 lavede jeg min første hjemmeside vha. HTML og Javascript. Jeg lærte de to sprog ud fra oversigter, vejledninger og kodeeksempler på Internet. Men i årenes løb har sprogene udviklet sig, og mange af anvisningerne er nu forældede.

Mange af de kodestumper, som brugerne på eksperten.dk indsender, er nogle de har fundet rundt omkring på Internet. En del af dem indeholder browsercheck. Det er ofte tegn på, at programstumpen er nogle år gammel og bør moderniseres. Et browsercheck er en kodestump (ofte kaldet en "browser-sniffer"), som finder NAVNET på brugerens browser. Her har du et eksempel:

```
function browsersnif()
{
if (navigator.userAgent.toLowerCase().indexOf("netscape")>=0) {return "Netscape"}
if (document.all) {return "MSIE"}
return "Ukendt"}
}
```

Funktionen vil returnere "Netscape" hvis den første if-sætning er sand, "MSIE" hvis den anden if-sætning er sand, og "Ukendt" i alle andre tilfælde.

Der er brugt to forskellige checkmetoder i den viste funktion. Netscape-checket spørger browseren direkte, hvad den selv hedder. Det er altså et rent NAVNECHECK. Microsoft Internet Explorer-checket spørger browseren, om udtrykket "document.all" overhovedet giver mening for den. Det kunne man kalde et FEATURECHECK.

Senere i scriptet vil resultatet af dette browsercheck blive brugt til at vælge imellem, hvilke instruktioner der skal udføres. Her et eksempel (som selvfølgelig ikke virker uden noget mere kode):

```
var browser=browsersnif()
if (browser == "Netscape") {Netscape_function()}
if (browser == "MSIE") {MSIE_function()}
```

## NAVIGATOR.USERAGENT

Ved at skrive sådan her i adressefeltet i din browser:

```
java script:alert(navigator.userAgent)
```

kan du få oplysninger om din egen browser og dit styresystem.

I min browser Microsoft Internet Explorer 6.0 er navigator.userAgent =  
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; i-NavFourF)"

I min browser Opera 7.54 er navigator.userAgent =  
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1) Opera 7.54 [da]"

I min browser Netscape 7.2 er navigator.userAgent =  
"Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2 (ax)"

I min browser Mozilla Firefox 1.0 er navigator.userAgent =  
"Mozilla/5.0 (Windows; U; Windows NT 5.1; da-DK; rv:1.7.5) Gecko/20041118 Firefox/1.0"

Hvis jeg selv i dag (2005) ville bruge et browsercheck til at fortælle, hvilken af mine browsere der kører min kode lige nu, så kunne det se sådan her ud:

```
function browsersnif()  
{  
var ua=navigator.userAgent.toLowerCase()  
if (ua.indexOf("opera")>=0) {return "Opera"}  
if (ua.indexOf("msie")>=0) {return "MSIE"} // Bemærk rækkefølgen af msie- og opera-checket!  
if (ua.indexOf("netscape")>=0) {return "Netscape"}  
if (ua.indexOf("firefox")>=0) {return "Firefox"}  
return "Ukendt" // Denne linje tager højde for alle mulige andre browsere, f.eks. Safari  
}  
alert("Browseren er en "+browsersnif())
```

Opera er en meget kræsen browser - vil kun æde kode, som er meget korrekt.  
Vil du tjekke om din kode virker næsten alle steder, så brug Opera.  
Hvis Opera viser din side som den skal, så er det sandsynligt at de fleste browsere gør det samme.

Mozilla Firefox er næsten altædende - tolerant over for en del småfejl.  
Vil du have en browser, der kan forstå næsten alt, så brug Firefox.

## BROWSERCHECK ER SJÆLDENT OPTIMALT

I grunden er browsercheck i mange tilfælde en uhensigtsmæssig metode.  
Hvis det på nogen måde kan undgås at bruge browsercheck, så er det bedst at gøre dette.  
Lad være med at spørge browseren hvad den hedder!

Grundene er disse:

1. Browsercheck kræver stor viden hos programmøren.  
Du skal vide præcis hvilke egenskaber hver enkelt browser har. Det er ikke nok at vide, at "nogle browsere gør sådan, og andre browsere gør sådan" - Du skal holde rede på, præcis hvilke browsere der gør hvad.
2. Visse browsere kan indstilles til at give fejlagtige navneoplysninger om sig selv med vilje - jeg forstår ikke rigtig hvorfor, men sådan er det.
3. Browsercheck tager ikke højde for ukendte browsere.  
Der kan findes browsere på markedet, som jeg aldrig har hørt om - eller der kan blive frigivet en splinterny browser i morgen. Min kode kan derfor ikke tage højde for disse browsere.

4. Mange Browsercheck er forældede.

Det er omsonst at checke, om brugeren har en Netscape 4 eller en Internet Explorer 3 browser. INGEN kører med sådanne forældede browsere i dag. (Og hvis der alligevel skulle være en enkelt Netscape 4-bruger tilbage i verden, så er vedkommende allerede nu blevet vænnet til, at de fleste moderne sider er "lukket land".)

Et browsercheck har normalt begrænset levetid - fordi der hele tiden udkommer nye browsere. Og vi kan ikke engang være sikre på at nye versioner af den samme browser vil være bagud-kompatible.

Erfarne programmører er for længst holdt op med at skrive kode, som tager højde for forældede browsere som Internet Explorer 3 og Netscape 4. I de fleste scripts bruger de slet ikke browsercheck. Til gengæld kender disse programmører en masse browsersvagheder, som de derfor mere eller mindre automatisk skriver udenom.

#### ALTERNATIVET: FEATURECHECK

Jeg ved, at mine browsere hver for sig har deres særheder.

Nogle af disse særheder eller svagheder bliver jeg først opmærksom på, når jeg tester mine sider i de forskellige browsere.

Der findes kode, som Netscape ikke kan håndtere, f.eks. "document.all".

Heldigvis kan denne kode erstattes af "document.getElementById", som alle moderne browsere forstår.

Der findes kode, som Internet Explorer ikke kan håndtere, f.eks. "window.innerWidth".

Heldigvis kan denne kode erstattes af "document.body.clientWidth", som alle moderne browsere forstår.

Det er ofte muligt at finde en notation, som alle browsere forstår.

Men i nogle tilfælde er det alligevel nødvendigt at skrive et stykke kode i mere end én version.

Man kan i mange tilfælde erstatte et browsercheck med noget andet og i virkeligheden bedre.

En gammel talemåde siger: "Du skal gå til hovedet, ikke til halen ..."

Oplysning om browserens NAVN er HALEN. Oplysning om browserens HÅNDBLING af en konkret kode er HOVEDET.

Derfor vil jeg gerne slå til lyd for, at man i stedet for at spørge browseren hvad den hedder, spørger browseren hvordan den vil håndtere en konkret kode.

I de fleste tilfælde kan programmøren være ligeglad med, hvilken browser brugeren benytter.

Det eneste interessante er at vide, hvilke instruktioner der kan forstås og udføres hos den aktuelle bruger. Med andre ord, et browsercheck er ofte overflødig.

Man kan lave bedre og mere levedygtig kode vha. et Featurecheck:

Check hvad browseren GØR (eller VIL gøre) i den KONKRETE situation.

Det er stadig vigtigt for programmøren at vide, at browsere opfører sig forskelligt i bestemte situationer.

Heldigvis er det som oftest ikke nødvendigt at HUSKE præcis hvilke browsere der gør hvad.

Man kan nøjes med at vide, at disse forskelle findes.

Og så kan man i det konkrete tilfælde spørge den konkrete browser: Hvordan reagerer DU på denne kode?

Du kan aldrig sikre dig 100%, at din side vil se rigtig ud i enhver browser.

De færreste er fortrolige med alle browsere og hvilke svagheder de har.  
De færreste har adgang til at teste en side i alle browsere.  
De færreste kender den browser, som først kommer på gaden i morgen.

En ukendt browser kan have andre svagheder end dem, du selv har kendskab til.  
Men sandsynligheden er stor for, at den vil reagere på en konkret kode ligesom Opera, Netscape, Firefox eller Explorer.

Derfor er Featurecheck oftest bedre end Browsercheck.

(En undtagelse er, når man benytter sig af special-features, f.eks. et filter skrevet af Microsoft og udelukkende brugbart i Microsoft Internet Explorer.)

#### EKSEMPEL: EN RULLETEKST

Jeg fik til opgave at lave en rulletekst.

Den rullende tekst skulle være så lang, at den fyldte mere end det definerede tekstområde på 600 pixels. Teksten skulle køre ud i ét, sådan at den hele tiden blev bragt i forlængelse af sig selv, dvs. rullefeltet måtte aldrig blive tomt.

Jeg valgte denne metode (RulleMetode1):

1. Anbring teksten i et rullefelt i form af en div.  
<Løkke>
  2. Rul 1 pixel ad gangen vha. style.left--, indtil du er kommet igennem første bogstav;
  3. Flyt første bogstav bagest i din string;
  4. Anbring den nye string i din div vha. innerHTML;
  5. Nulstil style.left;
- </Løkke>

Denne kode fungerer fint i Internet Explorer og Opera.

Men det viser sig at den giver en flakkende tekst i Netscape og Firefox.

Disse browsere vil nemlig vise et kortvarigt glimt af en blank div, hvergang man ændrer innerHTML.

(Hvis man nøjedes med at ændre innerHTML mens rullefeltet er tomt, vil brugeren ikke opleve nogen flakken i Netscape. Men i den bestilte rulletekst må rullefeltet aldrig være tomt. Der vil altid være bogstaver på skærmen.)

Så jeg lavede en ny kode i det håb, at problemet hermed var løst (RulleMetode2):

1. Sæt din tekststreng S=S+S førend du anbringer den i rullefeltet;  
<Løkke>
  2. Rul 1 pixel ad gangen vha. style.left--, indtil du er kommet igennem første halvdel af din string;
  3. Nulstil style.left;
- </Løkke>

Denne kode fungerer fint i Netscape og Mozilla Firefox.

En test viser at koden også fungerer i Internet Explorer.

Men i Opera opstår et nyt problem:

Rullefeltet bliver bredere end vinduet, sådan at der fremkommer en overflødig scrollbar i bunden af vinduet.

Opera medregner åbenbart også det usynlige overflow i bredden!

Status er altså:

I browseren Opera bør man bruge RulleMetode1.  
I browserne Netscape og Mozilla Firefox bør man bruge RulleMetode2.  
I browseren Internet Explorer kan man bruge begge metoder.

Man kunne fristes til at bruge et browsercheck:  
if (browser == "opera") {RulleMetode1()} else {RulleMetode2()}  
Men det ville være en falliterklæring.

En række test viser, at:  
I Netscape og Mozilla Firefox vil teksten aldrig fylde mere end Bodybredden.  
I Opera og Internet Explorer fylder teksten derimod mere end Bodybredden.

Så det lykkedes at erstatte browserchecket med dette Featurecheck:  
if (findDivbredde(Rulletekst+Rulletekst) > 600) {RulleMetode1()} else {RulleMetode2()}

- - -

Måske vil enhver browser, der behandler overflow på samme måde som Opera, også lave "flakfri" opdatering af innerHTML. Måske. Jeg kan ikke vide det. Ellers må brugeren finde sig i lidt flakken.

Jeg har lavet en generel løsning, som er tilpasset særhederne i alle mine 4 browsere.  
Jeg har taget højde for de forskelle, som jeg kender til.  
Intetsteds i min kode spørger jeg om browserens NAVN.  
Der er ikke garanti for, at min kode giver et pænt resultat i browsere, som jeg ikke kender.  
Men der vil sikkert være nogle af de ukendte browsere, der i en konkret situation gør ligesom Opera, og andre browsere, der gør ligesom Mozilla Firefox.  
Og måske er der browsere, som gør noget helt andet.  
Det kan jeg ikke vide, og det må jeg leve med, indtil en venlig bruger sender mig en meddelelse om fejlen.

Hvis jeg udelukkende havde brugt NAVNECHECK i min kode, så ville min kode have ringere chancer for at fungere i de browsere jeg ikke kender.

En bemærkning på falderebet:  
Hvis brugeren manuelt ændrer størrelsen på sit vindue, mens rulleteksten er i gang, så kan der forekomme uhensigtsmæssigheder.  
Disse uhensigtsmæssigheder kan du selv tage højde for; den sikreste metode er at lave en setTimeout, som løbende aflæser vinduets bredde - og ved ændringer tildeler nye værdier til rulle-variableerne.

SLUT PÅ ARTIKEL! - Her følger et HTML-dokument:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Tilpasning til skærm og browser</title>
<script language="javascript" type="text/javascript">
/* DETTE SCRIPT SØRGER FOR SKÆRMTILPASNING AF ET TEKSTOMRÅDE */

var DivBredde=600; /* antal pixels i bredden */
function AngivDimensioner()
{
```

```

document.getElementById("b1").innerHTML=window.screen.availWidth+" pixels";
document.getElementById("h1").innerHTML=window.screen.height+" pixels";
document.getElementById("b2").innerHTML=document.documentElement.clientWidth+" pixels";
document.getElementById("h2").innerHTML=document.documentElement.clientHeight+" pixels";
document.getElementById("b3").innerHTML=document.getElementById("b2").innerHTML;
document.getElementById("h3").innerHTML=document.getElementById("h2").innerHTML;
document.getElementById("BodyDiv").style.height=document.documentElement.clientHeight+"px";
}
function ResizeBodyDiv()
{
document.getElementById("b4").innerHTML=DivBredde+" pixels";
document.getElementById("BodyDiv").style.width=DivBredde+"px";
document.getElementById("BodyDiv").style.height="auto";
document.getElementById("h4").innerHTML=document.getElementById("BodyDiv").offsetHeight+" pixels";
alert("Slut på resize...\n\nPrøv igen med forskellig højde og bredde på vinduet, og både med og uden 'Vis
oversigt'!")
}
</script>
<script language="javascript" type="text/javascript">
/* DETTE SCRIPT SØRGER FOR BROWSERTILPASNING AF EN RULLETEKST */

```

```

var Rulletekst=" *** Denne rulletekst fungerer i både Microsoft Internet Explorer 6.0, Opera 7.54, Netscape
7.2 og Mozilla Firefox 1.0";
Rulletekst=changechar(Rulletekst," ","\240"); /* For at undgå problemer med blanke i start og slut */
var Minimumbredde=580; /* mindste tekst der må anbringes i Rullefelt */
var movepixels=1 /* antal pixels der flyttes ad gangen */
var milliseconds=20; /* interval mellem opdateringer */
var Tegnbredde, Antaltegn; /* bruges af Rullemetode1 */
var scrolled, Tekstbredde; /* bruges af Rullemetode2 */
var Timer; /* timeout-variabel */

```

```

function changechar(S,oldchar,newchar)
{
return S.split(oldchar).join(newchar);
}
function findDivbredde(S)
{
var U=document.getElementById("usynligdiv");
U.innerHTML=S;
var width=parseInt(U.offsetWidth);
U.innerHTML="";
return width;
}
function findTekstbredde(S)
{
var width=0, U=document.getElementById("usynligtekst");
S = S.split("");
for (var i=0; i<S.length; i++)
{
U.innerHTML=S[i]; width+=parseInt(U.offsetWidth);
}
U.innerHTML="";
return width;
}
function RulleMetode1()

```

```

{
var id=document.getElementById("rullefelt");
Tegnbredde-=movepixels;
if (Tegnbredde <= 0)
{
Tegnbredde+=findTekstbredde(Rulletekst.substring(0,1));
Rulletekst=Rulletekst.substring(1,Rulletekst.length)+Rulletekst.substring(0,1);
Antaltegn--;
var showstring=Rulletekst.substring(0,Antaltegn), i=Antaltegn;
while (findDivbredde(showstring) < Minimumbredde) {showstring+=Rulletekst.substring(i,i+1); i++};
Antaltegn=i;
id.innerHTML=showstring;
}
id.style.left = Tegnbredde+"px";
Timer=window.setTimeout("RulleMetode1()",milliseconds);
}
function RulleMetode2()
{
scrolled-=movepixels; if (-scrolled >= Tekstbredde) {scrolled+=Tekstbredde};
var id=document.getElementById("rullefelt");
id.style.left = scrolled+"px";
id.style.width = (Minimumbredde-scrolled)+"px";
Timer=window.setTimeout("RulleMetode2()",milliseconds);
}

function StartRulletekst()
{
window.clearTimeout(Timer);
var tekstbredde=findTekstbredde(Rulletekst);
var W=document.body.clientWidth; /* Skærmbredden */
var L=0; if (W > 600) {L=Math.round((W-600)/2)}; /* Afstanden fra skærmkant til rulletekst */
var Lstyle=document.getElementById("rulleleft").style;
Lstyle.left=(-L-20)+"px"; Lstyle.width=(L+20+60)+"px"; /* De 20 px udjævner usikkerhed i f.m.
clientWidth) */
document.getElementById("rulleright").style.width=(L+60)+"px";

if (findDivbredde(Rulletekst+Rulletekst) > 600) /* Risiko for en overflødig scrollbar */
{
alert("Rullemetode2 vil medføre at 'rullediv' får en bredde > 600px.\n\nDerfor bruges RulleMetode1");
var showstring="", i=0;
while (findDivbredde(showstring) < Minimumbredde) {showstring+=Rulletekst.substring(i,i+1); i++};
Antaltegn=i;
Tegnbredde=findTekstbredde(Rulletekst.substring(0,1));
document.getElementById("rullefelt").innerHTML=showstring;
RulleMetode1();
}
else
{
alert("Rullemetode2 vil IKKE medføre at 'rullediv' får en bredde > 600px.\n\nDerfor bruges RulleMetode2");
Tekstbredde=findTekstbredde(Rulletekst);
Rulletekst+=Rulletekst;
document.getElementById("rullefelt").innerHTML=Rulletekst;
scrolled=0;
RulleMetode2();
}
}

```



```

}
</script>
<style type="text/css">
body { font-family:arial,verdana,Helvetica,sans-serif; font-size:16px; }
#usynligtekst { position:absolute; left:0px; top:0px; font-weight:bold; height:19px; font-size:15px;
visibility:hidden; }
#usynligdiv { position:absolute; left:0px; top:0px; font-weight:bold; height:19px; font-size:15px;
visibility:hidden; }
#BodyDiv { background-color:#eee; width:100%; }
h1 { font-size:30px; }
td { font-size:16px; }
#rullebackground { position:relative; left:0px; width:600px; top:0px; height:19px; background-color:#111;
}
#rullefelt {position:absolute; left:0px; width:580px; top:0px; height:19px; z-index:3; overflow:hidden; font-
weight:bold; color:yellow; font-size:15px; }
#rulleleft {position:absolute; left:0px; width:60px; top:0px; height:19px; z-index:4; background-
color:white; }
#rulleright {position:absolute; left:540px; width:60px; top:0px; height:19px; z-index:4; background-
color:white; }
</style>
</head>
<body onLoad="AngivDimensioner()">
<!-- Den følgende del af body indeholder tabellen -->
<center>
<div id="BodyDiv">
<h1>Tilpasning til skærm</h1>
<table width="80%" cellspacing="0">
<tr><td width="50%"></td><td width="50%"></td></tr>
<tr bgcolor="yellow"><td>Skærmbredde:</td><td id="b1" align="center"
valign="bottom">&nbsp;</td></tr>
<tr bgcolor="cyan"><td>Skærmhøjde:</td><td id="h1" align="center"
valign="bottom">&nbsp;</td></tr>
<tr bgcolor="yellow"><td>Synlig Bodybredde:</td><td id="b2" align="center"
valign="bottom">&nbsp;</td></tr>
<tr bgcolor="cyan"><td>Synlig Bodyhøjde:</td><td id="h2" align="center"
valign="bottom">&nbsp;</td></tr>
</table>
<h1>"BodyDiv" før Resize:</h1>
<table width="80%" cellspacing="0">
<tr><td width="50%"></td><td width="50%"></td></tr>
<tr bgcolor="yellow"><td>Maksimal synlig Divbredde:</td><td id="b3" align="center"
valign="bottom">&nbsp;</td></tr>
<tr bgcolor="cyan"><td>Maksimal synlig Divhøjde:</td><td id="h3" align="center"
valign="bottom">&nbsp;</td></tr>
</table>
&nbsp;<br>
<button onclick="ResizeBodyDiv()">Udfør resize</button>
&nbsp;<p>
<h1>"BodyDiv" efter Resize:</h1>
<table width="80%" cellspacing="0">
<tr><td width="50%"></td><td width="50%"></td></tr>
<tr bgcolor="yellow"><td>Aktuel Divbredde:</td><td id="b4" align="center"
valign="bottom">&nbsp;</td></tr>
<tr bgcolor="cyan"><td>Aktuel Divhøjde:</td><td id="h4" align="center"
valign="bottom">&nbsp;</td></tr>

```

```
</table>
&nbsp;
[/div] <!-- id="BodyDiv" -->
</center>
<!-- Den følgende del af body indeholder rulleteksten -->
<center>
&nbsp;<br>
<button onclick="StartRulletekst()">Start rulletekst</button>
&nbsp;<p>
<div id="rullebackground">
<div id="usynligtekst">[/div]
<div id="usynligdiv">[/div]
<div id="rullefelt">[/div]<div id="rulleleft">&nbsp;[/div]<div id="rulleright">&nbsp;[/div]
[/div] <!-- id="rullebackground" -->
</center>
</body>
</html>
```

### **Kommentar af roenving d. 30. Jul 2005 | 1**

Det ser ikke ud til at man kan fjerne irrelevante kommentarer, så man må vel nøjes med at konstatere, at den deaktiverede bruger ingeborg08 bare har lukket I\*\*\* ud ...

### **Kommentar af foxmulder58 d. 31. Jul 2005 | 2**

Fin artikel hvor du går i dybden og hvor du illustrerer hvordan man får bugt med de problemer der kan være med en fremvisning af en side korrekt i en given browser.

Nogle af eksempplerne er lidt svære at forstå, men du kunne have afsluttet din artikel med et eksempel på hvordan man løser problemet uden JavaScript (i tilfælde af brugeren skulle have deaktiveret selvsamme i browseren).

Eventuelt en brug af flere designs til forskellige opløsninger.

mvh  
Mads

### **Kommentar af qtax87 (nedlagt brugerprofil) d. 29. Sep 2006 | 3**

ingeborg08> slap da lige af!  
Fin artikel synes der er mange gode eksempler, og gode tanker/ider omkring brugeren af det.

### **Kommentar af serverservice d. 16. Aug 2005 | 4**

En artikel som jeg har ledt længe efter , men skal bruge lidt mere tid på de detaljer. Mange detaljer - Godt lavet og fuld score herfra.

### **Kommentar af visualdeveloper d. 15. Aug 2005 | 5**

### **Kommentar af ingeborg08 d. 29. Jul 2005 | 6**

Mage til lortartikel og makværk skal man sørme lede længe efter!!! Du må have en dårlig smag i munden.