



Start på javascript

En begynder artikel til javascript som vil gennemgå:

1: Intro

2: Variabler

3: Arrays

4: Matematik

5: Betingelser

6: Løkker

7: Funktioner

8: Facitliste (Til de opgaver der kommer igennem artiklen)

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

Forord:

Da jeg selv for et par måneder siden ville starte på javascript havde jeg svært ved at finde nogle gode artikler på dansk.

Hvilket jeg syntes var ret ærgeligt eftersom jeg i hvertfald synes det er nemmere at lære et sprog ved at læse artikler på dansk.

Så nu efter et par måneder hvor jeg selv mener at jeg har rimeligt godt styr på det grundlæggende i javascript, så satte jeg mig for at skrive en artikel som tager det helt fra bunden.

Skulle der være nogle ting jeg har misforstået så sig det endelig og så retter jeg det.

1: Intro

2: Variabler

3: Arrays

4: Matematik

5: Betingelser

6: Løkker

7: Funktioner

8: Facitliste (Til de opgaver der kommer igennem artiklen)

INTRO TIL JAVASCRIPT

Javascript er et såkaldt ClientSide Sprog, i modsætning til f.eks PHP og ASP som er ServerSide Sprog. Det vil sige at Javascript ikke kan kommunikere med serveren ligesom PHP og ASP kan, det kan derimod kommunikere med Klienten/Browseren.

Hvis du kan andre programmeringssprog (nok især PHP), så vil du ikke finde Javascript ret svært at lære da syntaksten er meget ens.

Men lad os hoppe ud i vores første script.

VORES FØRSTE SCRIPT

```
<script type="text/javascript">
document.write("Hello World");
</script>
```

Det første man ser er `<script type="text/javascript">` og det fortæller browseren at den nu skal til at behandle Javascript i stedet for almindelig HTML, du har måske set noget andre javascript hvor det bruger

`<script language="javascript">` i stedet - det gjorde man før i tiden, men i den nye generation af javascript bruger man altså `type`.

Så ser man koden `document.write("");` - det er det stykke kode som fortæller at vi vil udskrive noget til skærmen, det er nemlig ikke ligesom i almindelig HTML hvor alt hvad du skriver bliver udskrevet, Hvis du kan PHP kan det sammenlignes med `echo()` og `print()` og i ASP kan det sammenlignes med `Response.write`.

Alt hvad der står inden for paranteserne bliver udskrevet til browseren.

Og til sidst ser man `</script>` - Det er fordi som du jo nok ved skal man ALTID afslutte sine tags. Ellers kan man risikere at en hel side ikke virker:)

Se det var jo meget nemt at udskrive noget tekst.

Nu skal vi til at lege lidt med variabler.:O

VARIABLER

Og hvad er variabler så? - Jo hvis du kan et hvilket som helst andet programmeringssprog så ved du hvad det er - for det bliver brugt HELE tiden i alle programmeringssprog, så det er noget man skal have 100% styr på hvordan fungerer.

Men hvis du ikke ved det, så vil jeg forklare det her.

En variabel kan sammenlignes med en slags "kasse" hvor du kan putte noget ned i og opbevare det deri indtil du får brug for det.

En variabels navn skal starte med enten et bogstav eller en underscore (`_`), den må ikke starte med et tal. Hvis du synes det lyder mærkeligt, så kig lidt på denne kode:

```
<script type="text/javascript">
var vores_variabel = "Hello World";
document.write(vores_variabel);
</script>
```

Nu kom der noget nyt ind i vores kode nemlig `var vores_variabel` - og det er faktisk vores første variabel, når man laver en variabel - eller "sætter" den som det hedder i programmeringssprog - så starter man i javascript altid med at skrive "var" som er en forkortelse for "variable", så ved browseren at den skal fortolke koden som en variabel.

Vi har givet vores variabel en værdi ved at sige `= "Hello World";`.

Nu kom der noget kode som du burde kunne huske (`document.write`).

Men denne gang er der **ikke** nogle anførelstegn inde i parantesen som du sikkert har opdaget ;)

Og det er fordi at hvis du have gjort det, så havde Javascript opfattet det som en "streng" ("streng" er programmeringsord for "tekst") - og ikke som en variabel som vi vil have den til.

Dvs at hvis vi have brugt anførelstegn så ville der kommer til at stå `"vores_variabel"` i stedet for `"Hello World"` som jo var variabelens værdi.

Nu kan vi finde ud af at opbevare noget tekst i vores variabler - men vi skal vel også kunne opbevare tal? Det gøres faktisk på samme måde som med strenge.

Bare uden anførelstegn

```
var vores_tal = 5;
```

Grunden til at der ikke skal være anførelstegn rundt om et tal er at javascript så fortolker det som en streng i stedet for et tal, og du kan f.eks ikke lave regnestykker med tekst.

Så lang så godt. Nu skal vi igang med noget der hedder arrays :O

ARRAYS

Arrays er faktisk en slags udvidede variabler, og så alligevel ikke... Her kommer noget kode så forklarer jeg bagefter.

```
<script type="text/javascript">
var vores_array = new Array();
  vores_array[0] = "Hej";
  vores_array[1] = "med";
  vores_array[2] = "dig";
</script>
```

Det første som vi ikke har set før er *new Array()* - men det er ganske enkelt fordi vi vil fortælle at vi skal igang med et array.

Efter det kommer der noget som faktisk ligner variabler - og så alligevel ikke, for der er nogle firkantede parenteser på ([]).

Et array består af et X antal variabler med firkantede parenteser på. Inde i de parenteser står der et tal (0-X).

Et array starter ved 0.

Det kan umiddelbart virke som en idiotisk måde at gøre det på da man jo kan spare en masse kode ved bare at bruge variabler -

det er også rigtigt nok, men du bliver glad for arrays når du kommer mere ind i programmeringen.

I en senere artikel vil jeg komme ind på fler dimensionelle arrays.

Men hvis du gerne vil udskrive noget fra dit array, skal du skrive navnet på array'et + nummeret.
eks

```
document.write(vores_array[2]);
```

Den vil udskrive "dig" :)

Nu skal vi til at gå videre - hvis du ikke har helt styr på det vi har gennemgået indtil videre så læs det igen, for det er et MUST at have 100% styr på det.

Hvis du har forstået det og er frisk på mere - så hent dig en kop kaffe, for nu bliver det lidt sværere ;)

Vi skal nemlig igang med noget matematik.:O

Ja det lyder kedeligt, men du vil helt sikkert få brug for det hvis du vil gøre noget ud af din programmering.

MATEMATIK

Matematik i JavaScript minder utroligt meget om det du lærer i skolen, så det er et hurtigt kapitel.

De operatører vi har til rådighed er

- + - Plus
- - Minus
- * - Gange
- / - Dividere
- % - Modulus (tilovers fra en division)

Her kommer en gennemgang af de forskellige operatører.

Plus

Plus bruges ligesom i alt andet matematik til at lægge tal sammen, så 2+2 vil give 4.

Man kan også godt lægge variabler sammen.

eks

```
<script type="text/javascript">
var tal1 = 2;
var tal2 = 2;
var resultat = tal1+tal2;
</script>
```

På den måde kan man lægge variabler sammen så længe de indeholder tal.

Minus, Gange, Dividere

Foregår på samme måde som *plus*

Modulus

Modulus er nok en operator du ikke har set før medmindre du har programmeret andre sprog. Men modulus giver det der er tilover efter en dividering.

Det vil sige at $5\%2$ giver 1 for 2 går op i 5 to gange, og så er der én tilovers.

Det er den éne som modulus returnerer.

Det var lidt om det basale matematik med javascript.

Nu skal vi igang med "betingelser".

BETINGELSER

I javascript findes der noget der hedder betingelser, de kan bruges til at returnere "noget" hvis en forespørgsel er sand eller "nogetandet" hvis den er falsk.

Det lyder måske lidt forvirrende, men her kommer et eksempel med noget forklaring bagefter.

```
<script type="text/javascript">
var dyr = "kat";

if(dyr=="kat"){
    document.write("Dyret er en kat");
}
</script>
```

Først har vi lavet en variabel (dyr) som vi har givet værdien "kat".

Så kommer der noget nyt i vores kode som er *if()*.

"if" er det engelske ord for "hvis", så den spørger altså om "hvis noget".

Inde i vores if står der *dyr=="kat"* det er den forespørgsel som vi laver i vores if.

Læg mærke til at vi har brugt "==" (dobbel lighedstegn).

Det har vi gjort fordi vi skal sammenligne vores værdier, hvis du havde brugt et enkelt lighedstegn havde variabelen dyr blevet sat til "kat" igen.

De grundlæggende sammenligningsmuligheder er:

- == - Lig med
- != - IKKE lig med
- > - Større end
- < - Mindre end
- <= - Mindre eller lig med
- >= - Større eller lig med

Ekstra info

- ++ - Lægges +1 til et tal
- - Trækker -1 fra et tal

(Kommer vi til at bruge senere).

Der findes også flere, men dem vil jeg ikke komme ind på endnu.

Men tilbage til vores forespørgsel `dyr=="kat"` - der spørger vi om variabelen "dyr" er lig med "kat" hvis den er udskriver vi "Dyret er en kat" til browseren, ellers sker der ingenting.

Nu kunne det være meget passende at gøre noget andet hvis dyr IKKE er lig med "kat".

Til det skal vi bruge "else" som er engelsk for "ellers".

Så vores betingelse kunne se sådan her ud:

```
<script type="text/javascript">
var dyr = "kat";

if(dyr=="kat"){
    document.write("Dyret er en kat");
} else {
    document.write("Dyret er ikke en kat");
}
</script>
```

Her spørger vi om dyr er en kat - hvis den er skal den udskrive "dyret er en kat" ellers skal den udskrive "dyret er ikke en kat".

Det burde være til at forstå ellers må du kigge på det én gang til.

Nu skal vi prøve at lave flere tjek i vores betingelse, til det skal vi bruge "if-elseif".

Det lyder måske forvirrende, men det er nemt nok.

Her kommer lidt kode.

```
<script type="text/javascript">
var dyr = "kat";

if(dyr=="kat"){
    document.write("Dyret er en kat");
} else if(dyr=="hund" {
    document.write("Dyret er en hund");
} else {
    document.write("Dyret er hverken en kat eller en hund");
}
</script>
```

Der har vi lavet en "else if" - vores betingelse tjekker altså; er dyr lig med kat - udskriv "dyret er en kat". Ellers hvis dyr er lig med hund - udskriv "dyret er en hund". Ellers skal den skrive "dyret er hverken en kat eller en hund".

På den måde kan vi lave flere forespørgelser i stedet for at vi skal lave flere if'er inde i hinanden.

Det var det mest grundlæggende inden for betingelser i javascript.

Det næste vi skal kigge på er noget der hedder løkker.

LØKKER

Løkker er noget som kan køre en forespørgsel flere - lige så mange gange som forespørgelsen bliver opfyldt.

Vi skal kigge på 2 forskellige løkker; "for" og "while".

Lad os starte med while, den har 2 parametre som skal være opfyldt før den kan køre;

```
while(noget forespørgsels noget andet){  
  <!--gør noget-->  
}
```

Tidligere i artiklen gav jeg dig lidt "ekstra info" nemlig ++ og -- som henholdsvis siger +1 & -1 til et tal. Det kan vi bl.a. bruge i vores løkker.

Vi kan lave en løkke som hele tiden lægger +1 til et tal indtil tallet når 100.

Det kommer til at se sådan her ud.

```
<script type="text/javascript">  
var i = 0;  
  while(i<=100){  
    document.write(i);  
    i++  
  }  
</script>
```

Der sætter vi først en variabel lig med 0.

Så laver vi en løkke som skal køre så længe at tallet er mindre eller lig med 100.

I slutningen af løkken lægger vi +1 til vores variabel, og genstarter løkken.

Det var et eksempel på en simpel while løkke, vi kan faktisk få det samme resultat med en for løkke som jeg snakkede om tidligere.

Det kommer til at se sådan her ud.

```
<script type="text/javascript">  
  for(var i=0; i<=100; i++){  
    document.write(i);  
  }  
</script>
```

I modsætning til en while løkke så har for løkken 3 parametre.

for(startværdi ; betingelse ; hvad-skal-der-sker-ved-hver-gennemgang).

Opgave 1.0

Det var lige kort om løkker, selvfølgelig kan man bruge løkker til andet end at udskrive en talrække, så derfor kommer der her en øvelse til dig.

Du skal lave et array med 10 forskellige navne og bruge en for løkke til at udskrive dem alle sammen.

Jeg skriver løsningen i slutningen af artiklen, men vil helt klart råde dig til selv at lave det, da det er en rigtig god måde at lære på.

Nu skal vi igang med at lave funktioner og bruge javascripts indbyggede funktioner.

FUNKTIONER

Funktioner i javascript kan spare dig for en masse arbejde, der findes et hav af funktioner i javascript, og ellers kan du lave dine egne.

Jeg vil langt fra komme ind på alle funktionerne i javascript, men der findes en masse hjemmesider på nettet hvor du kan læse om dem.

Et eksempel på en funktion i javascript kunne være "Math.random()" som genererer et tilfældigt tal mellem 0 og 1.

```
<script type="text/javascript">
  var tal = Math.random();
  document.write(tal);
</script>
```

For at få funktionen til at generere et tilfældigt tal mellem 0 og 10 f.eks kommer den til at se lidt anderledes ud.

Nemlig sådan her:

```
<script type="text/javascript">
  var tal = Math.random()*10;
  document.write(tal);
</script>
```

Og for at få alle de frygtelig decimaler væk, kan vi bruge endnu en funktion som hedder "Math.floor()" som afrunder et tal.

eks

```
<script type="text/javascript">
  var tal = Math.random()*10;
  tal = Math.floor(tal);
  document.write(tal);
</script>
```

Et andet eksempel på en funktion kan være substring() som kan skære i en streng efter x antal bogstaver.

```
<script type="text/javascript">
  var tekst = "Hej med dig!";
  tekst = tekst.substring(0, 5);
  document.write(tekst);
</script>
```

Først sætter vi en variabel med værdien "Hej med dig!".

Så bruger vi funktionen på vores variabel - det gør vi ved at skrive "variabel.funktion".

I funktionens parametre angiver vi først starttegnet og derefter sluttegnet.

Så vores lille script vil udskrive "Hej m".

Så fik du lidt indblik i hvordan funktioner virker, så nu skal vi igang med at lave vores egen funktion. Vores funktion skal trække 2 tal fra hinanden - men den skal altid trække det mindste fra det største. Her kommer koden til funktionen, så kommer der lidt forklaring bagefter.

```
<script type="text/javascript">
var facit
function vores_funktion(x, y){
  if(x<y){
    facit = y-x;
```

```
    return facit;
  } else {
    facit = x-y;
    return facit
  }
}
</script>
```

Nu begynder vores koder at blive lidt større, men det gør ikke noget så længe man forstår dem, så det vil jeg prøve at få dig til nu.

Allerførst har vi skrevet "function vores_funktion" det gør vi for at fortælle at vi skal lave en funktion "vores_funktion" er så navnet på vores funktion.

Så har vi skrevet "(x, y)" det er vores funktions parametre, vores har så 2 parametre. (2 tal).

Så laver vi et tjek på hvilket tal der er størst, så vi er sikre på at den trækker det mindste fra det største. Og så har vi skrevet "return facit;" det er det vores funktion returnerer, altså variabelen facit.

For at kalde vores funktion kan vi gøre det på denne måde

```
<script type="text/javascript">
  var resultat = vores_funktion(12, 8);
  document.write(resultat);
</script>
```

Det bruger vi vores funktion på tallene 12 og 8.

```
<html>
<head>
<script type="text/javascript">
var facit
function vores_funktion(x, y){
  if(x<y){
    facit = y-x;
    return facit;
  } else {
    facit = x-y;
    return facit
  }
}
</script>
</head>
<body>
<script type="text/javascript">
var resultat = vores_funktion(12, 8);
document.write(resultat);
</script>
</body>
</html>
```

Ganske simpelt.

Men nu tror jeg vist det er på tide at du laver en ny opgave ;)

Opgave 1.1

Denne gang skal du lave en funktion som altid gør et tal positivt, lige meget om tallet er under eller over 0. Altså en funktion der tager den numeriske værdi af tallet.(|tal|)

Det var en lille gennemgang omkring funktioner i javascript, håber det er forklaret godt nok :)

FACIT LISTE

Løsning på Opgave 1.0

```
<script type="text/javascript">
var navne = new Array();
  navne[0] = "Peter";
  navne[1] = "Klaus";
  navne[2] = "Kim";
  navne[3] = "Ditte";
  navne[4] = "Lotte";
  navne[5] = "Mik";
  navne[6] = "Louise";
  navne[7] = "Iben";
  navne[8] = "Mikkel";
  navne[9] = "Lars";

for(var i=0; i<=9; i++){
  document.write(navne[i]);
}
</script>
```

Løsning på opgave 1.1

```
<script type="text/javascript">
var facit
function numerisk_vaerdi(tal){
  if(tal<0){
    facit = -tal;
    return facit;
  } else {
    facit = tal;
    return facit
  }
}
</script>

<script type="text/javascript">
var resultat = numerisk_vaerdi(-12);
document.write(resultat);
</script>
```

Det var alt for denne gang, jeg håber du har lært noget af denne artikel.
Du kan se nogle forskellige javascrpts jeg har lavet på <http://akmedia.dk/cases>

Mvh. ak

SVAR PÅ KOMMENTARER

Tingholm: Ja det er rigtigt, men som jeg skriver i introen så ville jeg lave en artikel som tog det helt fra bunden.

Webudvikleren: Det er rigtigt - men det er med vilje da jeg ellers også skulle til at forklare hvad "length" gør - og det syntes jeg ikke passede ind der :-)

Mclemens: Ideen med denne artikel er at give læseren den nødvendige viden til at skrive javascrpts, altså lære syntaksten og logikken bag javascrpts.
Jeg er pt. igang med en ny artikel hvor jeg vil gennemgå de mest basale events (onmouseover, onmouseout, onclick osv...)
Jeg syntes at artiklen ville blive for lang og "sej" at læse hvis man var begynder og skulle lære det hele på én gang :-)

Har gjort artiklen gratis

Kommentar af ismand d. 25. Apr 2006 | 1

Lidt tam. I mine øjne er der for meget "talesprog" og for lidt tankeproces. Blot fordi det er en begynderartikel, betyder det ikke at der ikke kan stilles krav til sine læsere -- De læser den jo som bekendt for at lære noget.

Kommentar af olebole d. 22. Apr 2006 | 2

'document.write' og 'innerHTML' ryger ikke ud af JavaScript. De har aldrig været en del af ECMA-standard (som er den standard, der definerer JavaScript) - og heller ikke af andre standarder. De hører begge til under browserens proprietære ECMA-DOM bindings. Derudover savner jeg det aller vigtigste og mest grundlæggende: At JavaScript er objekt-baseret - og hvad det betyder for sproget og dets anvendelse ;o)

Kommentar af tingholm d. 20. Apr 2006 | 3

som titlen siger... START... dette er en ok artikel hvis du ikke har skrevet php, java eller noget som helst andet... hvis du har skrevet andet kode før kan du lære indholdet af artiklen på 5 linier i stedet...

Kommentar af coderdk d. 04. Jan 2007 | 4

Fin artikel til n00bs :) I øvrigt synes jeg du bør indentere koden ;)

Kommentar af virus d. 12. Sep 2006 | 5

1.1 Math.abs(-123)

Kommentar af nubi19 d. 21. Apr 2006 | 6

Jeg er ikke enig tingholm, selvom du kender andre sprog kender du ikke nødvendigvis javascrpts syntax (omend den ligner f.eks. php meget, og C endnu mere) eller for den sags skyld de forskellige kommandoer,

etc.

Jeg kan allerede javascript, men hvis jeg ikke kunne ville jeg helt givet kunne efter denne artikel, så jeg mener det er et rigtig godt stykke arbejde du har gjort webstuff. Du kommer omkring alle de basale men uhyre vigtige ting, og gør det på en god og nem forståelig måde - helt klart værd at læse inden man går i krig med javascript.

Kommentar af dustie d. 29. Nov 2006 | 7

Kommentar af _basil d. 18. Apr 2006 | 8

Super introduktion til JS

Kommentar af webudvikleren d. 18. Apr 2006 | 9

En hurtig gennemgang af artiklen - det ser godt ud.

```
Men hvorfor lave for(var i=0; i<=9; i++){  
når du kan tælle med "length"  
for(var i=0; i<foobar.length(); i++){
```

ellers rigtig fin gennemgang af emnet :-)

Kommentar af mclemens d. 22. Sep 2006 | 10

- - - Hvis du ikke kender noget til javascript - ligesom overskriften siger... Så kan artiklen varmt anbefales - som artikel til begyndere er den meget god... Skal dog sige at jeg kender javascript og synes som webudvikleren hentyder til - godt, at artiklen kunne have berørt emner som array (og strings) lidt dybere - der er intet ved at lære javascript, hvis man ikke lærer nok til at lave dynamiske effekter... Der er ikke noget om muligheden for at få fat i objekter og manipulere med dem... Der bliver også brugt document.write i hele artiklen - jeg ved dog ikke om den ryger ud sammen med innerHTML her senere, men alt i alt en fin begynder artikel som varmt kan anbefales... (og ja, jeg ved godt det måske kan virke forvirrende med shorttags, style defineringer onmouseover, out, click, focus o.s.v. men uden alle disse funktioner hvorfor så bruge javascript - alt det andet der skal bruges på de måder der er nævnt i artiklen kunne (næsten) ligeså godt laves med f.eks. php... Der er ikke nævnt 1 eneste dynamisk effekt baseret på brugerens handling... og det er lidt trist når man starter med denne introduktion: [... Det vil sige at Javascript ikke kan kommunikere med serveren ligesom PHP og ASP kan, det kan derimod kommunikere med Klienten/Browseren. ...] { relaterer til kommentarer... - - - Jeps, forstod godt konceptet med artiklen og skrev også, at hvis man ikke kendte javascript, så kunne den varmt anbefales (tilføjet ordet varmt nu så det ikke misforståes) - - - Ok, webstuff lyder godt du er ved at kigge på en artikel med events :)

Kommentar af madsing d. 19. May 2006 | 11

Rigtig god intro til Java, opgaver så man "tvinges" til at afprøve sine nye færdigheder. Super nice!

Kommentar af phil-profil d. 05. Jan 2007 | 12