# Javascript

Javascript is een scripttaal binnen de browser. (cliëntside) Dit is iets heel anders dan Java. Java wordt uitgevoerd op de laptop/pc van de gebruiker.

Dit zijn de belangrijkste verschillen tussen Java en JavaScript:

* Java is een OOP-programmeertaal terwijl JavaScript een OOP-scripttaal is.
* Met Java worden applicaties gemaakt die in een virtuele machine kunnen worden uitgevoerd. JavaScript-code kan alleen in een browser worden uitgevoerd.
* Java-code moet worden gecompileerd, terwijl JavaScript-code enkel uit tekst bestaat.
* Voor beide zijn andere plug-ins nodig.

Kijk naar de structuur van de onderstaande Javascript code. Het woord function wordt gebruikt om aan te geven dat het om een functie gaat. Een functie doet iets, het voert een serie opdrachten uit. Een functie kan van alles zijn: optellen, aftrekken, zoeken, bereken rente bedrag etc.  
  
De naam van de functie is HelloWorld. De twee haakjes () daarin kunnen waarden worden meegegeven. De accolades { } geven het begin en eind aan van functie. Daartussen staat de instructie. De functie staat daarin het HEAD gedeelte van de code, en 'wacht' tot hij uitgevoerd moet worden. Als er op de link wordt geklikt wordt de functie geactiveerd.

**<HTML>  
<HEAD>  
<TITLE></TITLE>  
<SCRIPT language="Javascript">  
  
function HelloWorld()**  
**{  
 alert ("Hello World!");  
}  
  
</SCRIPT>  
  
</HEAD>  
<BODY>  
  
<A HREF="javascript:HelloWorld()">Klik Hier</A>  
</BODY>  
</HTML>**

Door verkeerd hoofdletter gebruik kun je errors krijgen. Bijvoorbeeld MijnVariabele bestaat niet terwijl het mijnvariabele is. Voordat we verdergaan moeten we nog even iets duidelijk maken. Javascript en Java zijn twee compleet andere dingen. Wat we net deden is Javascript, GEEN Java. Java is iets heel anders. Het enige wat deze twee gemeen hebben zijn de letters Java in hun naam. Lang geleden maakte Netscape een nieuwe taal voor in hun browser, ze noemden het Livescript. Rond die tijd won de Java programmeertaal aan populariteit waarop de heren van Netscape de naam van hun taal veranderden in Javascript om mee te profiteren van dit succes. Sindsdien worden deze twee voortdurend door elkaar gehaald. Om nog meer verwarring te zaaien besloot Microsoft, dat ontdekt had dat Javascript een zeer handige taal was, Javascript ook in hun browser op te nemen. Zij noemden het echter Jscript.

De verwarring gaat nog verder. Niet alleen zijn er verschillen tussen Netscape's Javascript en Jscript van Microsoft, er zijn ook verschillen in de ondersteuning van deze talen, afhankelijk van hoe actueel je browser is. Gelukkig hebben we ook nog goed nieuws. De Javascripts die je hier krijgt zijn basisstof en deze worden ondersteund door het merendeel van de browsers. ALLE scripts die wij hier in deze oefeningen behandelen zijn door de grote browsers uit te voeren.

Verder nog iets heel belangrijks: **Javascript is HOOFDLETTERGEVOELIG.** De functie HelloWorld is voor Javascript niet hetzelfde als helloworld. Denk hier steeds goed aan!  
  
Variabelen & Alert  
  
Wat is een variabele? Stel: x=5

x is een variabele, en de waarde ervan is in dit geval 5. Kan x ook iets anders zijn? Natuurlijk, x=6. Nu is x gelijk aan 6 (vandaar de naam variabele, de waarde ervan kan variëren). Kan een variabele iets anders zijn dan een nummer? Ja, bijvoorbeeld x="Bart". Nu is x gelijk aan het woord "Bart".

Kan de naam van een variabele iets anders zijn dan een letter zoals x? Simpel: mijnnaam="Bart" Hoe kunnen we de variabele opnemen in een functie?

<HTML>  
<HEAD>  
<TITLE></TITLE>  
  
<SCRIPT language="Javascript">  
  
function HelloWorld()  
{  
 **var mijnnaam = "Bart";  
 alert ("Hallo: " + mijnnaam);**  
}  
  
</SCRIPT>  
  
</HEAD>  
<BODY>  
  
<A HREF="javascript:HelloWorld()">Klik hier</A>  
  
</BODY>dr  
</HTML>

Bij een tekst gebruik je aanhalingstekens. Bij variabelen gebruiken, gebruiken we geen aanhalingstekens.   
(zie: alert-methode)

## De promptbox

De mogelijkheden van Javascript zijn indrukwekkend. Je kunt er veel richtingen mee op. We zullen proberen om zo veel mogelijk basismogelijkheden te laten zien en ze uit te leggen. Laten we een zogenaamde promptbox toevoegen om eerst een naam te vragen en vervolgens een persoonlijke groet te laten verschijnen. De promptbox is een soort alert maar dan met een invulveld. De prompt box heeft een parameter voor de vraag en de standaardwaarde van het invulveld. Dit staat tussen de haakjes van de Prompt-methode.

<script language="Javascript">

function HalloPrompt()  
{  
 mijnnaam = prompt("Wat is je naam?", "");  
 alert ('Hallo 5' + mijnnaam);  
}

</script>

Een functie kan ook waarden meegeven. Hieronder een voorbeeld:

<script language="Javascript">

function HalloPrompt(naam)  
{  
mijnnaam = prompt  
alert ('Hallo ' + naam);  
}  
</script>  
  
<body>  
<A HREF="javascript:HalloPrompt ('Bart')">Zeg hallo tegen Bart</A>  
<BR><A HREF="javascript:HalloPrompt ('Mathieu')">Zeg hallo tegen Mathieu</A>  
<BR><A HREF="javascript:HalloPrompt ('Bram')">Zeg hallo tegen Bram</A>  
</body>

Zie je wat er hier gebeurd? We zijn waarden aan het doorgeven aan de functie. **naam** is een variabele die gebruikt is in de functie. Als we Bart, Matthieu of iets anders doorgeven aan de functie, wordt dit gebruikt op de plaats van de variabele **naam**. Dit meegeven van variabelen is erg handig. Je kunt functies één keer schrijven en duizenden keren gebruiken, waarbij je steeds een ander resultaat krijgt omdat de variabele verschilt.

## Javascript opdracht 1

Maak een webpagina en voeg drie afbeeldingen in.  Maak het zó, dat er als je op een afbeelding klikt, een alertbox verschijnt met de melding: "Je klikte op 1" (of 2, of 3).

**C**ombineer de promptbox met de laatste oefening, zodat bij elke klik op een afbeelding een naam wordt gevraagd en zorg dat er vervolgens een melding als "Hallo Bart, je klikte op 1" verschijnt.

**Tip bij cancel**

naam = prompt("Wat is je naam?", "");  
cancel = "geannuleerd";  
if (naam==null)   
{  
 naam = cancel;  
 }

alert ('Hallo ' + naam);

90% van wat je hier zult leren zal van dit soort oefeningen afkomstig zijn. Het is belangrijk om deze trainingen goed te benutten. Als je elk van deze oefeningen maakt, dan zal je het Javascript steeds beter gaan begrijpen. Met dit in gedachten gaan we naar de volgende oefening.

## Events en berekeningen

We spreken van een event als er iets gebeurd. Bijvoorbeeld een klik met de muis, of wanneer je ergens overheen gaat met de muis, of gewoon zodra de pagina wordt geladen.

Met een **onLoad**-event in de body van onze code kunnen we een alert box laten verschijnen zodra de pagina wordt geladen.

<BODY onLoad="HelloWorld()">

Veel mensen gebruiken **onClick** of **onMouseOver**-events voor links. Hier is hetzelfde voorbeeld met een paar kleine veranderingen:

**<a href="#" onMouseOver="EenFunctie ()">Hello</a>**

**<input type="button" value="Knop" onClick="EenFunctie()">**

Er bestaat ook een onMouseOut-event. De functie werkt niet totdat de muis van de link af gaat.

**Wat betekent de** **; ?**  
De puntkomma moet gebruikt worden aan het eind van elke instructie. Hij vertelt de browser waar een instructie eindigt, en waar de volgende begint. Je kunt ook meerdere instructies op één regel plaatsen, gescheiden door een puntkomma:

instructie1;  
instructie2;  
instructie3;

is hetzelfde als

instructie1; instructie2; instructie3;

**Berekeningen**

Kopieer dit en laat het uitvoeren:

<HTML>

<HEAD>

<TITLE></TITLE>

<SCRIPT language="javascript">

function Rekensom()

{

 var eerstegetal = prompt("Voer een getal in", "");

 var tweedegetal = prompt("Voer nog een getal in", "");

 var  totaal = eerstegetal \* tweedegetal;

   alert (eerstegetal + " x " + tweedegetal + " = " + totaal);

}

</SCRIPT>

</HEAD>

<BODY>

<A HREF="javascript:Rekensom()">Klik hier</A>

</BODY>

</HTML>

**De functie** vraagt eenvoudigweg twee getallen en telt deze bij elkaar op. Denk erom  
dat in de meeste programmeertalen \* wordt gebruikt om te vermenigvuldigen en / voor delen.

Misschien gebeurt er dit: 3 + 6 = 36? Dit komt doordat de getallen als strings (reeksen van letters of cijfers) worden gezien. Als je de opdracht + geeft, worden deze strings met Javascript gewoon achter elkaar gezet. Als we de getallen eerst met 1 vermenigvuldigen, zal de computer ze als getal gaan zien (Integer) in plaats van als string. De uitkomst zal dan wel goed worden.

**var totaal = (eerstegetal \* 1)  + (tweedegetal \* 1);**

Dit probleem kan opgelost worden met de Javascript functie Parseint();

**var getal1 = parseInt("10");**

**of**

**var getal1 = parseInt(prompt("Getal invoeren", ""));**

Deze code zet string(tekst) om naar integer (getallen)

Rekenkundige operatoren komen vaak overeen bij verschillende programmeertalen:

  12 + 2     12 plus 2

  12 - 2     12 min 2

  12 \* 2     12 keer 2

  12 / 2     12 gedeeld door 2

De gebruikelijke volgorde van rekenen wordt gehanteerd:   
  
6 + 3 \* 4 = 18   
  
Het is niet 36 want vermenigvuldigen en delen gaan voor optellen en aftrekken. Als we 6 bij 3 willen optellen, en vervolgens pas het resultaat met 3 vermenigvuldigen, dan moeten we haakjes gebruiken:   
  
(6 + 3) \* 4 = 36

## Objecten properties en methoden

Een object in Javascript is een venster, een frame, een afbeelding, een textbox, het document zelf, etc.

Een property is een eigenschap van een object: de locatie van een document, de achtergrondkleur van een document, de breedte van een afbeelding, enz.

Een methode is in principe een functie. Een methode *doet* iets. **close()** is bijvoorbeeld een methode. **window.close()** sluit een venster.

Wat kunnen we doen met objecten en eigenschappen?

**Dit is een HTML formulier:**

<FORM NAME="mijnform">

<INPUT TYPE="text" NAME="mijntextbox" VALUE="">

</FORM>

**Kijk mee: we hebben een browservenster...**

window

**In het venster hebben we een document...**

window.document

**In het document hebben we een formulier...**

window.document.mijnform

window.document.mijnform.input

En de invoer heeft een waarde...

window.document.mijnform.input.value

Op dit moment is de waarde niets, maar niets is ook een waarde. (het veld is namelijk leeg)

Dit is een hiërarchie tussen objecten. Op deze manier verwijs je naar verschillende eigenschappen van objecten in je pagina. Je kunt het hiermee vergelijken:

wereld.land.provincie.stad.straat.huis.persoon.naam

Dit verwijst naar de naam van een persoon. Zijn lengte kan ook:

wereld.land.provincie.stad.straat.huis.persoon.lengte

Dit is de manier waarop de eigenschappen van de objecten op je webpagina kunt veranderen. Laten we teruggaan naar de textbox die we eerder zagen, de HTML-code ervan was:

<FORM NAME="mijnformulier">

<INPUT TYPE="text" NAME="mijntextbox" VALUE="">

</FORM>

We kunnen verwijzen naar de waarde van de textbox met:

window.document.mijnformulier.input.value

Een document kan natuurlijk meer dan één formulier hebben. Om ze uit elkaar te houden, geven we ze namen. Het formulier hierboven is genaamd **mijnformulier**. Hetzelfde geldt voor inputs van een formulier. Een formulier kan meerdere inputs hebben. Daarom geven we ook inputs een naam (in dit geval is er maar één input, genaamd mijntextbox).

Als we nu de naam van het formulier en de naam van de input gebruiken, kunnen we de waarde van de box oproepen:

**window.document.mijnformulier.mijntextbox.value**

## Javascript opdracht 2: Formulier, Prompt en Alert.

#### A.. Maak een HTML pagina met daarin een Javascript. Zorg ervoor dat er twee variabelen worden ingelezen via een Javascript prompt. Vervolgens worden deze twee getallen met elkaar vermenigvuldigd. Het resultaat wordt in een alert weergegeven.

Daarna moet dat getal met een ander getal worden vermenigvuldigd en op deze manier in  
een alertbox worden weergegeven: "5 \* 3 = 15".

Maak de pagina zo, dat nadat het antwoord in de alertbox wordt gegeven, een ander getal wordt gevraagd en wordt toegevoegd aan het vorige antwoord. Een nieuwe alertbox moet verschijnen met daarin het nieuwe antwoord: (5 \* 3) + 2 = 17

**B.**

Maak een HTML pagina met daarin een Javascript. Zorg ervoor dat er twee variabelen ingelezen kunnen worden d.m.v. een HTML formulier met twee invoervelden. Zodra er op de HTML button van het formulier geklikt wordt, wordt een Javascript gestart wat de twee getallen vermenigvuldigd en het resultaat op het scherm afdrukt d.m.v. een Alert. Maak bij deze opdracht gebruik van functies.

Javascript voorbeeld:  
<http://www.ictacademie.info/javascriptform.html>

Met Javascript kunnen we ook de achtergrond van een document veranderen. Kijk hier eens naar:

<HTML>

<HEAD>

<TITLE></TITLE>

<SCRIPT language="Javascript">

function Achtergrondkleur(mijnkleur)

{

window.document.bgColor = mijnkleur;

}

</SCRIPT>

</HEAD>

<BODY>

<A HREF="javascript:Achtergrondkleur('red')">Rood</A>&nbsp;&nbsp;&nbsp;

<A HREF="javascript:Achtergrondkleur('green')">Groen</A>&nbsp;&nbsp;&nbsp;

<A HREF="javascript:Achtergrondkleur('blue')">Blauw</A>

</BODY>

</HTML>

**bgColor** is een eigenschap van het document. Verander in de code de achtergrondkleur en ontdek dat je altijd de juiste achtergrondkleur in de alertbox krijgt. Verwijder deze regel uit de code en zie wat er gebeurd. Kijk nog eens naar de code zodat je goed weet wat er gebeurd.

## Javascript opdracht 3: Kleuren

**A**

Voeg aan het bovenstaande script drie kleuren toe waar je uit kan kiezen. Zorg er daarnaast voor dat de originele kleur wordt bewaard, zodat die als zevende kleur kan worden gekozen als de standaardkleur.

In de vorige (en komende) uitwerking(en) stond het script **na** de <BODY> regel. De reden hiervoor is simpel. Sommige browsers laden de achtergrondkleur pas na deze regel. Als je het script hiervoor plaatst, kan de kleur dus niet worden geladen en dan zal het script niet goed werken. Hieruit blijkt dat het altijd verstandig is om je scripts op meerdere browsers te testen. Het is gebruikelijk om het script in het <HEAD>gedeelte te plaatsen, maar dat hoeft niet, en in dit geval is het dus beter om het in het <BODY>gedeelte te doen.

**B:**Zet op de pagina 4 afbeeldingen van verschillende kleuren. De kleurcodes moeten in de plaatjes staan. Zorg ervoor dat de achtergrondkleur van de pagina in deze kleuren verandert zodra je met de muis over een plaatje gaat. (onMouseover= “functienaam()”;) Als de muis weer van het plaatje af gaat, moet de achtergrondkleur weer in de standaard veranderen. (onMouseout= “functienaam()”;)

**Voorbeeld code:**

<http://ictacademie.info/javascriptachtergrondkleur.html>

## Javascript Opdracht 4 Javascript Calculator

Maak een rekenmachine met een plus/min en vermenigvuldig en delen knop. Maak daarbij gebruik van 3 tekstvelden en alert het resultaat in zowel het tekstvak als in een alertbox.

Ga aan de slag met deze code: (zet dit zelf in een valide html code)

<head><title>Rekenmachine</title>

<script language="Javascript">

function optellen()

{  
// 3 variabelen zijn aangemaakt om gegevens in te kunnen opslaan.  
var eerste,tweede,resultaat;

//Neem de eerste waarde en converteer een string (tekst) naar een float, dat is een gebroken getal.  
eerste=parseFloat(document.form.getal1.value);

//Neem de tweede waarde en converteert de string naar een float (gebroken getal).  
tweede=parseFloat(document.form.getal2.value);

// Dit is het resultaat.  
resultaat=eerste+tweede;

//Alert van het resultaat.  
alert("Het resultaat is: "+resultaat);

// Show het resultaat in een tekstvak.  
document.form.antwoord.value=resultaat;  
}

</script></head>

<form name="form" method="post" action="">  
<input type="text" name="getal1" id="getal1">  
<input type="text" name="getal2" id="getal2">  
<input type="text" name="antwoord" id="antwoord">  
<input type="button" name="plus" id="plus" value="+" onClick="optellen()">  
</form>

Maak zelf 3 extra functies aan waarmee je kunt aftrekken en vermenigvuldigen en delen.  
In de body plaats je de “form code” en zorg je ervoor dat de buttons gekoppeld zijn aan de functies die je hebt aan gemaakt. Je kunt de tekst en invoervelden in een tabel plaatsen om ze netjes uit te lijnen.

**Let op: geef de input een andere naam als de functie, dus: optellen en plus.**

## If & then statement

Het principe: als iets zo is, doe dan iets:

if (voorwaarde)  
{  
 doe iets;  
}

**Voorbeeld:**

<script language="Javascript">  
  
function Hallo()  
{  
 x = 6;  
  
 if ( x > 4)  
 {  
 alert("x is groter dan 4!");  
 }  
}  
  
</script>  
  
<body>  
<a href="javascript:Hallo()">Klik hier</a>  
</body>

Een **if-then** statement is in principe makkelijk te begrijpen. Het is ook een van de belangrijkste. Dit simpele commando (en zijn variaties) is het brein van je programma's. Je kunt de computer er keuzes mee laten maken.

Een uitbreiding van het **if** statement is het **if-else**

Het principe: als iets zo is, doe dan iets. Zoniet, doe iets anders:

Ons eerdere voorbeeld maar nu iets uitgebreid. We stellen x=3 :    
  
function Hallo()  
{  
 x = 3;  
  
 if ( x > 4)  
 {  
 alert("Hey, x is groter dan 4!");  
 }  
 else  
 {  
 alert("Hey, x is NIET groter dan 4!");  
 }}  
</script>

Het if-then statement in de praktijk

<SCRIPT language="Javascript">  
  
function Hallo()  
{  
 x = prompt("Voer een getal in","");  
  
 if (x == 6)  
 {  
 alert("Wow! " + x + " is gelijk aan 6!");  
 }  
 else  
 {  
 if ( x > 4)  
 {  
 alert("Hey, " + x + " is groter dan 4!");  
 }  
 else  
 {  
 alert("Hey, " + x + " is niet groter dan 4!");  
 }  
 }  
 }  
</SCRIPT>

Eerst wordt gekeken of x gelijk is aan zes. Als dat zo is, krijgen we een bericht. Als dit niet zo is, treedt een tweede **if-then** statement in werking om te testen of x groter is dan 4 of niet.

Waarom gebruiken we in deze code een dubbel is gelijk-teken: 'x == 6' ? Het verschil zit hierin:

x = 6 de waarde van x is nu 6 (opdracht)  
x == 6 x wordt vergeleken met 6 (test op gelijkheid)

Nu we hier toch mee bezig zijn... Dit is een overzicht van de mogelijkheden:

x > 6 x is groter dan 6  
x < 6 x is kleiner dan 6  
x >= 6 x is groter dan of gelijk aan 6  
x <= 6 x is kleiner dan of gelijk aan 6  
x != 6 x is niet gelijk aan 6

Javascript Opdracht 5:

A  
Maak een webpagina die het volgende doet: Wanneer je op een link klikt, komt er een promptbox die om een getal vraagt. Als het getal lager dan 100 is moet er een alertbox verschijnen die zegt wat het getal was... "Je getal is 28". Is het getal groter dan 100, dan moet er een alertbox komen met de melding "Je getal is te groot voor me." Als het getal exact 100 is, toon dan een alertbox met de tekst: "Bingo, je getal is exact 100."

Pas de laatste opdracht zo aan, dat er een getal wordt gevraagd via een input/textbox in plaats van een promptbox.

B  
- Verander de achtergrondkleur naar **ORANJE** als het cijfer lager dan **100** is.   
- Is het getal groter dan **100**, dan moet de achtergrondkleur **GROEN** worden.   
- Als het getal gelijk is aan **100** moet de achtergrond **ROOD** worden en komt er een alertbox met de melding 'Bingo!'.   
- Als het getal groter is dan **500**, maak de achtergrond dan **BLAUW** en laat een promptbox verschijnen die om een lager getal vraagt en alles opnieuw laat beginnen.   
  
Tip: om opnieuw te beginnen moet de functie zichzelf op gaan roepen.

Deze laatste opdracht is echt programmeren, je moet er flink logisch voor na gaan denken. Dit is heel anders dan het kopiëren en plakken van Javascripts.

## De while opdracht

while (voorwaarde)  
{  
 doe iets;  
}

Bekijk het volgende...

<script language="Javascript">  
  
function Toevoeger()  
{  
getal = 1;

 while (getal < 5)  
{  
alert(getal + " is minder dan 5");  
**getal = getal + 1;**  
}  
}  
  
</script>  
  
  
<body>  
<A HREF="javascript:Toevoeger()">Klik hier</A>  
</body>

Bekijk dit voorbeeld goed. Het komt regelmatig voor dat je in je script een getal met 1 wilt verhogen of verlagen. Daarom is het handig om te weten dat: getal = getal + 1; ook kan worden geschreven als

getal++;

Op dezelfde wijze is

getal--;

hetzelfde als

getal = getal - 1;

Je komt dit veel tegen in Javascript, en ook in veel andere programmeertalen.

## Javascript Opdracht 6

Voordat je aan deze opdracht begint moet je het voorgaande goed gelezen hebben. We gaan met deze opdrachten leren gebruik te maken van de: IF functie, While en For loop.

**A: IF**

Maak een webpagina die het volgende doet: Wanneer je op een link klikt,

komt er een promptbox die vraagt om je leeftijd. Als het getal lager dan **18** is moet er

een alertbox verschijnen die zegt: “je bent nog een kind”. Is het getal groter

dan 18, dan moet er een alertbox komen met de melding "Je bent een volwassen persoon”. Als het getal exact 18 is, toon dan een alertbox met de tekst: “Je bent net volwassen geworden, gefeliciteerd”.

**Voorbeeld IF:** <http://www.ictacademie.info/javascriptif.html>

**B: While**

Maak een HTML website met daarin een Javascript dat via een Javascript Prompt 10x een getal inleest. Je dient gebruik te maken van een while lus. Bij elke invoer van een getal staat er een teller bij de Javascript Prompt. De eerste keer staat de teller op 1, de tweede keer op 2 enz. Nadat de 10 getallen zijn ingelezen worden ze bij elkaar opgeteld en wordt het resultaat getoond in een Alert.

**Let op:** **tekst moet tussen aanhalingstekens en variabelen niet! Met een + kun je tekst en variabelen aan elkaar kopppelen.**

**Voorbeeld While:** <http://www.ictacademie.info/javascriptwhileloop.html>

**C: FOR**

Herhaal opdracht B maar maak nu gebruik van een for-lus.

**Voorbeeld FOR:** <http://www.ictacademie.info/javascriptforloop.html>

## Javascript Opdracht 7

Maak een HTML website met daarin een Javascript. In de HTML site zet je een button door middel van een HTML formulier. Zodra je op de button klikt wordt er een Javascript gestart. Het script dat gestart wordt detecteert je browser en kan je vinden onder aan de opdracht.

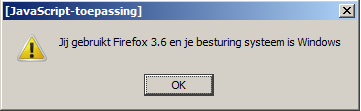
Zorg ervoor dat als men op de HTML button uit het formulier drukt het script gestart wordt en dat er vervolgens een alert wordt uitgevoerd die het resultaat op het scherm schrijft. Gebruik de onderstaande code van het alert om een Javascript Alert te krijgen zoals hieronder staat. Hiervoor maak je een functie aan.

Een voorbeeld van een functie hier te vinden…

<http://www.ictacademie.info/javascriptfunctie.html>

In het voorbeeld staat een foutje in Internet Explorer, los dit op. Zodat het systeem ook in Internet Explorer goed werkt. Internet Explorer is namelijk geen Mozilla ;)

alert("U gebruikt " + BrowserDetect.browser + " " + BrowserDetect.version + " en je besturing systeem is " + BrowserDetect.OS);



**Voorbeeld script:** <http://www.ictacademie.info/javascriptbrowser.html>

<script language="Javascript">

var BrowserDetect = {

init: function () {

this.browser = this.searchString(this.dataBrowser) || "An unknown browser";

this.version = this.searchVersion(navigator.userAgent)

|| this.searchVersion(navigator.appVersion)

|| "an unknown version";

this.OS = this.searchString(this.dataOS) || "an unknown OS";

},

searchString: function (data) {

for (var i=0;i<data.length;i++) {

var dataString = data[i].string;

var dataProp = data[i].prop;

this.versionSearchString = data[i].versionSearch || data[i].identity;

if (dataString) {

if (dataString.indexOf(data[i].subString) != -1)

return data[i].identity;

}

else if (dataProp)

return data[i].identity;

}

},

searchVersion: function (dataString) {

var index = dataString.indexOf(this.versionSearchString);

if (index == -1) return;

return parseFloat(dataString.substring(index+this.versionSearchString.length+1));

},

dataBrowser: [

{

string: navigator.userAgent,

subString: "Chrome",

identity: "Chrome"

},

{ string: navigator.userAgent,

subString: "OmniWeb",

versionSearch: "OmniWeb/",

identity: "OmniWeb"

},

{

string: navigator.vendor,

subString: "Apple",

identity: "Safari",

versionSearch: "Version"

},

{

prop: window.opera,

identity: "Opera"

},

{

string: navigator.vendor,

subString: "iCab",

identity: "iCab"

},

{

string: navigator.vendor,

subString: "KDE",

identity: "Konqueror"

},

{

string: navigator.userAgent,

subString: "Firefox",

identity: "Firefox"

},

{

string: navigator.vendor,

subString: "Camino",

identity: "Camino"

},

{ // for newer Netscapes (6+)

string: navigator.userAgent,

subString: "Netscape",

identity: "Netscape"

},

{

string: navigator.userAgent,

subString: "MSIE",

identity: "Explorer",

versionSearch: "MSIE"

},

{

string: navigator.userAgent,

subString: "Gecko",

identity: "Mozilla",

versionSearch: "rv"

},

{ // for older Netscapes (4-)

string: navigator.userAgent,

subString: "Mozilla",

identity: "Netscape",

versionSearch: "Mozilla"

}

],

dataOS : [

{

string: navigator.platform,

subString: "Win",

identity: "Windows"

},

{

string: navigator.platform,

subString: "Mac",

identity: "Mac"

},

{

string: navigator.userAgent,

subString: "iPhone",

identity: "iPhone/iPod"

},

{

string: navigator.platform,

subString: "Linux",

identity: "Linux"

}

]

};

BrowserDetect.init();

</script>

Arrays

Een rij (in het Engels een array) is een groep van gegevens, maar een rij is ook een object, en als een object heeft het bepaalde eigenschappen. Er zijn ingebouwde rijen zoals alle plaatjes op een pagina. Kijk naar de volgende code voor wat meer duidelijkheid.

<script language="Javascript">

function Sourcezien(naam)

{

window.document.mijnformulier.input.value = window.document.images[position].src;

}

function aantalplaatjes()

{

var aantal = window.document.images.length;

}

</script>

<body>

<A HREF="#" onMouseOver="Sourcezien(0); "><IMG SRC="../plaatjes/bier.gif" BORDER="0"></A>

<A HREF="#" onMouseOver="Sourcezien(1); "><IMG SRC="../plaatjes/hamburger.gif" BORDER="0"></A>

<A HREF="#" onMouseOver="Sourcezien(2); "><IMG SRC="../plaatjes/boter.gif" BORDER="0"></A>

<A HREF="#" onMouseOver="Sourcezien(3); "><IMG SRC="../plaatjes/wortel.gif" BORDER="0"></A>

<A HREF="#" onMouseOver="Sourcezien(4); "><IMG SRC="../plaatjes/kaas.gif" BORDER="0"></A>

<A HREF="#" onMouseOver="Sourcezien(5); "><IMG SRC="../plaatjes/kersen.gif" BORDER="0"></A>

<FORM NAME="mijnformulier">

<INPUT TYPE="text" NAME="input" VALUE="" SIZE=70>

</FORM>

<A HREF="javascript:aantalplaatjes()">klik hier</A> Voor het nummer van het aantal plaatjes op deze pagina

</body>

Dit laat min of meer een rij van afbeeldingen (images) zien. Deze rij afbeeldingen is een rij van alle plaatjes op de pagina, namelijk, **images[]**. De browser is niet geïnteresseerd in de naam, maar in het indexnummer van de plaatjes, dat begint bij 0. De eerste afbeelding is **images[0]**, de tweede is **images[1]**, de derde is **images[2]**, enz. De totale rij heeft zo ook zijn eigenschappen, waarvan één de lengte is. Dit heb je gezien bij de alertbox die verschijnt bij het klikken op de link en de function aantalplaatjes() activeerd. Een individueel plaatje heeft zo ook zijn eigenschappen, zoals de src (source/bron). Bij de functie Bronbekijken() wordt de bron die bij een bepaald indexnummer van de afbeelding hoort, in de waarde (value) van het formulier gezet. Zo krijg je de bron in de tekstbalk.

Hoewel deze code groot is, is het niet verschrikkelijk moeilijk. Kijk goed hoe het werkt, het kan best zijn dat dit niet de eerste keer lukt.

Je klikte op het plaatje met indexnr. 3 van de 6 plaatjes.

De bron is C:\map\map\burger.gif

Verder mag je niet zomaar 6 invullen. Het getal moet een variabele zijn. Hiervoor moet je gebruik maken van de 'length' eigenschap van de rij.

Een andere eigenschap van de plaatjes is de **naam**. In de code kan je aan de tag van de het plaatje een naam toevoegen zoals hier:

<IMG SRC="plaatje.gif" HEIGHT=32 WIDTH=32 NAME="plt01">

Als je dit hebt gedaan kan je dit verder in je script toepassen ( bijvoorbeeld in een functie om een bron te vinden). Je krijgt dan iets vergelijkbaars als bij het indexnummer.

window.document.images['plt01'].src

Er zijn dus twee manieren om een plaatje in een image rij aan te duiden, met het indexnummer of met de naam. Dit moet redelijk duidelijk zijn...

## Javascript Opdracht 8

Maak een nieuwe webpagina met 12 plaatjes van etenswaren. Geef ze allemaal een specifieke naam. Als je er op klikt, moet opnieuw de bron, positie en bron verschijnen in een alertbox. Gebruik de manier die je net geleerd hebt: verwijzen met naam en niet met indexnummer. Nu moet je dus weten hoe je de naam van een plaatje gebruikt voor het bepalen van de bron.

**Tip voor het totaal aantal plaatjes (als er 6 plaatjes teveel worden geteld.**

aantal = parseInt(window.document.images.length - 6);

## Lijsten

Een ander soort rij is de optielijst, de zogenaamde "drop down list" Een voorbeeld van deze lijst is de volgende...

<script language="Javascript">  
  
function DeBeste()  
{  
beste = window.document.formulier.bestevdw.selectedIndex;

alert(window.document.formulier.bestevdw[beste].value);  
}  
  
</script>  
  
<BODY>  
<FORM NAME="formulier">  
<SELECT NAME="bestevdw" onChange="DeBeste()">  
<OPTION VALUE="piet">Piet  
<OPTION VALUE="mathieu"> Mathieu  
<OPTION VALUE="bart"> Bart  
<OPTION VALUE="einstein"> Einstein  
<OPTION VALUE="bill"> Bill Gates  
<OPTION VALUE="angel">Angel ruyter  
</SELECT>  
</FORM>  
  
</BODY>

De option value is voor de waarde van de invoer. Deze kan later gebruikt worden. Wat na de tag

**<option value>** staat, komt in de lijst te staan. In het volgende script maken we gebruik van de waarde (value) die je aan opties kunt toekennen. Hiervoor maken we gebruik van een function.

In dit script hebben we gebruik gemaakt van de eventhandler **onChange**. Wanneer je een andere naam kiest in de lijst, start hij de functie "DeBeste". Als de functie "DeBeste" wordt gestart, zal deze een alertbox tonen met het indexnummer van de lijst. (deze begint met de waarde 0)

Kijk ook even naar de verwijzing **"beste = window.document.form.bestevdw.selectedIndex;"**

beste is de waarde die in de alertbox komt te staan.

* form verwijst naar de naam van het formulier.
* bestevdw verwijst naar de geslecteerde naam.

Het is natuurlijk leuker om een naam in de alertbox te zien verschijnen in plaats van een indexnummer. De selectedIndex is een eigenschap van de rij, en de waarden zijn eigenschappen van de afzonderlijke opties.

De array die we hebben heet **bestevdw**. De eerste optie is bestevdw[0], de tweede is bestevdw[1], de derde is bestevdw[2], enz. Wat we nu doen is de selectedIndex gebruiken en opslaan in de variabele **beste**. Beste is dus het indexnummer. Vervolgens kunnen we de individuele waarde van de optie oproepen met 'value'. Kijk voor de duidelijkheid naar de volgende code die lijkt op de vorige, maar de benodigde aanpassingen heeft.

Kijk goed hoe het werkt. Let vooral op hoe de verwijzingen werken.

## Locatie objecten

Een ander object dat je op zijn tijd moeten gebruiken is het **location** (locatie) object. Het location object is een eigenschap van het window (venster) object...

window.location

En net zoals andere objecten heeft het zo zijn eigenschappen. Een eigenschap die vaak gebruikt wordt is de **href** eigenschap...

window.location.href

Dit specificeert de URL van het openstaande venster.

Kijk naar het volgende script...

<HTML>  
<HEAD>  
<TITLE></TITLE>  
  
<SCRIPT language="Javascript">  
  
function Urlzien()  
{  
 alert(window.location.href);  
}  
  
function GoUrl()  
{  
 window.location.href = "anderepagina.html";  
}  
  
</SCRIPT>  
  
</HEAD>  
<BODY>  
<A HREF="javascript:Urlzien()">klik hier</A> voor de URL van deze pagina.<BR>  
<A HREF="javascript:GoUrl()">klik hier</A> om naar de andere pagina te gaan.  
</BODY>  
</HTML>

De functie Urlzien() neemt de locatie van de huidige pagina en zet deze in een alertbox. De functie GoUrl() verandert de locatie van de pagina, zodat de aangegeven pagina geladen wordt in hetzelfde venster.

Je kunt natuurlijk ook zelf een rij/array maken. We gaan hier een Array met kleuren vormen.

**Kleuren**   
rood   
blauw   
groen   
geel   
paars   
oranje

Allereerst moet je verklaren dat er een nieuwe rij aankomt. Dat doe je met de volgende code:

kleuren = new Array();

Vervolgens moet je de verschillende waarden van de rij bepalen.

kleuren = new Array();

kleuren[0] = "rood";

kleuren[1] = "blauw";

kleuren[2] = "groen";

kleuren[3] = "geel";

kleuren[4] = "paars";

kleuren[5] = "oranje";

Ook hier wordt er met het tellen begonnen bij 0. De kleurenrij hierboven heeft 6 elementen, en we refereren ze elk met een nummer, zoals bij de volgende code:

<SCRIPT language="Javascript">  
kleuren = new Array();  
kleuren[0] = "rood";  
kleuren[1] = "blauw";  
kleuren[2] = "groen";  
kleuren[3] = "geel";  
kleuren[4] = "paars";  
kleuren[5] = "oranje";  
  
function KrijgMijnKleur()  
{  
 alert(kleuren[3]);  
}  
</SCRIPT>  
  
<BODY>  
<A HREF="javascript:KrijgMijnKleur()">Klik hier om mijn kleur te weten</A>  
</BODY>  
Rijen met voorwaarden

Deze pagina gaat door op de kleurenkiezer van de vorige les, dus zorg ervoor dat je les 13 ook hebt gedaan. We laten in dit script een kleur *verzenden* naar een functie. Het lijkt hetzelfde als het laatste script van de vorige les, maar bekijk de verschillen...

<SCRIPT language="Javascript">

kleuren = new Array();

kleuren[0] = "rood";

kleuren[1] = "blauw";

kleuren[2] = "groen";

kleuren[3] = "geel";

kleuren[4] = "paars";

kleuren[5] = "oranje";

function KrijgMijnKleur(MijnKleur)

{

alert(kleuren[MijnKleur]);

}

</SCRIPT>

<BODY>

<A HREF="javascript:KrijgMijnKleur(3)">Klik hier voor mijn kleur</A>

</BODY>

We komen al verder, maar er zijn nog veel puntjes op de i te zetten bij deze scripts. Je hebt vast gemerkt dat we bij de voorwaarden het getal 5 hebben gebruikt als maximum. Als we echter een kleur toevoegen, dan krijg je het probleem dat je **ALLE** getallen moet wijzigen, omdat het anders niet meer klopt. En nu komen we weer terug op les 11, waar je hebt geleerd om de lengte van rijen te gebruiken. We maken dus van de lengte een variabele.

**Hier is een oplossing.**

We gaan weer even terug naar een vorig punt, namelijk de beperking van invoer. Het laatste script hiervan zorgt ervoor dat het getal niet groter dan 5 kan zijn. Maar we moeten ook regelen dat het getal niet kleiner kan zijn dan 0. We moeten dus een functie hebben die controleert of het cijfer groter is dan 5 of kleiner is dan 0.

Kijk naar het volgende if-statement:

if (x > 5)

{

opdracht

}

Het controleert of x groter dan 5 is. We kunnen hier een andere voorwaarde aan toevoegen en testen of x > 5 *of* x < 0. Je krijgt dus dit:

if ((x > 5)||(x < 0))

{

opdracht

}

De OR operator bestaat uit twee "vertical slashes", ook wel pipes genoemd. || (De toets hiervoor zit onder de 'enter'-knop op je toetsenbord. Hij staat boven de 'backslash').

Vergelijkbaar met de OR operator is de AND operator. Deze bestaat uit twee en-tekens: &&.

if ((x > 5)&&(x < 0))

{

opdracht

}

De code hierboven geeft aan dat x groter moet zijn dan 5 *en* kleiner dan 0. De slimme mensen onder ons zien gelijk dat dit niet kan. Hopelijk begrijp je het principe en snap je dat de AND operator heel handig te gebruiken is bij andere scripts.

## 

## Javascript Opdracht 9

Voordat je aan deze opdracht begint moet je het Array deel goed gelezen hebben. We gaan met deze opdrachten leren gebruik te maken van een Array. Een array is een variabele met meerdere waarden.

Voorbeeld: <http://www.ictacademie.info/javascriptarray.html> (zie broncode!)

**http://www.ictacademie.info/bestevan.html**

**http://www.ictacademie.info/javascript\_9.html**

<SCRIPT language="Javascript">

function DeBeste()

{

beste = window.document.formulier.bestevdw.selectedIndex

alert(window.document.formulier.bestevdw[beste].value);

}

</SCRIPT>

</HEAD>

<BODY>

<FORM NAME="formulier">

<SELECT NAME="bestevdw" onChange="DeBeste()">

<OPTION VALUE=" piet"> Piet

<OPTION VALUE=" mathieu"> Mathieu

<OPTION VALUE=" bart"> Bart

<OPTION VALUE=" einstein" Einstein

<OPTION VALUE=" bill"> Bill Gates

<OPTION VALUE=" angel"> Angel ruyter

</SELECT>

</FORM>

</BODY>

</HTML>

**Opdracht A**

Maak een webpagina met een select element. Waar de gebruiker kan kiezen. Maak daarnaast 6 verschillende webpagina's, elk voor één van de namen van *Beste van…*. Je verzint zelf 6 beste ICT ontwikkelingen. In plaats van het verschijnen van een alertbox met de waarde/value "Mathieu", moet je de waarden van de opties in de lijst veranderen in de URL van pagina's met de namen (zoals mathieu.html). Daarna moet je in een functie de waarde ophalen en deze instellen als de nieuwe "href". Als je de code goed hebt, moet je dus een lijst met 6 links hebben.

**Tip**

**window.location.href = "anderepagina.html";**

**Opdracht B**

Maak nog een selectie lijst voor het kiezen van een kleur. Je kunt natuurlijk ook zelf een rij/array maken. We gaan hier een Array met kleuren vormen.

**Kleuren**

rood

blauw

groen

geel

paars

oranje

Allereerst moet je verklaren dat er een nieuwe rij aankomt. Dat doe je met de volgende

code:

kleuren = new Array();

Vervolgens moet je de verschillende waarden van de rij bepalen.

kleuren = new Array();

kleuren[0] = "rood";

kleuren[1] = "blauw";

kleuren[2] = "groen";

kleuren[3] = "geel";

kleuren[4] = "paars";

kleuren[5] = "oranje";

Ook hier wordt er met het tellen begonnen bij 0. De kleurenrij hierboven heeft 6

elementen, en we refereren ze elk met een nummer, zoals bij de volgende code:

<http://www.ictacademie.info/kleurenarray.html>

<HTML>

<HEAD>

<TITLE></TITLE>

<SCRIPT language="Javascript">

kleuren = new Array();

kleuren[0] = "rood";

kleuren[1] = "blauw";

kleuren[2] = "groen";

kleuren[3] = "geel";

kleuren[4] = "paars";

kleuren[5] = "oranje";

function KrijgMijnKleur()

{

alert(kleuren[3]);

}

</SCRIPT>

</HEAD>

<BODY>

<A HREF="javascript:KrijgMijnKleur()">Klik hier om mijn kleur te weten</A>

</BODY>

</HTML>

Zorg er nu zelf voor dat de waarden opgehaald worden door de selectie. Kijk hierbij goed naar het script van het A deel hoe dit gebeurt. Het mooist is natuurlijk dat na het selecteren een andere achtergrond krijgt.

## Document.write

Dit is een zogenaamde HTML opmerking: **<!-- De Opmerking -->**

En dit is een zogenaamde Javascript-opmerking: **/\* De Opmerking \*/**   
Alles wat tussen de **/\*** en de **\*/** staat wordt niet vertoond op de site, en kan dus voor opmerkingen dienen over de scripts. De opmerking kan verspreid zijn over meerdere lijnen.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

- Alles wat hier staat is de opmerking

- Je kunt hier alles schrijven, want het komt niet op het beeld

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Dit is de tweede javascript opmerking: // Opmerking   
Dit is de enkele lijn opmerking. Alles wat achter de // staat wordt genegeerd.

Dit is eigenlijk een methode van het window-object en technisch gezien moet er dan ook staan ***window.*document.write()**, maar de browser is slim genoeg en weet dat documents altijd in een window staan. Daarom mag je window. weg laten. Later gaan we knoeien met windows en andere frames, en dan gaan we ons pas zorgen maken over dat probleem. Nu, als we praten over een alleenstaand document in 1 window, negeren we de window.

Het handige van schrijven in Javascript is dat je ook HTML tags kunt gebruiken.

<SCRIPT language="Javascript">

document.write("<CENTER><I><B>Hallo Javascript fan!</B></I></CENTER>");

</SCRIPT>

De document.write-methode is heel handig en leuker dan HTML. Het voordeel is dat je de pagina dynamischer kan maken. Een voorbeeld is, dat ik weet dat jij heet en je bent jaar. (Weet je nog dat je die gegevens hebt ingevoerd?)

<SCRIPT language="Javascript">

naam = "Jan";

document.write(naam);

</SCRIPT>

Het schrijft gewoon de variabele **naam** op de pagina.

## Document.write()

Wanneer je schrijft op een pagina, kan je document.write zo vaak gebruiken als je wil:

document.write("<P><B>Een gedicht voor " + naam + "...<BR></B>");

document.write("<I>Rozen verwelken,<BR>");

document.write("Schepen vergaan,<BR>");

document.write("Javascript is leuk,<BR>");

document.write("En die liefde blijft altijd bestaan!,<BR></I>");

Zoals je ziet kan je ook HTML-code meegeven aan het script. Document.write() is een veel-gebuikte code, en we zullen het ook gaan gebruiken in de volgende scripts. Er is nog iets wat je moet weten voordat we doorgaan.

## Document.writeln()

document.writeln() zorgt ervoor, dat een zin die aan het einde van de pagina komt, wordt afgebroken en op een nieuwe regel begint. Als je een heel lange zin maakt met document.write(), dan zal dat op één regel blijven en moet je horizontaal scrollen om verder te lezen. Document.writeln() zorgt dus voor een betere opmaak.

## Javascript Opdracht 10

Gebruik een formulier om een naam, klas en een toetsvraag op te vragen met radiobuttons (meerkeuze). Schrijf vervolgens met document.write() de antwoorden op het scherm met documentwrite. Controleer met een IF/Else functie de antwoorden.

**http://www.ictacademie.info/javascript\_10.html**

## Wiskunde in Javascript.

Javascript heeft ook een 'Wiskunde'-object, ofwel Math-object. Dit is een functie met veel handige toepassingen. De basiscode van dit object is **Math.method(number)**. Het ziet er misschien vrij moeilijk uit, maar dit valt mee. Kijk maar eens naar deze code, waarmee je de wortel van 9 uitrekent:

<HTML>  
<HEAD>  
<TITLE></TITLE>  
  
<SCRIPT language="Javascript">  
  
**function Wortel()  
{  
 getal = Math.sqrt(9);  
 alert(getal);  
}**  
</SCRIPT>  
  
</HEAD>  
<BODY>  
  
<A HREF="javascript:Wortel()">Klik hier</A>  
  
</BODY>  
</HTML>

Je ziet dat dit niet al te moeilijk is.

Math.round() is een andere Math-methode. Deze functie rond de ingevoerde waarde af op het dichtstbij liggende gehele getal.

Er zijn vele andere wiskundige methoden.

Math.floor() rond af naar beneden. Math.floor(3.2) is 3. Math.floor(3.9) is 3.

Math.ceil() rond af naar boven. Math.ceil(3.2) is 4. Math.ceil(3.9) is 4.

Math.round() rond alles lager dan 3.5 af op 3, en 3.5 of meer op 4.

Hoe moet je afronden op 1 decimaal nauwkeurig? Simpel: vermenigvuldig het getal met 10, rond het af op gehelen, en deel het dan door 10.

Verder bestaan de methoden Math.sin(), Math.cos() en Math.tan(), waarmee je de cosinus, sinus en tangens van een getal kunt uitrekenen. Tevens bestaat de functie Math.max(). Math.max(x,y) geeft als uitkomst de grootste van twee getallen.

alert(Math.max(2,8)); geeft een pop-up scherm met de uitkomst 8. Alert(Math.min(2,8)); zal als uitkomst 2 geven. Er zijn nog enkele andere Math-functies die we niet zullen behandelen, maar deze zijn natuurlijk terug te vinden op het internet.

## Javascript Opdracht 11

**A**

Maak een formulier waarmee wordt gevraagd om een getal in te voeren en het formulier geeft de wortel terug in een alert. De uitkomst moet een afgeronde waarde hebben. De wortel van 9 is bijvoorbeeld 3. De wortel van 10 is 3.3333 en afgerond is de uitkomst 3.

**B**

Maak daarna een script waarmee de rente van een spaarrekening berekend kan worden. Zorg dus voor de volgende velden in het formulier.

* Inleg (1000 euro)
* Rente percentage (1,2%)
* Looptijd (maanden of jaren)
* Saldo na de looptijd

[www.ictacademie.info/javascriptwortel.html](http://www.ictacademie.info/javascriptwortel.html)

[www.ictacademie.info/javascriptafronden.html](http://www.ictacademie.info/javascriptafronden.html)

**http://www.ictacademie.info/javascript\_11.html**

## Het 'for'-statement

Het 'for' statement is een veelgebruikt en nuttige functie in Javascript om herhalingen toe te passen. Hier heb je een voorbeeld:

<HTML>  
<HEAD>  
<TITLE></TITLE>  
<SCRIPT language="Javascript">  
function PopUp()  
{  
 for ( x = 0; x < 5; x++)  
 {  
 alert(x);  
 }  
}  
</SCRIPT>  
  
</HEAD>  
<BODY>  
<A HREF="javascript:PopUp()">Klik hier</A>  
</BODY>  
</HTML>

Dit is in stappen wat er gebeurt:

1. for (x = 0; x < 5; x++ ) - x begint bij 0
2. for ( x = 0; x < 5;x++ ) - Als x kleiner is dan 5, voer dan de opdracht die tussen de accolades staat uit, in dit geval: alert(x);
3. for ( x = 0; x < 5; x++) - Verhoog de waarde van x met 1 en ga naar stap 2.
4. De herhaling stopt zodra x de waarde 5 heeft bereikt. (In dit geval stopt de herhaling dus na 4, en niet na 5!)

## Javascript Opdracht 12

Schrijf een script waarbij de wortels van alle gehele getallen tussen 0 en 20 worden uitgerekend, en vervolgens in één decimaal nauwkeurig worden weergegeven in een alertbox. (knop > functie > for loop > wortels)

De wortel van 0 is 0  
De wortel van 1 is 1  
De wortel van 2 is 1.4  
etc...

**Tip:** Je kunt met deze code ervoor zorgen dat de een zin op een nieuwe regel komt: - **\n**....(De n van nieuwe lijn.) of </br>.

alert("Hand\nhand\nvingers\nduim");

Maak daarna nog een kleine verandering in de vorige opdracht, waardoor de uitkomst in twee decimalen nauwkeurig wordt afgerond.

## Formulieren in Javascript

We hebben in deze cursus al eens het onderwerp formulieren behandeld, en in deze les gaan we daar iets verder op in.

Er zijn voor het maken van bepaalde functies in Javascript verschillende mogelijkheden. Je zult tijdens het werken met Javascript zelf ontdekken wat jij de makkelijkste en handigste manieren vindt, en je zult ontdekken hoe je je script kunt verbeteren en vereenvoudigen. Indien jij een totaal andere oplossing voor een probleem vindt dan onze oplossing, zou het heel goed kunnen dat jouw manier ook een goede is. Een veel voorkomend onderdeel van een formulier is de 'radio'-knop. Bekijk hier het voorbeeld:

<HTML>

<HEAD>

<TITLE></TITLE>

<SCRIPT language="Javascript">

function Fruit()

{

alert(window.document.mijnformulier.fruit.value);

}

</SCRIPT>

</HEAD>

<BODY>

<FORM NAME="mijnformulier">

<INPUT TYPE="radio" NAME="fruit" onClick="window.document.mijnformulier.fruit.value='Sinaasappels'">Sinaasappels & Mandarijnen<BR>

<INPUT TYPE="radio" NAME="fruit" onClick="window.document.mijnformulier.fruit.value'Bananen'">Bananen<BR>

<INPUT TYPE="radio" NAME="fruit" onClick="window.document.mijnformulier.fruit.value='Peren'">Peren, Nectarines & Pruimen<BR>

Kies jouw favoriete fruit en <A HREF="javascript:Fruit()">klik hier</A>.

</FORM>

</BODY>

</HTML>

Er zijn in dit geval een paar dingen die je moet begrijpen. Er zijn weliswaar drie verschillende elementen, maar de drie knoppen hebben allemaal dezelfde naam. Daarom hebben ze samen maar één waarde. Ondanks dat ze allemaal apart kunnen worden aangewezen en veranderd, blijft het een groep en daarom worden ze gezien als één invoer.

Merk ook op dat, wanneer de pagina voor het eerst wordt geladen, de waarde "undefined" is. (Zelfs als we via HTML een knop selecteren, blijft de waarde voor Javascript "undefined") Wanneer we met de muis een van de radio-knoppen aanklikken, stelt een OnClick-functie de waarde vast. Wanneer we de link aanklikken, leest de functie de waarde die is ingesteld door de onClick-functie. Probeer eerst goed te begrijpen hoe dit werkt voordat je verder gaat.

## Het gebruik van Radio-Knoppen

We kunnen radio-knoppen selecteren en de-selecteren met behulp van de muis, maar ook met een code. De basiscode is:

button.checked = true   
  of   
button.checked = false

Checked is een eigenschap van het object radio-knop. Bekijk het volgende voorbeeld:

<HTML>

<HEAD>

<TITLE></TITLE>

<SCRIPT language="Javascript">

function Fruit(soortfruit, fruitnaam)

{

window.document.mijnformulier.fruit[soortfruit].checked = true;

window.document.mijnformulier.fruit.value = fruitnaam;

}

</SCRIPT>

</HEAD>

<BODY>

<FORM NAME="mijnformulier">

<INPUT TYPE="radio" NAME="fruit" onClick="window.document.mijnformulier.fruit.value = 'sinaasappels'">Sinaasappels & Mandarijnen<BR>

<INPUT TYPE="radio" NAME="fruit" onClick="window.document.mijnformulier.fruit.value = 'bananen'">Bananen<BR>

<INPUT TYPE="radio" NAME="fruit" onClick="window.document.mijnformulier.fruit.value = 'peren'">Peren, Nectarines & Pruimen<BR>

Om Sinaasappels te selecteren <A HREF="javascript:Fruit(0,'sinaasappels')">Klik hier</A>.<BR>

Om Bananen te selecteren <A HREF="javascript:Fruit(1,'bananen')">Klik hier</A>.<BR>

Om Peren te selecteren <A HREF="javascript:Fruit(2,'peren')">Klik hier</A>.<BR>

Om de huidige waarde af te lezen <A HREF="javascript:alert(window.document.mijnformulier.fruit.value)">Klik hier</A>.

</FORM>

</BODY>

</HTML>

Zie je wat er gebeurt? De groep knoppen is een rij.....**fruit[0]**, **fruit[1]** en **fruit[2]**, en hiernaar kan worden verwezen. De functie hierboven zegt: selecteer het eerste element van de groep, in dit geval dus de sinaasappelen. De [ ] tekens worden vaak gebruikt voor een array (variabele met meerdere waarden). Met een regel wordt de huidige waarde weergegeven in een alertbox.

## Javascript Opdracht 13

Maak een formulier met 5 checkboxen. Verzin zelf een leuke vraag met meerdere antwoord-mogelijkheden. De checkboxen moeten **geselecteerd** kunnen worden met een eigen knop. Daarnaast maak je een knop waarmee alles gereset kan worden (Niet geselecteerd)!

De tekst uit box 1 naar box 2 wordt verplaatst, die van box 2 naar box 3 en die van box 3 naar box 4. De tekst uit box 4 verdwijnt.

**Tip:** Er is geen methode om tekst uit een textbox te verwijderen, maar je kan wel de waarde laten veranderen in "".

## Meerdere variabelen mee sturen in een functie

Bestudeer het onderstaande voorbeeld om meer dan één variabele met een functie mee te sturen.

<HTML>

<HEAD>

<TITLE></TITLE>

<SCRIPT language="Javascript">

function DoeIets(ding1,ding2)

{

alert(ding1 + ding2)

}

</SCRIPT>

</HEAD>

<BODY>

<A HREF="javascript:DoeIets(2,' Rozen')">Klik hier</A>

</BODY>

</HTML>

**http://www.ictacademie.info/javascript\_13.html**

## Het gebruik van meerdere functies binnen één script.

Vaak heb je voor een goedwerkend script meerdere functies nodig. Je roept dan een functie aan, die op zijn beurt een andere functie gebruikt.

<SCRIPT language="Javascript">

function ZegHallo()

{

VraagNaam();

alert("Hallo " + mijnnaam);

}

function VraagNaam()

{

mijnnaam = prompt("Wat is je naam?", "");

}

</SCRIPT>

</HEAD>

<BODY>

<A HREF="javascript:ZegHallo()">Klik hier</A>

</BODY>

</HTML>

Zie je wat er gebeurt? De link roept de functie ZegHallo() aan, die op zijn beurt VraagNaam() aanroept. Nadat VraagNaam() is uitgevoerd, wordt er terug gekeerd naar ZegHallo() en verschijnt er een alert-venster.

Er is een belangrijke regel die je moet weten: Een variabele die je declareert in een functie, is alleen in die functie beschikbaar. Een variabele die daarbuiten en daarvoor gedeclareerd is, kan wel overal gebruikt worden.

Maar als je kijkt naar de functie VraagNaam(), zie je dat de variabele 'mijnnaam' is gedeclareerd, die vervolgens wordt gebruikt in het alert-venster van ZegHallo(). Hoe kan dat? Nou, we hebben 'mijnnaam' niet echt gedeclareerd, maar gewoon gebruikt. Om werkelijk een variabele te declareren, moet je die vooraf laten gaan met 'var'. Om het verschil te merken, moet je in het script hierboven 'mijnnaam' declareren in VraagNaam(). var mijnnaam = prompt("Wat is je naam?", "");

Dit werkt dus niet. Waarom werkte het in het eerste geval wel? Omdat de variabele niet was gedeclareerd met 'var', zag Javascript deze variabele als een globale variabele. (Een globale variabele is een variabele die overal binnen het script kan worden toegepast)

Waarom is dit belangrijk? Waarom niet gewoon alle variabelen globaal maken? In grote scripts, waar je veel gebruik maakt van variabelen, kan het voorkomen dat je variabelen in verschillende functies eenzelfde naam wilt geven. Als je wilt dat Javascript deze variabelen niet door elkaar gaat halen, is het gemakkelijker om ze met 'var' te declareren binnen de functie. Maar in het geval dat jij een kort script schrijft, is het gebruiken van globale variabelen waarschijnlijk makkelijker. Je staat dus zelf voor de keuze.

Via de onderwerpen 'Verschillende functies binnen één script' en globale variabelen/locale variabelen komen we bij een belangrijk onderdeel van Javascript: *herbruikbaarheid*. Een herbruikbare functie is een functie, die een belangrijke, korte taak uitvoert die je in verschillende andere functies kunt gebruiken. Het is dus een soort handig stuk 'gereedschap' binnen je script.

Hier volgt een voorbeeld van een herbruikbare functie. Het is de 'Willekeurige Getallen Generator'.

<HTML>

<HEAD>

<TITLE></TITLE>

<SCRIPT language="Javascript">

// Willekeurige Getallengenerator. Als max=3 dan geeft de functie 1,2 or 3

function WillekeurigGetal(max)

{

return (Math.floor(Math.random()\*max))+1;  
}

function Maakgetal()

{

mijnnummer = WillekeurigGetal(10);

alert(mijnnummer);

}

</SCRIPT>

</HEAD>

<BODY>

<A HREF="javascript:Maakgetal()">Klik hier</A>

</BODY>  
</HTML>

Elke keer wanneer we een willekeurig getal tussen 1 en een maximum nodig hebben, roepen we de functie WillekeurigGetal(maximum) aan. Als we dus een willekeurig getal willen tussen de 1 en 100, kiezen we WillekeurigGetal(100).

## Javascript Opdracht 14

Gebruik de Willekeurige Getallengenerator om een willekeurig getal tussen de 1 en 100 te verkrijgen. Als het getal lager is dan 50, zorg dan voor een alertbox met een uitvoer als: Yo, 32 is lager dan 50!, en indien het getal groter of gelijk is aan 50, zorg dan voor een uitvoer als: Hey, 72 is groter of gelijk aan 50! Voeg een **for** statement toe zodat de functie 5 keer achter elkaar herhaald wordt.

## Gemiddelde uitrekenen

Hier heb je een voorbeeld van een functie die in sommige gevallen handig kan zijn. Je voert 3 getallen in, en de functie rekent het gemiddelde uit.

function Middelaar()

{

gemiddelde = ( (getal1\*1) + (getal2\*1) + (getal3\*1) ) / 3;

}

We vermenigvuldigen alle getallen met 1, omdat de browser anders denkt dat het gewoon om een getallenrij gaat. Hij ziet dan bijvoorbeeld 13+39 aan voor 1339.

Merk op dat in deze functie alleen maar een waarde wordt berekend, maar geen uitvoer wordt gegeven. We kunnen dit veranderen door een simpele regel toe te voegen:

function Middelaar(getal1, getal2, getal3)

{

gemiddelde = ( (getal1\*1) + (getal2\*1) + (getal3\*1) ) / 3;

return gemiddelde;

}

Of nog gemakkelijker:

function Middelaar(getal1,getal2,getal3)

{

return gemiddelde = ( (getal1\*1) + (getal2\*1) + (getal3\*1) ) / 3;

## Javascript Opdracht 15

Gebruik de onveranderde functie van hierboven voor deze oefening. Maak 3 tekstboxen en een knop. Wanneer deze knop wordt ingedrukt, rekent de functie het gemiddelde uit en geeft een uitvoer als deze:

Het gemiddelde van 10, 22 en 13 is 15. Je oplossing moet dus de drie waarden nemen, deze naar Middelaar() sturen, en de uitkomst presenteren in een alertbox. Zorg ervoor dat de functie de uitkomsten afrond in twee decimalen.

**http://www.ictacademie.info/javascript\_15.html**

## Formuliergegevens controleren met JavaScript

Wanneer je interactie wilt voorzien met de bezoekers op je website ga je onvermijdelijk te maken krijgen met invoerformulieren. Helaas misbruiken vele mensen deze interactiemogelijkheid door onzinnige dingen te versturen.

Dan heb ik het nog niet eens gehad over de professionele spammers die er hun bezigheid van hebben gemaakt om ons als webmasters ongewenste mail te bezorgen. Gelukkig kun je dit misbruik beperken door je invoervelden te controleren, vooraleer de gegevens daadwerkelijk worden verzonden. Je kunt controleren of er wel degelijk iets is ingevuld én dat wat ingevuld is, overeenkomt met wat je verwacht dat in een bepaald veld zou moeten worden ingevuld.

We kunnen deze invoercheck uitvoeren met JavaScript of met PHP. Als je een waterdichte oplossing wilt, zul je PHP moeten gebruiken. Niet iedereen beschikt echter over deze mogelijkheid. Wat je dan kunt doen, is je invoervelden controleren met behulp van JavaScript. Let op, deze methode is niet waterdicht omdat de bezoeker JavaScript kan uitschakelen in zijn browser, maar het is een extra controle.

## Lege velden niet toestaan

We gaan beginnen met simpelweg te controleren of er wel degelijk iets is ingevuld in een invoerveld. Wanneer iemand niks invult, willen we voorkomen dat we een bericht zonder inhoud in onze mailbox aantreffen.

We gaan een functie controleer\_leeg() programmeren. Deze functie moet aangeroepen worden wanneer de gebruiker het invoerveld verlaat; hiervoor gebruiken we de event handler onchange. De functie ziet er zo uit:

<script type="text/javascript">

function controleer(contact)

{

if(contact.naam.value == "")

{

alert("Je hebt niks ingevuld.");

}

}

</script>

In de functie stoppen we als parameter de naam van het formulier, in dit geval contact. Vervolgens controleren we de waarde (value) van het invoerveld naam dat zich bevindt in het formulier contact. We roepen deze functie in ons formulier als volgt aan:

<form name="contact" method="post">

Naam:<input type="text" name="naam" onchange="controleer(this.form)" />

</form>

De event handler onchange roept hier de functie controleer\_leeg() aan en vertelt ons dat het over dit (this) formulier gaat, namelijk het formulier contact. Getallen controleren.

In ons formulier willen we ook een invoerveld toevoegen dat vraagt naar de leeftijd van de bezoeker. Ook hier zouden we kunnen controleren of de bezoeker iets heeft ingevuld, maar we kunnen dit veel specifieker doen.

De bezoeker gaat hier altijd een getal moeten invullen; dit getal kan niet kleiner zijn dan 0 en niet groter dan 102 jaar. We gaan dit testen binnen de bestaande functie. In dit voorbeeld gaan we enkel waarden tussen 18 jaar en 65 jaar toestaan.

Voeg onderstaande controle toe aan de bestaande functie:

if(contact.leeftijd.value < "18" || contact.leeftijd.value > "65")

{

alert("Je hebt als leeftijd "+contact.leeftijd.value+" jaar opgegeven. Dit is geen correcte waarde.");

}

Voeg ook de code toe aan je invoerveld om de functie te koppelen aan het invoerveld:

Leeftijd: <input type="text" name="leeftijd" onchange="controleer(this.form)">

## E-mailadres controleren

In ons formulier voorzien we ook nog een invoerveld dat naar het e-mailadres van de bezoeker vraagt. We zouden gewoon kunnen controleren of er enige invoer is, maar ook hier gaan we de controle specifieker aanpakken. In een e-mailadres komt altijd een "@" voor, dus we controleren of dit teken in het veld voorkomt.

Breid daarom de bestaande functie uit met deze code:

if(tekst.email.value =="" || tekst.email.value.indexOf("@") == -1)

{

alert("Dit e-mailadres is niet geldig.");

}

We koppelen de functie aan het invoerveld email:

E-mailadres: <input type="text" name="email" onchange="controleer(this.form)">

## Lengte controleren

Als laatste gaan we controleren of een bericht dat ingevoerd is niet te lang is. Voeg daarom onderstaande code aan de functie toe:

if(contact.opmerkingen.value.length > "30")

{

alert ("Uw opmerking is te lang.");

}

Voeg een tekstveld toe aan uw formulier om de gebruiker de kans te geven enkele opmerkingen in te voeren:

Omschrijving:

<textarea name="opmerkingen" cols="100" rows="5" Onchange="controleer(this.form);"> </textarea>

De volledige functie ziet er nu als volgt uit:

function controleer(contact)

{

if(contact.naam.value == "")

{

alert("Je hebt niks ingevuld.");

}

if(contact.leeftijd.value < "18" || contact.leeftijd.value > "65")

{

alert("Je hebt als leeftijd "+contact.leeftijd.value+" jaar opgegeven. Dit is geen correcte waarde.");

}

if(contact.email.value =="" || contact.email.value.indexOf("@") == -1)

{

alert("Dit e-mailadres is niet geldig.");

}

if(contact.opmerkingen.value.length > "30")

{

alert ("Uw opmerking is te lang.");

}

}

Alle velden van het formulier zijn nu onderworpen aan een controle.

#### 

## Javascript Opdracht 16

Maak een contactformulier waarbij meerder velden worden gecontroleerd. Zorg in ieder geval voor:

* een e-mail check;
* een leeg veld check;
* een lengte check.
* een knop

# 