

Javascript

Stefan Bucsics

Javascript ist eine Interpretersprache die entweder vom Webserver (serverseitiges Javascript) oder von neueren Webbrowsern (clientseitiges Javascript) ausgeführt wird. Allerdings gibt es dabei auch Versionsunterschiede, die manchmal beachtet werden müssen. Die empfohlenen Standards der Sprache sind im Internet unter <http://www.w3c.org/DOM/> oder <http://www.ecma.ch/> zu finden.

Im folgenden wird nur clientseitiges Javascript betrachtet. Hier wird Javascript hauptsächlich zur Kontrolle von Benutzeraktionen, zur Überprüfung von Formulareingaben und zum Erstellen dynamischer HTML-Seiten verwendet.

Diese Zusammenfassung ist als Arbeitsbehelf in der Schule gedacht und erhebt keinerlei Anspruch auf Vollständigkeit. Die hier beschriebene Sprachversion ist JavaScript1.1.

Sprachsyntax

- Javascript orientiert sich hauptsächlich an C++ und ist objekt-orientiert.
- Groß- und Kleinschreibweise wird unterschieden.
- Jeder Befehl wird mit einem Strichpunkt abgeschlossen.
- Der Kode von Javascript wird im HTML-Kode platziert. Dies kann an jeder beliebigen Stelle erfolgen, auch im Header-Teil des Dokumentes oder als Attribut in einem HTML-Tag.
- Ausserhalb eines HTML-Tags wird Javascript durch die HTML-Tags

```
<SCRIPT LANGUAGE = "JavaScript">
  Javascript-Anweisungen;
</SCRIPT>
eingeschlossen.
```

Kommentarzeichen

```
/* mehrzeiliger Kommentar */
// einzeiliger Kommentar
<!-- einzeiliger Kommentar
```

Variablen

Variablen werden nicht typisiert, die Typisierung erfolgt bei der Wertzuweisung automatisch. Man soll die Variablen aber am Anfang des Programms oder der Funktion deklarieren:

```
var Variablenname;
```

Wertzuweisung

```
Variablenname = Wert;
```

Einfache Variablentypen

Zahlen

```
1, 1.2, 1.23e+6, 0567 (octal), 0x1AF (hexadezimal)
```

Strings

```
"Text", 'Text', 'Name= "Peter"', "so geht's auch"
```

Boolean

```
true, false
```

null

```
null
```

Funktion

```
function(x){return x*x;}
```

Zusammengesetzte Variablentypen

Objekte

```
var NeuObjekt = {x:1, y:2};
```

Die Eigenschaften sind dann mit NeuObjekt.x und NeuObjekt.y anzusprechen.

Arrays

Ein Array ist ein durchnummeriertes (,indiziertes') Feld von Werten mit einem Variablennamen. Die Nummerierung erfolgt in eckigen Klammern:

```
MeinArray[1] = 2.3;
MeinArray[2] = "Otto";
MeinArray[5] = 0;
```

Eine Initialisierung des Arrays ist nicht notwendig. Neue Arrayelemente können jederzeit hinzugefügt werden. Im obigen Beispiel sind die Elemente MeinArray[3] und MeinArray[4] nicht definiert.

Eine Definition

```
var NeuArray = [1, 2, 3, "vier"];
```

ist ebenfalls möglich. Das erste Element des Arrays hat den Index 0.

In Javascript sind auch **assoziative Arrays** möglich. Die Indizes werden hier durch eigene Namen ersetzt. Dies ermöglicht eine leichtere Lesbarkeit.

```
Lager['Apfel'] = 12;
Lager['Birne'] = 8;
```

Operatoren

Entsprechen weitestgehend denen in der Sprache C++.

Numerisch

```
+, -, *, /, %, ++, --
```

Vergleich

```
<, >, <=, >=, ==, !=
```

Zuweisung

```
=, +=, -=, *=, /=
```

Logisch

```
&&, ||, !
```

String

```
Gesamtstring = String1 + String2;
```

Kontrollstrukturen

```
if (Bed) { Befehle; }
if (Bed) { Befehle; } else { Befehle; }
if (Bed1) { Befehle; } elseif (Bed2) { Befehle; } else { Befehle; }
switch (Ausdruck) {case Wert1: Befehle; break; case Wert2: Befehle; break; default: Befehle;}
while ( LaufBed) { Befehle; }
do { Befehle; } while (LaufBed);
for (Initialisierung; Laufbedingung; Ende)
  { Befehle; }
for (Variable in Objekt) { Befehle; }
```

Beispiele

```
if (wert == 1) {
  document.write ("Wert = eins");
}
```

Zeilenweise Ausgabe gerader Zahlen von 2 bis 10:

```
for (i=2; i<=10; i+=2) {
  document.writeln (i);
}
```

Eigenschaften des Navigatorobjekts ausgeben:

```
for (wert in navigator) {
  document.writeln (wert);
}
```

Anmerkung: Eine Schleife kann mit dem Befehl `break` vorzeitig beendet werden. Mit dem Befehl `continue` kann der momentane Schleifendurchgang beendet und der nächste Durchgang gestartet werden.

Funktionen

Funktionen werden üblicherweise am Anfang des Codes definiert:

```
function Name (Parameterliste)
{ Befehle;
  return Wert;
}
```

Aufruf

```
Rückgabewert = Name(Argumentenliste);
```

oder über eine Variable

Definition

```
var Quadrat = function(x){return x*x};
```

Aufruf

```
document.write(Quadrat(3));
```

Variablen, die innerhalb einer Funktion mit `var Variablenname;` deklariert werden, gelten nur lokal. Somit sind sie von Variablen außerhalb der Funktion zu unterscheiden, auch wenn diese denselben Namen haben!

Objekte

Javascript enthält viele vorgegebene Objekte. Jedes Objekt besitzt einen Namen, bestimmte Eigenschaften und Methoden (das sind Funktionen, die mit diesem Objekt zusammenhängen). Außerdem sind mit bestimmten Objekten Ereignisse (wie `onClick`) verbunden, auf die das Objekt reagieren kann.

Vergibt man im HTML-Code den Objekten eigene Namen (Attribut: `NAME = „ObjektName“`), so können diese Objekte über ihren Namen angesprochen werden. Die Objekteigenschaften und -methoden sind mittels

```
ObjektName.ObjektEigenschaft
```

```
ObjektName.ObjektMethode()
```

anzusprechen.

Beispiel

```
document.Formul arname. Ei ngabefel dName. val ue = „Hal lo!“;
```

Alle Objekte mit ihren Eigenschaften, Methoden und Ereignissen kann man einschlägiger Literatur entnehmen. Hier sind nur die wichtigsten erwähnt:

An oberster Stelle der Objekthierarchie befinden sich die Objekte

`Math`, `navigator` und `window`.

Das Objekt `Math` enthält die Mathematikfunktionen.

Das Objekt `navigator` repräsentiert den Webbrowser (mit den Eigenschaften `appName`, `appName`, `appVersion`, `userAgent`) und dient meist zur Feststellung, ob der Benutzer einen für die verwendete HTML-Version geeigneten Browser besitzt.

Das Objekt `window` ist das meistgebrauchte Objekt (Defaultwert) und symbolisiert das Browserfenster. Es besitzt weitere Unterobjekte wie `history`, `location`, `frames[]` und `document`.

Das Objekt `document` wiederum repräsentiert das im Browserfenster gezeigte Dokument. Es enthält den HTML-Code, sowie die Objekte `links[]`, `images[]`, `forms[]`, usw., die alle als (assoziative) Arrays angesprochen werden können.

Die meistgebrauchte Methode des `document`-Objektes ist

```
document.write(*Text*);
```

Es ist möglich, eigene Objekte zu definieren:

```
// Definition der Konstruktorfunktion
function Auto(typ, farbe, ps, besitzer) {
  this.typ = typ;
  this.farbe = farbe;
  this.ps = ps;
  this.besitzer = besitzer;
}
// Definition einer Methode
function Besi tzerwechsel (name) {
  this.besi tzer = name;
}
// 'Instanzi erung' ei ner Objektvari ablen
Wagen = new Auto("BMW", "rot", 150, "Mayer");
// Zuwei sung der Methode
Wagen.Besi tzer = Besi tzerwechsel;
// Ändern der Eigenschaft 'farbe'
Wagen.farbe = "blau";
// Aufruf der Methode Besi tzer und Ändern des Namens
Wagen.Besi tzer("Mül ler");
```

Events

Beispiel für den Einsatz von Ereignissen

```
<FORM>
< INPUT
  TYPE ="button"
  NAME = "knopf"
  VALUE = "Bi tte drueck mi ch!"
  onCl ick = "al ert('Danke!');">
</FORM>
```

Die Events von Javascript 1.2 werden nicht von allen Browsern unterstützt. Daher sind in der Tabelle nur Ereignisse angeführt, die von Javascript 1.1 unterstützt werden.

Ereignis	Ausgelöst durch	Verbunden mit
<code>onAbort</code>	Abbrechen des Ladevorgangs	Image
<code>onBlur</code>	Verlust des Fokus	Text, Window, alle Formularobjekte
<code>onChange</code>	Verändern des Inhaltes	Select, Textinput
<code>onClick</code>	einmaliges Klicken; um die Standardaktionen auszuschalten, muss man false zurückgeben	Link, Button
<code>onError</code>	Fehler beim Laden des Bildes	Image
<code>onFocus</code>	Erhalt des Fokus	Text, Window, alle Formularobjekte
<code>onLoad</code>	Beenden des Ladevorgangs	Window, Image
<code>onMouseOut</code>	Fortbewegen der Maus	Link
<code>onMouseOver</code>	Bewegen der Maus über das Objekt	Link
<code>onReset</code>	Drücken des Resetknopfes beim Formular; um das eigentliche Reset auszuschalten, false zurückgeben	Formular
<code>onSubmit</code>	Drücken des Submitknopfes beim Formular; um das eigentliche Submit auszuschalten, false zurückgeben	Formular
<code>onUnload</code>	Beenden eines Dokumentes	Window

Der Internetexplorer 4.0 von Microsoft unterstützt eine Fülle von weiteren Events bei praktisch allen Objekten.

Anwendungsbeispiele

Persönliche Begrüßung

```
<SCRIPT LANGUAGE = "JavaScript">
var Name = prompt("Bitte Deinen Namen eingeben:");
// Aufforderung zur Namenseingabe
document.writeln("<H1>Hallo liebe(r) " + Name + "</H1>");
// Ausgabe des eingegebenen Namens
</SCRIPT>
```

Zurück

```
<A HREF = # onClick = "history.back();return false;">zurück</A>
// entspricht der BACK-Ikone des Browsers
Echtzeituhr in Statuszeile
<SCRIPT LANGUAGE = "JavaScript">
// holt die aktuellen Stunden, Minuten und Sekunden ...
function uhrzeit()
{
var zeit = new Date(); // aktuelle Systemzeit holen
var stunden = zeit.(); // Stundenzahl
var minuten = zeit.getMinutes(); // Minutenzahl
var sekunden = zeit.getSeconds(); // Sekundenzahl
// Ausgabe aller Werte in die Statuszeile
status = stunden + " : " + minuten + " : " + sekunden;
// Aufruf der Funktion ,uhrzeit()'
// alle weiteren 1000 Millisekunden
setTimeout ("uhrzeit()", 1000);
}
uhrzeit(); // erster Aufruf der Funktion ,uhrzeit()'
</SCRIPT>
```

Letzte Änderung

```
<SCRIPT LANGUAGE = "JavaScript">
// neues Datumobjekt mit dem Änderungsdatum
var zeit = new Date(document.lastModified);
// Monate von 0 bis 11
document.writeln(zeit.getDate() + ". "+
(1+zeit.getMonth())+"." + zeit.getFullYear() );
</SCRIPT>
```

Browserbestimmung

```
<SCRIPT LANGUAGE = "JavaScript">
Document.writeln
("<H1>Ich weiss etwas um!;ber Ihren Browser!</H1>");
document.writeln ("appName: " +
navigator.appCodeName + "<P>");
document.writeln ("appName: " + navigator.appName + "<P>");
document.writeln ("appVersion: " + navigator.appVersion + "<P>");
document.writeln ("userAgent: " + navigator.userAgent + "<P>");
</SCRIPT>
```

Alle Browser-Plugins

```
<SCRIPT LANGUAGE = "JavaScript">
// alle Plugins bilden ein Array
for (i=0; i<navigator.plugins.length; i++) {
document.writeln(navigator.plugins[i].name + "<P>");
}
</SCRIPT>
```

Roll Over

```
<SCRIPT LANGUAGE = "JavaScript">
//Vorladen der Images
var Standard=new Image(); var Highlight=new Image();
Standard.src = "image/standard.gif";
Highlight.src = "image/highlight.gif";
</SCRIPT>
<A HREF = #
onmouseover = "document.images[0].src = Highlight.src;"
onmouseout = "document.images[0].src = Standard.src;">
<IMG SRC = "image/standard.gif" BORDER = 0></A>
```

*Ein Bankier ist ein Mensch, der einen
Schirm verleiht, wenn die Sonne scheint,
und der ihn sofort zurück haben will,
wenn es zu regnen beginnt.*

MARK TWAIN

DIGITAL communication

ISDN · TELEFONANLAGEN · NETZWERKE · COMPUTER · WEB-DESIGN

**April, April der macht
die Preise wie er will.**

Gültig im April.

GESKO OFFICE 1106 ISDN-Telefonanlage

- 1x So extern, 1x So intern
- 6x Nebenstellen analog, 1x TOR
- 8x Nebenstellen digital
- Durchwahl, Gebühren, CLIP, CTI
- Anrufumleitung
- Windows-Software, opt. MAC
- Gratis Support



ATS 4.990.-

AVM Fritz!Card ISDN-Karte

- passive ISDN-Karte
- für Win 3.11/95/98/NT
- Fritz!32 Software auf CD
- G3 Fax senden/empfangen
- Internet / Mailbox / Eurofile
- HDLC / X.75 / V.110 / V.120
- Gratis Support



ATS 990.-

Canon FAX B150 Telefaxgerät

- Normalpapierfaxgerät
- Nummernspeicher
- Display mit 2 Zeilen/20 Zeichen
- Faxseitenspeicher
- Anschlußkabel (3m)
- österreichische Postzulassung
- Gratis Support



ATS 3.990.-

Hagenuk OCIP ISDN-Schnurlostelefon + TA

- ISDN, DECT, Rufnummeranzeige
- bis zu 6 Mobilteile, 2 x a/b für Fax, AB, ...
- Plug & Play, Durchwahl mit MSN
- Telefonbuch für 20 Namen+Nummern
- Anruferliste, gratis Interngespräche
- 80h Standby, 8 Std. Gespräch
- Gratis Support



ATS 3.990.-

Irtümer und Fehler vorbehalten.

Digital Communication

1050 Wien, Gassergasse 19/5
Tel.: 01/548-5000-0, Fax: 01/548-5050
<http://www.digicom.at>
Mo-Do : 9-18, Fr. 9-15