Teil 1: Javascript-Grundlagen

Javascript verstehen

Wollen Sie Bewegung, Interaktion oder Mauseffekte auf Ihre Homepage bringen, führt an Javascript kein Weg vorbei. Im ersten Teil der Serie lernen Sie die Sprachelemente und Funktionsweise kennen

Javascript ist eine beliebte Skriptsprache und hat außer der Namensähnlichkeit nichts mit Java zu tun. Javascript erlaubt es, statische HTML-Seiten dynamisch zu gestalten, etwa indem sie auf Mausbewegungen oder Tastatureingaben reagieren. In Kombination mit Cascading Stylesheets können Sie Seitenelemente bewegen oder verschwinden lassen. Mehr dazu lesen Sie im Artikel „Dynamisches HTML“ in com! 10/2004 ab Seite 130. Außerdem realisieren Sie mit Javascript Rollover-Effekte, Navigationsmenüs, Spiele, Maustrailer, Formularabfragen, Pop-ups und vieles mehr. Zahlreiche vorgefertigte Skripts können Sie nach Ihren Bedürfnissen anpassen. Dazu benötigen Sie jedoch ein fundamentales Verständnis der Sprache. In diesem Artikel lernen Sie die grundlegenden Sprachelemente von Javascript kennen. Alle Beispiele finden Sie auf der com!-Heft-CD 1 in der Rubrik „HomeP@ge“, „Praxis & Tuning“.

Javascript-Serie

Die dreiteilige com!-Serie macht Sie mit den Grundlagen und Einsatzmöglichkeiten von Javascript vertraut.

- Teil 1: **Javascript verstehen**
- Teil 2: **Event-Handler einsetzen**
- Teil 3: **Fertige Javascripts anpassen**

Die Funktionsweise

Sie schreiben Javascript direkt in den HTML-Code. Wenn ein Besucher eine solche Seite im Browser aufruft, führt dieser das Skript aus.

Anders als bei PHP oder Perl erfolgt die Ausführung auf Seiten des Clients, also im Browser des Surfers und nicht auf dem Webserver. Allerdings hat jeder Browser seine eigene Javascript-Implementierung. Zu Zeiten des Browser-Kriegs zwischen Netscape und Microsoft kochte jeder Hersteller sein eigenes Süppchen. Viele Skripts funktionierten nur mit dem Internet Explorer, andere nur mit Netscape. Heute sind diese Unterschiede weitgehend behoben. Moderne Browser wie Firefox unterstützen ECMA-262, also Javascript 1.5, weitgehend – siehe auch den Kasten „Javascript-Geschichte“ auf Seite 134.

Mittels einer Schnittstelle zum so genannten Document Object Model (DOM) können Sie Elemente einer Webseite noch manipulieren, nachdem diese zum Browser übertragen wurde.

Vorbereitungen treffen

Was brauchen Sie für Javascript? Anders als bei PHP sind Sie nicht auf die Konfiguration des Webservers angewiesen, denn die Skripts laufen ja im Browser. Sie benötigen lediglich einen Text-Editor, um Skripts zu erstellen oder anzupassen. Verwenden Sie aber keinesfalls Microsoft Word. Geeignet sind etwa Wordpad oder ein Editor mit Syntax-Highlighting wie Macromedia Homesite.

Es gibt zwei Möglichkeiten, Javascript auf Webseiten einzusetzen: Sie können Javascript direkt in HTML-Seiten schreiben oder eine separate externe Javascript-Datei einbinden. Für die erste Methode verwenden Sie

```
<script language="javascript"
type="text/javascript">
Hier steht das Skript
</script>
```

Ehemals pflegte man den Code mit den Kommentarzeichen `<!–` und `// –>` zu umgeben, um alte Browser zu versorgen, die kein Javascript verstehen. Das ist heutzutage eigentlich nicht mehr nötig, schadet aber auch nicht. Und wohin mit dem Skript? Im Allgemeinen platzieren Sie

Javascripts im **<head>**-Bereich Ihrer Webseite. Damit stellen Sie sicher, dass der Browser das Skript vor dem Rest der Seite lädt. Nur wenn Sie etwa einen Text auf der Seite ausgeben wollen, stellen Sie das Skript an die entsprechende Stelle im **<body>**-Bereich.

Die zweite Möglichkeit, bei der Sie das Skript in eine externe Datei auslagern, funktioniert so:

```
<script type="text/javascript"
src="skript.js">
</script>
```

Damit binden Sie das Skript im **<head>**-Bereich einer HTML-Seite ein. Das spart Speicherplatz und erleichtert Änderungen am Skript, da Sie nicht mehr in jeder einzelnen HTML-Datei, sondern nur noch in der zentralen Javascript-Datei den Code ändern. Beachten Sie, dass Sie bei externen Javascripts die den Code umgebenden **<script>**-Tags weglassen müssen.

Das erste Skript

Es ist fast schon zur Tradition geworden, in einer neuen Sprache ein Programm zu schreiben, das nichts anderes macht, als den Text „Hallo Welt" auf dem Bildschirm auszugeben. Dazu haben Sie mit Javascript gleich mehrere Möglichkeiten:

Geben Sie wie oben beschrieben die **<script>**-Tags in den Body einer HTML-Seite ein – Sie wollen ja etwas auf der Seite ausgeben – und schreiben Sie dazwischen den Befehl **`alert("Hallo Welt!");`**. Speichern Sie die Datei und rufen Sie diese in Ihrem Browser auf. Es erscheint eine so genannte Alert-Box mit dem Text „Hallo Welt!". Damit haben Sie auch gleich Ihre erste Javascript-Funktion kennen gelernt, **`alert()`**. Was Sie zwischen die runden Klammern schreiben, gibt das Skript in einer Alert-Box aus. Eine weitere Regel: Nach jedem Javascript-Befehl steht ein Strichpunkt.

Das erste Skript: Über die Alert-Box können Sie beliebige Texte ausgeben und etwa auf falsche Formulareingaben hinweisen

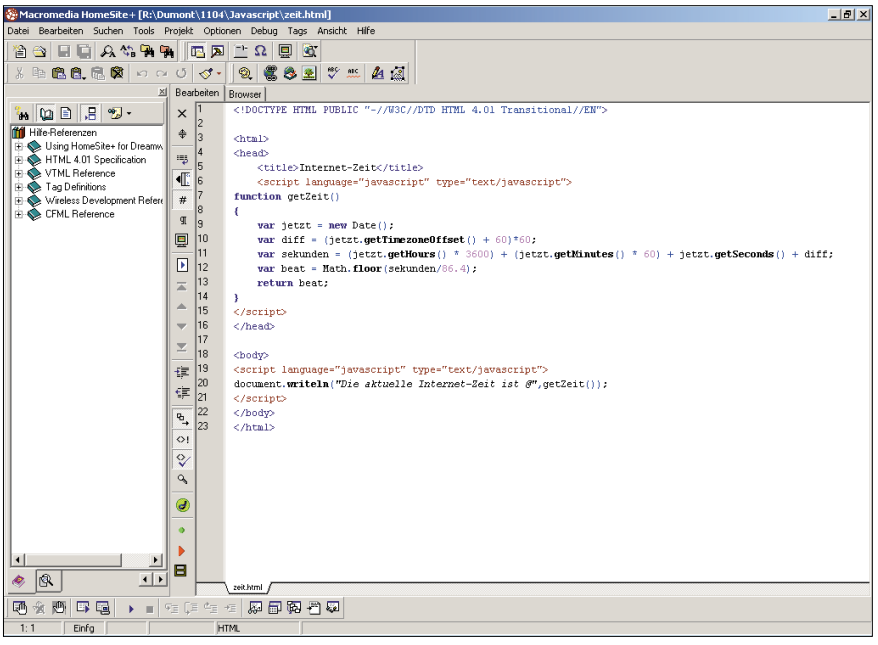
Als zweite Lösung können Sie auch

```
document.write("Hallo Welt!");
```

schreiben. In diesem Fall erscheint keine Alert-Box, sondern der Text wird auf die Webseite geschrieben.

Zum besseren Verständnis ein kurzer Ausflug in das Document Objekt Model (DOM): Es beschreibt, wie sämtliche Seitenelemente, etwa Bilder, Formularfelder oder Paragrafen, zu der übergeordneten Struktur, also der Seite selbst, in Beziehung stehen. Indem Sie ein Element mit seinem korrekten DOM-Namen ansprechen, können Sie es beeinflussen und verändern.

Das **`document`**-Objekt ist ein solches Element. Es bezieht sich auf den Inhalt, der in einem Browser-Fenster angezeigt wird. In der Objekthierarchie von Javascript liegt es direkt unterhalb des **`window`**-Objekts. Die Methode **`document.write()`** schreibt einen Text in die aktuelle Webseite. Verwenden Sie **`writeln`** statt **`write`**, fügt sie auch noch einen Zeilenumbruch hinzu. Mehr zu Objekten lesen Sie im folgenden Abschnitt.



Vorbereitungen treffen: Ein Editor mit Syntax-Highlighting bietet Vorteile, ist aber nicht zwingend nötig

Sprach-Grundlagen

Um komplexere Skripts verstehen und anpassen zu können, benötigen Sie einige Kenntnisse von Sprachelementen, die immer wiederkehren.

Variablen sind Speicherbereiche, in denen Sie Daten speichern. Sie können den Wert einer Variablen jederzeit ändern und Sie deklarieren sie in der Art

```
var ganzezahl = 42;
var kommazahl = 3.1415;
var zeichenkette ="Hans";
```

Der Variablentyp ergibt sich aus der Wertzuweisung. Ein kleines Beispielskript bestimmt zwei Variablen, multipliziert diese und gibt das Ergebnis aus.

```
var i = 45;
var j = 91.1;
var produkt = i * j;
document.write
(i, "*", j, "=", produkt);
```

Sollen bestimmte Arbeitsabläufe wiederholt stattfinden, leisten Schleifen gute Dienste.

Die **`for`**-Schleife verstehen Sie am besten anhand eines Beispiels:

```
var summe = 0;
for (var i=0; i<=100; i=i+1)
{
 summe=summe + i;
}
document.write("Die Summe der
Zahlen 1 bis 100 ist ",summe);
```

Die **`for`**-Schleife führt die Anweisungen zwischen den geschweiften Klammern durch, so lange die Bedingung **`i<=100`** wahr ist. Nach jedem Durchlauf erhöht sich der Wert der Variablen i um 1 **`(i=i+1)`**.

Die Aufgabenstellung lässt sich ebenfalls mit einer **`while`**-Schleife lösen. Während die Bedingung wahr ist, führt sie die Anweisungen aus:

```
var summe = 0;
var i=0;
while (i<=100)
{
 summe=summe + i;
 i++;
}
document.write("Die Summe der
Zahlen 1 bis 100 ist ",summe);
```

Achten Sie darauf, dass eine funktionierende Abbruchbedingung vorhanden ist, sonst produzieren Sie eine Endlosschleife. **`i++`** ist gleichbedeutend mit **`i=i+1`**. ▶

Sehr ähnlich funktioniert die Schleife **do while**. Der wichtige Unterschied ist, dass die Schleife mindestens ein Mal durchläuft, da die Bedingung erst am Ende geprüft wird.

```
do
{
 summe=summe + i;
 i++;
}
while (i<=100)
document.write("Die Summe der
Zahlen 1 bis 100 ist ",summe);
```

Eine große Bedeutung hat auch die Konstruktion **if else**, also: Wenn etwas zutrifft, dann führe etwas aus, ansonsten führe etwas anderes aus.

```
var i=8;
var j=5;
if(i<j)
{
 document.write
(i, " ist kleiner als ", j);
}
else
{
 document.write
(i, " ist größer als ", j);
}
```

Arrays sind vereinfacht gesagt eine Sammlung von Werten. Sie legen ein solches Array an mit

```
var meinArray = new Array();
```

Die Zuweisung von Werten erfolgt über einen Index, der immer bei 0 beginnt.

```
meinArray[0] = 1001;
```

Funktionen sind wichtig, um Programme zu strukturieren. Sie erlauben es, mehrfach verwendete Anweisungsfolgen unter einem Namen zusammenzufassen. Ein Beispiel:

```
function quadrat(x)
{
x=x*x;
return x;
}
document.write("Das
Quadrat von 123 ist ",
quadrat(123));
```

Objekte sind fest umgrenzte Datenelemente mit Eigenschaften und meist auch mit objektgebundenen Funktionen, Methoden genannt. Ein Objekt haben Sie ja bereits mit **document** kennen gelernt. Zur Veranschaulichung: Ein Buch ist ein Objekt. Es hat verschiedene Eigenschaften wie Titel und Seitenzahl. In Javascript würden Sie für ein aufgeschlagenes Buch **buch.aufgeschlagen** schreiben, also zuerst das Objekt, dann mit einem Punkt getrennt die Eigenschaft. Eine Methode wäre **buch.zuschlagen()** oder **buch.aufschlagen(299)**.

Javascript kennt mehrere Arten von vordefinierten Objekten wie **Object, Screen, Array, String, Date** und **Math**. Andere Objekte gehören zum DOM wie **window, documents** und **form**.

Das Objekt **window** ist das oberste Objekt der Objektfamilie und steht für das Browser-Fenster. Die Methode **window.close();** etwa schließt das aktuelle Browser-Fenster. **window** können Sie auch weglassen. So hat **close();** die gleiche Wirkung wie **window.close();**.

Event-Handler legen fest, wie der Browser auf ein bestimmtes Ereignis reagiert. Ein solches Ereignis kann beispielsweise eine Mausbewegung, ein Mausklick, ein Doppelklick oder eine Tastatureingabe sein. Mehr zu Event-Handlern lesen Sie in Teil 2 der Serie.

Fertige Skripts sind meist kommentiert. In Eigenproduktionen sollten Sie ebenfalls Beschreibungen einfügen, damit Sie das Skript auch nach einem halben Jahr noch verstehen. Einzeilige Kommentare beginnen Sie mit der Zeichenfolge **//**. Der Browser ignoriert den nachfolgenden Text bis zum Zeilenende. Mehrzeilige Kommentare leiten Sie mit **/*** ein, und mit der umgekehrten Folge ***/** beenden Sie den Kommentar.

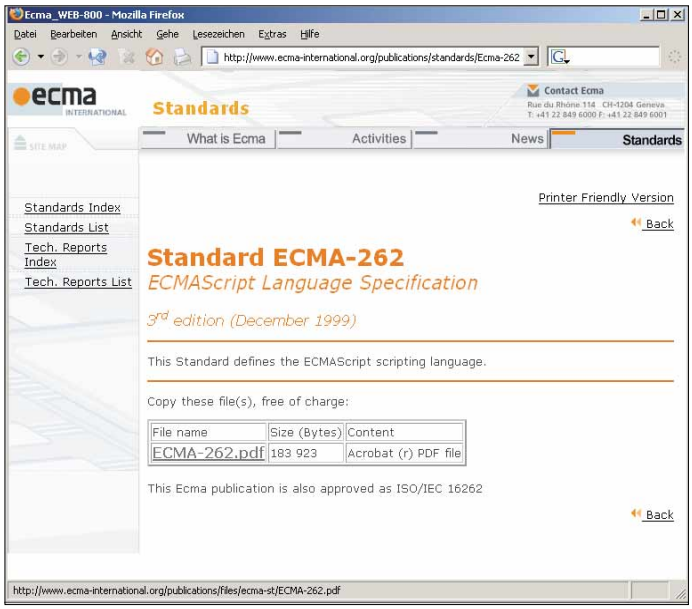
Temperatur-Rechner

Ein praktisches Beispiel, das zeigt, wie HTML und Javascript zusammenarbeiten, finden Sie in der Datei *temperatur.html* auf der com!-Heft-CD 1 unter „HomeP@ge“, „Praxis & Tuning“. Diese funktioniert als Rechner, der Temperaturangaben in Grad Celsius, Fahrenheit und Kelvin ausgibt. Geben Sie eine Zahl in eines der drei Felder ein, werden die anderen automatisch mit den entsprechenden Werten gefüllt. Auf ähnliche Weise könnten Sie auch einen Währungsrechner verwirklichen.

Öffnen Sie die Datei mit einem Text-Editor und sehen Sie sich den Body-Bereich an. Dort definieren Sie zunächst ein normales HTML-Formular und nennen es *felder* mit **<form name="felder">**. Anschließend folgen drei Eingabefelder für Grad Celsius, Fahrenheit und Kelvin in der Art

```
<input name="celsius"
onkeyup="wandle('C')">
Grad Celsius
```

Beim Loslassen der Taste **(onkeyup)** wird die Funktion **wandle** mit dem Parameter **C** für Celsius aufgeru-



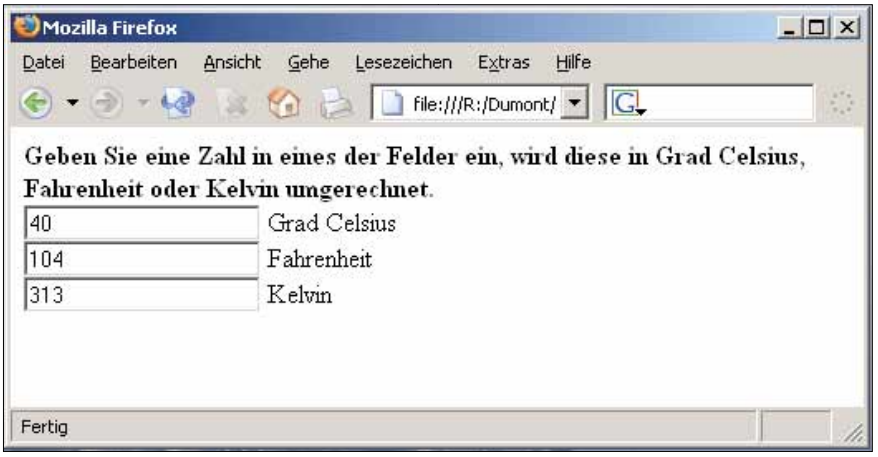
Sprach-Grundlagen: Auf der ECMA-Website finden Sie eine Javascript-Referenz

Javascript-Geschichte

Javascript wurde von Brendan Eich von Netscape Communications zunächst unter den Namen Mocha und Livescript entwickelt und später in Javascript umbenannt. Javascript hat nichts mit Java zu tun, lediglich die Syntax ist in beiden Sprachen ähnlich der der Programmiersprache C. Javascript 1.0 erblickte im März 1996 im Browser Netscape 2.0 das Licht der Welt. Microsoft zog nach mit einer eigenen Variante namens JScript mit Version 1 im Internet Explorer 3.0.

Nachdem Netscape den Code offen legte, erfolgte eine Standardisierung durch die European Association for Standardizing Information and Communication Systems (ECMA) mit Sitz in Genf.

Stand der Dinge ist ECMA-262 Edition 3, was im Prinzip Javascript 1.5 entspricht. Alle aktuellen Browser-Versionen sind mit ihrer Javascript-Implementierung weitgehend zu ECMA-262 kompatibel. Eine ausführliche Beschreibung finden Sie *unter www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf*. Das Chaos, als jeder Browser eine eigene Javascript-Variante beherbergte und Programmierer in den Wahnsinn trieb, scheint somit Vergangenheit zu sein.



Temperatur-Rechner: Mit einem kleinen Javascript rechnen Sie Temparatur-Einheiten ineinander um

fen. Diese Funktion steht im **<head>**-Bereich: **function wandle(grad)**. Dort erfolgt mit **if else** eine Abfrage, ob der übergebene Parameter **C, F** oder **K** ist. Je nachdem werden die entsprechenden Anweisungen zwischen den geschweiften Klammern ausgeführt. Tragen Sie in das Celsius-Feld etwas ein, trifft bereits die erste Abfrage zu und der folgende Code wird abgearbeitet:

```
if (grad=="C")
{
F=document.felder.celsius.value
* 9 / 5 + 32;
K=parseFloat(document.felder.
celsius.value) + 273.15;
document.felder.fahrenheit.value=
Math.round(F);
document.felder.kelvin.value=
Math.round(K);
}
```

Beachten Sie: Zuweisungen wie **y=4** erfolgen mit einem Gleichheitszeichen, logische Abfragen wie **if (grad=="C")** aber mit zwei.

Im Beispiel werden die Werte für Fahrenheit und Kelvin berechnet.

```
F=document.felder.celsius.value
* 9 / 5 + 32;
```

bedeutet: Nimm den Wert **(value)** aus dem Eingabefeld namens *celsius* des Formulars *felder* **(felder.celsius)** der aktuellen Webseite (**document**, korrekt wäre auch **window.document**), multipliziere diesen Wert mit neun Fünftel und addiere 32. Die Methode **Math.round(F)** rundet diesen Wert, und das Ganze wird in das entsprechende Formularfeld geschrieben mit **document.felder.fahrenheit.value.** Die Funktion **parseFloat** bei der Berechnung des Kelvin-Werts ist hier nötig, weil Javascript die Variable sonst auch als Zeichenkette auffassen könnte. Die Funktion wandelt Zeichenketten in Fließkommazahlen um.

Swatch-Beats

Die so genannte Internet-Zeit hat sich die Firma Swatch 1998 ausgedacht. Sie teilt einen Tag in 1000 Beats à 86,4 Sekunden ein, bezogen auf die Zeit der Stadt Biel in der Schweiz, die in der Zeitzone Central European Time (CET) liegt. Die Zeitangabe erfolgt mit vorangestelltem @. Ein Vorteil der Internet-Zeit: Will man sich im Internet etwa zum Chatten oder für ein Spiel verabreden, muss man nicht mehr die verschiedenen Zeitzonen berücksichtigen, denn eine Internet-Zeitangabe wie @688 ist weltweit gleich.

Mit einem kleinen Javascript geben Sie die Internet-Zeit auf Ihrer Homepage aus. Sie finden es in der Datei *zeit.html* auf der com!-Heft-CD 1. So funktioniert das Skript: Im Seitenkopf definieren Sie zunächst mit **function getZeit()** eine Funktion, in der Sie die Zeitberechnung zusammenfassen. Dort erstellen Sie mit **var jetzt = new Date();** ein neues Zeitobjekt mit der aktuellen lokalen PC-Zeit und weisen diese der Variablen **jetzt** zu. Die Methode **getTimezoneOffset()** gibt die Differenz zwischen der lokalen Zeit und der Greenwich Mean Time (GMT) in Minuten zurück. Da die CET immer um eine Stunde zur GMT differiert, addieren Sie 60 und multiplizieren mit 60, um die Differenz in Sekunden zu erhalten, zusammen also

```
var diff = (jetzt.
getTimezoneOffset() + 60)*60;
```

Nun berechnen Sie die Tageszeit in Sekunden und addieren die soeben berechnete Differenz:

```
var sekunden = (jetzt.getHours()
* 3600) + (jetzt.getMinutes()
* 60) + jetzt.getSeconds() + diff;
```

Anschließend teilen Sie die Zahl der Sekunden durch 86,4 und runden nach unten ab

```
var beat =
Math.floor(sekunden/86.4);
```

Das Ergebnis steht nun in der Variablen **beat**, deren Wert Sie mit **return beat;** zurückgeben.

Bleibt noch die Ausgabe der Internet-Zeit. Dazu verwenden Sie die bereits bekannte Konstruktion **document.write**, in der auch der Aufruf der Funktion **getZeit** erfolgt:

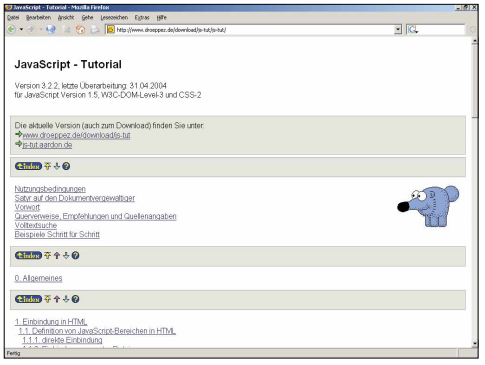
```
document.writeln("Die aktuelle
Internet-Zeit ist @",getZeit()); ■
```

Andreas Dumont
homepage@com-magazin.de

Javascript-Quellen

Unter den folgenden Adressen finden Sie Anleitungen und Einführungen zum Thema Javascript.

- ***www.droeppez.de/download/js-tut/js-tut***
- ***http://ventrue.myspace.ath.cx/js-tut***
- ***www.onlinetutorials.de/jsc-index.htm***
- ***www.javascript-fx.com/navigation***
- ***www.javascriptkit.com/dhtmltutor***
- ***www.webreference.com/js/***
- ***www.w3schools.com/js/default.asp***
- ***www.webteacher.com/javascript/ch02_3.html***
- ***www.javascript-world.de***
- ***www.htmlgoodies.com/primers/jsp***
- ***www.mozilla.org/js/scripting/***
- ***http://de.selfhtml.org/javascript/objekte/document.htm***



Ein gutes deutschsprachiges Javascript-Tutorial finden Sie auf den Webseiten von *www.droeppez.de*