

Mein zweites JavaScript

An JavaScript kommen Sie nicht vorbei! Mit unserem Workshop steigen Sie einfach in die Programmierung ein. Diesmal lernen Sie, Bilder zu manipulieren und Event-Handler einzusetzen.



Für die erste Übung zur Manipulation von Bildern reicht eine Webseite mit Foto und Button.

Ihnen unser Kochkurs. Auch wenn Sie noch nichts über JavaScript wissen, lohnt es sich weiterzulesen. Wir erklären Ihnen jede Zeile des Codes und alle nötigen Befehle.

Als Beispiel bauen Sie eine Webseite, die verschiedene Bilder einblendet, wenn Sie die Navigationsleiste mit der Maus überfahren. So erhalten Sie eine attraktive Startseite oder eine ungewöhnliche Grafikgalerie. Alle Beispiele finden Sie auf der Heft-CD unter *Aktuelles Heft*.



Bilder manipulieren

Als ersten Schritt schreiben Sie ein kleines Skript, das auf einer Webseite ein Bild austauscht, wenn Sie einen Button anklicken. So erfahren Sie, wie man in JavaScript einzelne Bilder und andere Elemente manipuliert.

Als Basis dient eine kleine HTML-Datei mit einem Bild und einem Button:

```
<html>
<head></head>
<body>
  
  <p>
    <form>
      <input type="button"
        value="Bild tauschen">
```

```
</form>
</body>
</html>
```

Bevor Sie mit dem Programmieren anfangen, ist es notwendig, das Bild auf der Webseite eindeutig zu identifizieren. Denn auf einer Seite können viele Bilder vorhanden sein. Typisch für HTML ist die Lösung, dem Foto über ein zusätzliches Attribut einen Namen zuzuweisen:

```

```

Damit ist das Bild über den Namen **landschaft** zu erreichen.

Objekte der Begierde

Jetzt wissen Sie, wie Sie dem Browser im Skript mitteilen, welches Bild er zu bearbeiten hat. Aber was soll mit dem

! Tipp Info Links

Event-Handler für Text

Wenn Sie ein Wort oder einen Satz im Text mit einem Event-Handler ausstatten wollen, müssen Sie zu einem Trick greifen, denn einen Handler können Sie nur in einem Tag eintragen. Es bietet sich an, die Textstellen, die eine Aktion auslösen, besonders zu formatieren – etwa fett oder kursiv. In die entsprechenden Tags **** und **<i>** bauen Sie dann den Event-Handler ein. Wollen Sie, dass der aktive Text wie normale Schrift aussieht, verwenden Sie Absatz-Tags. Mit **<p>...</p>** oder **<div>...</div>** markieren Sie einen ganzen Abschnitt. Soll nur ein Wort der Auslöser sein, so benutzen Sie **...**.



Auf Ihren Webseiten soll sich etwas bewegen? Ihre Sites sollen interaktiv auf Eingaben und Mausbewegungen der Besucher reagieren? Dann führt der schnellste Weg über JavaScript, eine einfache Programmiersprache. Dazu gibt es zahlreiche fertige Skripts zum Download, die sich rasch in die eigene Webseite kopieren lassen.

Doch danach fangen die Probleme meist an: Das Skript verursacht eine Fehlermeldung, der gewünschte Effekt bleibt aus, oder Sie sind mit dem Ergebnis nicht zufrieden. Jetzt müsste man mehr über JavaScript wissen! Genau dieses Know-how vermittelt

Bild passieren? Sie möchten den Inhalt des Bilds austauschen, indem Sie die Datei `bild1.jpg` durch `bild2.jpg` ersetzen.

In JavaScript gilt: Jedes Element ist ein Objekt, also auch das Bild `landschaft`. Ein solches Objekt hat Eigenschaften und Verhaltensweisen. Eine Eigenschaft eines Bildes ist der Dateiname. Diese Eigenschaft heißt `src` – wie das Attribut im ``-Tag. Um die Eigenschaft eines Objekts abzufragen oder zu verändern, hängen Sie das Schlüsselwort mit einem Punkt an die Objektbezeichnung:

```
landschaft.src
```

Damit der Browser ein neues Bild anzeigt, weisen Sie dieser Eigenschaft einfach einen neuen Dateinamen zu. Eine Zuweisung erledigt das Gleichheitszeichen, die Programmzeile lautet also:

```
landschaft.src="bild2.jpg"
```

Mehr ist nicht nötig, um auf einen Mausklick hin ein einzelnes Bild in Ihrer Webseite auszutauschen.

Skript starten

Der einfachste Weg, das Skript in den Quelltext einzubauen und zu starten führt über einen Button. Dazu fügen Sie den Event-Handler `onClick="..."` in das Tag `<input ...>` ein. Frei übersetzt ist das eine Routine, um ein Ereignis zu bearbeiten zur Ereignisbearbeitung – der englische Begriff klingt deutlich besser.

Mit dem Event-Handler teilen Sie dem Browser mit, dass er auf eine Aktion des Benutzers reagieren soll. Das `onClick` bedeutet, dass der Browser bei einem Klick auf den Button die Befehle in den Anführungszeichen ausführt. Im Beispiel fügen Sie die Befehlszeile aus dem letzten Abschnitt in die Anführungszeichen ein.

Dabei taucht ein Problem auf: Was machen Sie mit den doppelten Anführungszeichen? Schon die Befehlszeile enthält welche, und bei `onClick` müssen Sie ebenfalls Anführungszeichen setzen. Die Lösung: Für HTML sind einfache (') und doppelte Anführungsstriche (") gleichwertig. Sie müssen nur darauf achten, dass die Zeichen paarweise auftreten. Der komplette Tag für den Button sieht also so aus:

```
<input type="button" value="Bild tauschen"
onClick="landschaft.src=
'bild2.jpg' ">
```

Wenn Sie jetzt die Webseite in einen Browser laden und auf den Button klicken, dann ersetzt der Browser wie gewünscht das angezeigte Bild.

Vor und zurück

Für den praktischen Einsatz ist dieser Effekt kaum zu gebrauchen. In einem zweiten Schritt erweitern Sie deshalb die HTML-Datei so, dass sich das Bild ohne Klicken schon beim Berühren mit dem Mauszeiger ändert. Verlässt der Cursor den Bildbereich, soll wieder das erste Bild erscheinen. Der Button ist dann nicht mehr nötig.

Der Lösungsweg führt auch hier über Event-Handler, denn der Browser soll auf Benutzeraktionen reagieren. Die zwei gebrauchten Handler sind nahelegend `onMouseOver` (Mauszeiger über dem Element) und `onMouseOut` (Mauszeiger verlässt den Bereich über dem Element). Ob Sie dabei die Groß- und Kleinschreibung übernehmen, spielt bei JavaScript-Befehlen keine Rolle. Die gemischten Groß- und Kleinbuchstaben haben sich lediglich eingebürgert, weil sich der Name leichter lesen lässt.

Damit das Bild bei Mausberührung reagiert, setzen Sie beide Event-Hand-



Hat der Browser die Webseite aufgebaut, erscheint das Startbild.

ler in das ``-Tag. Das sieht dann folgendermaßen aus:

```

```

Animiertes Menü

Für ein animiertes Menü, das verschiedene Bilder zeigt, wenn Sie die Links mit der Maus überfahren, passen Sie die Event-Handler ein wenig an. Beim Überfahren mit der Maus soll sich ein anderes Element der Webseite, das Bild eben, ändern.

Das zweite Beispiel benutzt ein einfaches Tabellenlayout mit zwei Spalten: links die Navigationsleiste, rechts

! ...und immer an die Surfer denken!

Bei aller Begeisterung für JavaScript und die schönen Effekte sollten Sie eines nicht vergessen: die Besucher Ihrer Webseite.

Viele Surfer schalten JavaScript wegen Sicherheitsbedenken ab oder, um lästige Pop-Up-Fenster zu vermeiden. Planen Sie Ihre Seiten deshalb so, dass auch Anwender ohne JavaScript auf die Inhalte zugreifen und die Navigation benutzen können. Geben Sie zumindest bei abgeschaltetem JavaScript eine entsprechende Meldung aus. Am einfachsten

geht das, wenn Sie auf der Startseite `index.htm` per JavaScript zur eigentlichen Startseite, etwa `start.htm`, weiterleiten.

- Ist JavaScript aktiv, dann klappt die Weiterleitung und der Surfer kann Ihre Hauptseite komplett nutzen.
- Bei abgeschaltetem JavaScript zeigt der Browser `index.htm`. Auf dieser Seite teilen Sie dann mit, dass Ihre Hauptseite ohne JavaScript nicht korrekt angezeigt wird. Die Meldung setzen Sie zwischen die Tags `<noscript>` und `</noscript>`. Damit stellen Sie sicher, dass der Surfer

auch bei einer verzögerten Weiterleitung die Warnung nicht sieht, obwohl er JavaScript aktiviert hat. Der Code dazu sieht folgendermaßen aus:

```
<body>
<script
language="JavaScript">
<!--
window.location.href=
"schritt1.htm";
-->
</script>
<noscript>
Leider haben Sie
JavaScript ...
</noscript>
</body>
```


der Seiteninhalt. Die Links sind wie üblich mit dem `<a>`-Tag gekennzeichnet. Wenn der Mauszeiger einen Link berührt, soll rechts das passende Bild erscheinen, beim Verlassen wieder das Startbild. Achten Sie darauf, dass alle Bilder dieselbe Größe haben. Da nicht alle Browser die Bildgröße anpassen, erscheint das neue Foto eventuell verzerrt.

Das Bild trägt diesmal den Namen `foto`. Entsprechend fügen Sie in die Link-Tags die zwei Event-Handler ein:

```
<a href="rachel.htm"
onMouseOver="foto.src='
rachelsee.gif'"
onMouseOut="foto.src='
startbild.gif'">
Rachel
</a>
```

Besucht ein Surfer ohne aktiviertes JavaScript diese Webseite, hat er kei-

ne Probleme. Zwar bekommt er die anderen Bilder nicht zu sehen, die Teilseiten kann er aber normal aufrufen.

Bilder vorladen

Die Webseite macht mit ihrer JavaScript-Animation nun schon einen guten Eindruck. Eine Überraschung erleben Sie aber, wenn Sie die Datei auf Ihren Web-Server hochladen und dann im Browser betrachten: Beim ersten Überfahren eines Links dauert es etliche Sekunden, bis rechts das Bild erscheint. Danach geht es so schnell wie gewohnt.

Was passiert hier? Nach dem Start des Skripts über den Event-Handler holt der Browser das neue Foto vom Server. Da die Bilder zwischen 40 und 50 KByte groß sind, kann das einige Sekunden dauern. Beim Test auf der eigenen Festplatte liegt die Ladezeit dagegen im Bereich von Millisekunden.

Die Lösung bietet ein zusätzliches Skript. Während ein Besucher die Informationen auf der Startseite liest, liegt die Online-Verbindung brach. In dieser Zeit lädt das Skript im Hintergrund die zusätzlichen Fotos. Sollen die Bilder dann angezeigt werden, sind sie bereits im Browser-Cache vorhanden und schnell zu sehen.

Um den Inhalt des Cache beim Internet Explorer zu prüfen, rufen Sie *Extras/ Internetoptionen* auf und klicken im Register *Allgemein* auf den Button *Einstellungen*. Mit *Dateien anzeigen* sehen Sie den Inhalt des Cache.

Skript automatisch starten

Die Aufgabe hierbei ist klar: Ein Skript soll sich beim Aufbau der Webseite im Browser automatisch aktivieren und vom Web-Server die fraglichen Bilder anfordern. Dabei benötigen Sie für die JavaScript-Befehle keine spezielle Stelle im HTML-Code, damit der Browser ein Skript automatisch ausführt.

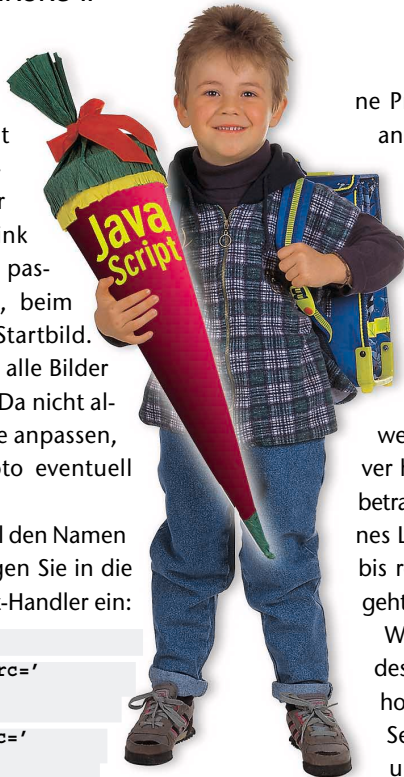
Der Browser weiß, dass es sich um ein Skript handelt, wenn Sie die Befehle mit folgendem Tag-Paar einschließen:

```
<script language="JavaScript">
<!--
Befehle
-->
</script>
```

Das Attribut `language` sagt dem Browser, um welche Skriptsprache es sich handelt. Die Kommentarzeichen `<!--` und `-->` sind eine Vorsichtsmaßnahme: Versteht ein Browser nichts von JavaScript, so interpretiert er die Befehle dazwischen als Kommentar und ignoriert sie.

In JavaScript gibt es keinen Befehl, um gezielt eine Bilddatei in den Browser-Cache zu laden. Deshalb wenden Sie einen Trick an: Sie erzeugen für jedes Bild eine Variable, einen Datenspeicher mit einem Namen. Dann füllen Sie jede Variable mit einer Kopie des Bildes vom Web-Server. Um diese Aufgabe zu erledigen, muss der Browser die Bilddatei online übertragen – damit landet sie automatisch im Cache. Die zwei Programmzeilen für ein Bild sehen so aus:

```
var bild1 = new Image();
bild1.src="jochberg.gif";
```



JavaScript-Bücher

Wir haben zwei Empfehlungen für JavaScript-Einsteiger und Fortgeschrittene.

Ein Klassiker für Fortgeschrittene ist das JavaScript Praxisbuch von Johannes Gamperl und Wolfgang Nefzger. Auf rund 550 Seiten haben die beiden Autoren die JavaScript-Grundlagen, Tipps und Tricks sowie viele Beispiele zusammengetragen. Zusätzlich liefert das Buch eine ausführliche JavaScript-Referenz zum Nachschlagen.

Insgesamt bietet das Buch Fortgeschrittenen viel Lesestoff.



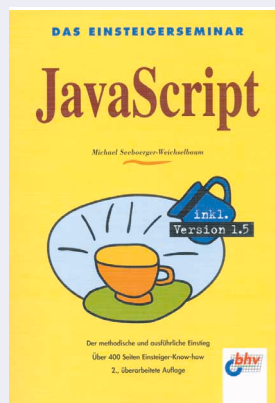
Das JavaScript Praxisbuch eignet sich gut für fortgeschrittene JavaScript-Fans.

■ Johannes Gamperl, Wolfgang Nefzger: JavaScript – Praxisbuch mit CD. Franzis' Verlag, 1999. ISBN 3-7723-7872-2. Preis: 89,95 Mark

Ideal für Einsteiger

Einsteiger sollten eher zum JavaScript-Seminar von Michael Seeboerger-Wechselbaum greifen. Das Werk liest sich flott, und Sie werden nicht mit Informationen überflutet, sondern langsam und verständlich an die Materie herangeführt. Anhand vieler Beispiele wie Pop-Up-Fenster oder Laufschriften lernen Sie die Grundlagen und erweitern Ihre Fähigkeiten in JavaScript. Bei dem Preis von 19,80 Mark fällt die Kaufentscheidung leicht.

JavaScript-Beginnern empfehlen wir das JavaScript-Einsteigerseminar.



■ Michael Seeboerger-Wechselbaum: Das Einsteigerseminar JavaScript. bhv-Verlag, 2001. ISBN 3-8266-72003. Preis: 19,80 Mark

Die erste Zeile definiert eine neue Variable namens `bild1`, die ein Foto (`new Image()`) aufnehmen soll. Vorher hat die Variable keinen Inhalt. Die zweite Zeile weist der Eigenschaft `src` der Variablen die Bilddatei zu. Diese Eigenschaft haben Sie ja bereits kennen gelernt.

Diese beiden Code-Zeilen wiederholen Sie für die restlichen Bilder. Achten Sie darauf, dass Sie den Namen der Variablen ändern (`bild2`, `bild3`, ...) und auch den Namen der Bilddatei jeweils anpassen.

Vielleicht haben Sie sich schon diese Fragen gestellt: Wenn Sie die zusätzlichen Bilder in eine Variable, also einen Datenspeicher im Browser laden, warum ist dann das Skript auf die Bilddateien im Cache angewiesen? Wäre es nicht sinnvoller, den Inhalt der Variablen direkt dem Bild auf der Webseite zuzuweisen? Etwa mit `foto.src=bild2.src` statt `foto.src='buchstein.gif'`.

Das ist logisch und funktioniert auch bei allen aktuellen Browsern – nur nicht bei Netscape 4.7. Hier pas-

Bäume und Elemente

Wenn Sie Elemente wie Bilder oder Formularfelder ansprechen wollen, lohnt es sich, mehr darüber zu wissen wie Browser arbeiten. Verstehen Sie das Prinzip, haben Sie einen Großteil von JavaScript im Griff.

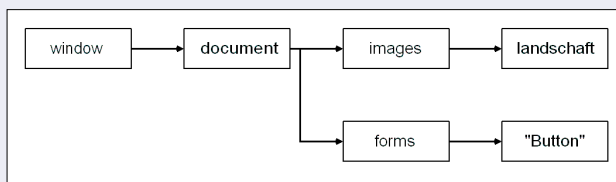
Das Browser-Fenster kann zahlreiche Elemente aufweisen: Webseiten, Frames, Links, Bilder, Formulare mit Eingabefeldern sowie Buttons, Anker und so weiter. Beim Aufbau der Seite erzeugt der

Browser sozusagen einen Baum, dessen Äste und Blätter die Elemente darstellen. Für die Beispielseite sieht der Baum wie abgebildet aus. Die oberste Ebene ist `window`, also das Browser-Fenster. Im Beispiel enthält das Fenster eine Webseite, das `document`. Auf der Seite gibt es für die Bilder eine eigene Gruppe `images`. Unter `images` ist das Beispieldokument mit seinem Namen `landschaft` aufgelistet. Der Button ist analog dazu in der Gruppe `forms` aufgeführt. Er hat keinen Namen, weil es für das kleine Beispiel

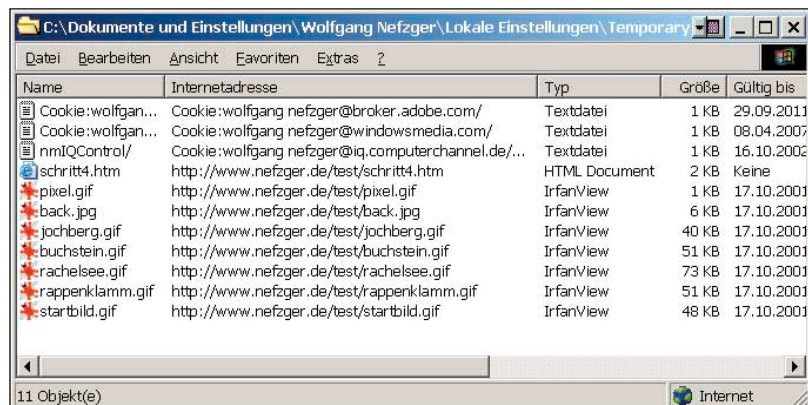
nicht notwendig ist. Um in diesem Schema ein Element anzusprechen, folgen Sie von der Wurzel bei `window` ausgehend den Pfeilen. Schreiben Sie die Namen der Elemente mit Punkten getrennt auf. Das Bild auf der Beispielseite hat in JavaScript also die Bezeichnung

`window.document.images.landschaft`

Wenn die Zuordnung eindeutig ist, begnügen sich die Browser mit der verkürzten Bezeichnung, im Zweifelsfall also mit dem Bildnamen ohne Zusatz, wie im Artikel verwendet.



Hier sehen Sie das Objektschema für eine Webseite mit einem Bild und einem Formular-Button.



Berührt der Mauszeiger den Link, erscheint rechts das passende Bild dazu.

siert gar nichts, und die Zuweisung bleibt wirkungslos.

Skript ans Ende

An welcher Stelle der HTML-Datei Ihrer Homepage bauen Sie nun das kurze Skript ein? HTML macht hier keine Vorschriften. Sie dürfen ein Skript in den Kopfbereich zwischen `<head>` und `</head>` einfügen oder irgendwo in den `<body>`-Container zwischen die HTML-Befehle für die Webseite. Es ist auch erlaubt, mehrere Skripts an verschiedenen Stellen einer HTML-Datei einzufügen.



Damit die Animation flüssig erscheint, müssen die Grafiken im Cache des Browsers liegen – hier im Internet Explorer.

Der beste Ort hängt davon ab, was Sie erreichen wollen. Im Beispiel kann es mit den großen Bildern 20 Sekunden oder länger dauern, bis alles geladen ist. Setzen Sie das Skript in den Kopfbereich, verzögert der Web-Browser den Seitenaufbau, bis alle Bilder geladen sind.

Besser ist es, das Skript an das Ende der HTML-Datei unmittelbar vor dem abschließenden `</body>`-Tag zu platzieren. In diesem Fall stellt der Browser die Webseite im Wesentlichen komplett dar und lädt anschließend im Hintergrund die Bilder. Es kann nur vorkommen, dass sich auch das Laden des Startbild auf der rechten Seite ein wenig verzögert. Texte sind aber bereits zu sehen.

Als Übungsmaterial haben wir das komplette Beispiel und alle Zwischenstufen inklusive Bilder für Sie auf der Heft-CD abgelegt. So können Sie am fertigen Objekt studieren, wie alles zusammenhängt. Spielen Sie auch mit dem Browser-Cache, und löschen Sie zwischendurch die darin enthaltenen Dateien. **Wolfgang Nefzger/ hs**