

3

Ein bisschen JavaScript

Das Internet hat sich in den vergangenen Jahren stark gewandelt. Vom reinen Informationsaustausch zwischen Wissenschaftlern ist das Internet zur multimedialen Spielwiese vieler geworden. HTML, die Grundlage der Dokumente im Internet, ist klassischerweise eine textorientierte Sprache. Zwar sind heute HTML-Seiten mit Frames, Grafiken, Animationen, Tabellen und Formularen eine Selbstverständlichkeit, aber die Wünsche der HTML-Programmierer lassen sich damit noch lange nicht befriedigen.

Im Jahr 1995 entwickelten die Unternehmen Netscape und Sun die Programmiersprache JavaScript. Was es mit dieser Programmiersprache auf sich hat, werden Sie in diesem Projekt sehen.

- Lektion 1** **Was ist JavaScript?**
- Lektion 2** **Funktionen**
- Lektion 3** **Eingabe- und Meldeboxen**
- Lektion 4** **Benutzerinformationen und andere nützliche Scripts**

Lektion 1

Was ist JavaScript?

JavaScript ist eine Programmiersprache, die speziell für den Einsatz im Internet konzipiert wurde. Ähnlich wie Java ist auch JavaScript plattform-unabhängig. Solange es einen Browser für ein Betriebssystem gibt, der JavaScript-fähig ist, können auf diesem Betriebssystem auch JavaScripts ausgeführt werden.

Das Schöne an JavaScript ist, dass es als objektorientierte Scriptsprache verhältnismäßig einfach zu handhaben ist. Besonders der hohe Verbreitungsgrad im Internet sorgt dafür, dass auch ungeübten Programmierern eine große Zahl fertiger JavaScripts zur Verfügung stehen.

Mit JavaScript können Sie z.B.:

- ▶ Elemente, wie Grafiken, Textblöcke usw., in einem HTML-Dokument während der Anzeige austauschen.
- ▶ Auf Benutzereingaben antworten. Z.B. kann ein Mausklick ausgewertet oder das Öffnen eines HTML-Dokuments gesteuert werden.
- ▶ Die Eingabe in ein Formular vor dem Verschicken überprüfen.

JavaScript-fähige Betriebssysteme

Für die folgenden Betriebssysteme existieren Browser der neueren Generation. Häufig handelt es sich dabei um Portierungen des Netscape Browsers.

- ▶ Windows95/98
- ▶ Windows NT/2000
- ▶ OS/2
- ▶ MacOS
- ▶ Linux
- ▶ Irix
- ▶ Solaris

Die Funktionsweise von JavaScript

JavaScript-Befehle werden innerhalb bestimmter HTML-Tags direkt in das HTML-Dokument geschrieben. Diese Befehle werden dann beim Interpretieren des HTML-Codes vom Browser als JavaScript-Befehle erkannt und ausgeführt, in der Reihenfolge, in der ein Browser ein HTML-Dokument ausliest.

Als objektorientierte Scriptsprache werden von JavaScript fast ausschließlich Objekte angesprochen. Objekte haben Eigenschaften und es gibt bestimmte Methoden, mit diesen Objekten umzugehen.

Mit Hilfe der Objekte lassen sich Funktionen erstellen, die anhand vorgegebener Ereignisse ausgeführt werden. Eine Funktion wiederum kann Einfluss auf das Aussehen Ihres HTML-Dokuments nehmen.

Lektion 2

Funktionen

Stellen Sie sich eine Funktion wie eine Werkzeugkiste vor. Sie legen alles hinein, was Sie zur Aufgabenbearbeitung benötigen. Dann tragen Sie die Werkzeugkiste an die Stelle, an der die Arbeit gemacht werden soll, und führen die Arbeit durch.

Anstelle von Hammer und Schraubendreher legen Sie Objekte, Eigenschaften, Variablen und Anweisungen in Ihre Funktionskiste. Ausführen können Sie diese Funktion zum Beispiel beim Laden eines HTML-Body-Elements oder wenn die Maus sich über eine Grafik bewegt. Nun gibt es Funktionen, die bereits im JavaScript-Standard enthalten sind, und andere, die Sie erst schreiben müssen oder fertig aus dem Internet holen. Zu den Standardfunktionen gehört z.B. `alert()`.

1. Fügen Sie in das Body-Element eines HTML-Dokuments das Attribut `onload` mit der ersten JavaScript-Funktion ein.

Tipp Die Funktion `alert()` wird ausgeführt, wenn das Body-Element geladen wird. Achten Sie immer auf die exakte Schreibweise.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Hallo Welt HTML-Dokument
    </TITLE>
  </HEAD>
  <BODY onload="alert('Hallo Welt')">
    <P>
      HTML-Dokument mit JavaScript Funktion.
    </P>
  </BODY>
</HTML>
```

Die erste JavaScript-Funktion ruft eine Dialogbox auf und zeigt den in Hochklammern dargestellten Inhalt an.

Der Text innerhalb der Dialogbox wird der Funktion innerhalb der runden Klammern in einfachen Anführungszeichen übergeben.

Modale Dialogboxen in JavaScript

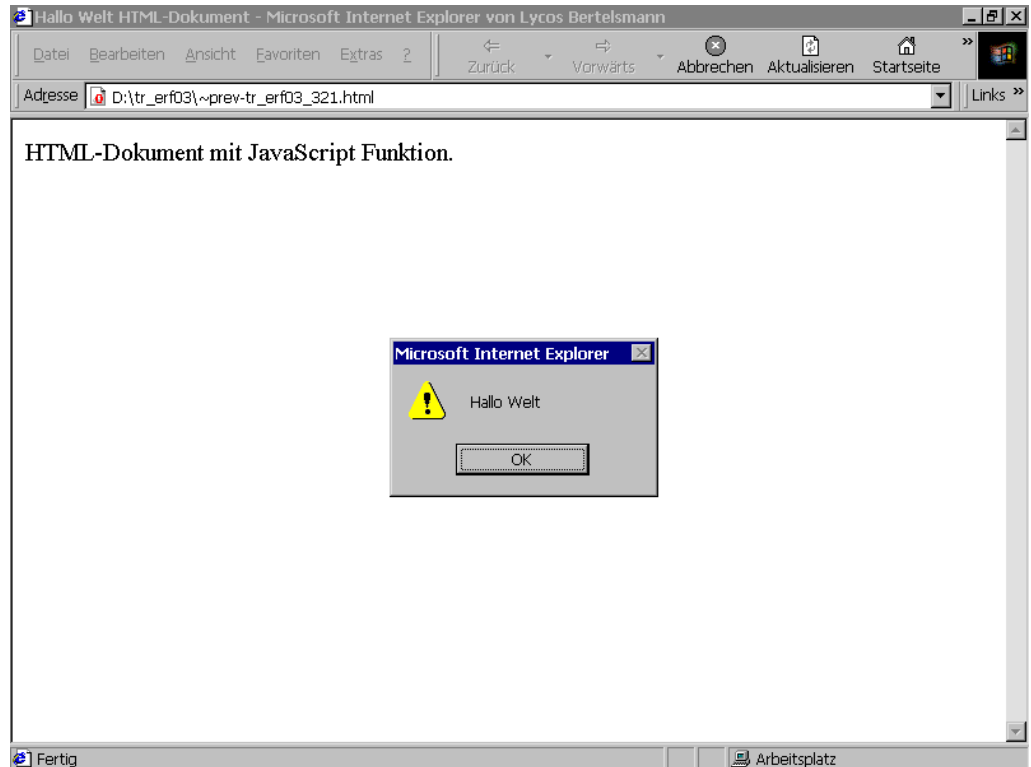
Ein Fenster, das den Focus auf sich zieht, bis es geschlossen wird, heißt modales Fenster. Bis ein solches modales Fenster geschlossen wird, stoppt die Abarbeitung folgender Programmzeilen.

Dialogbox	Beschreibung
<code>alert()</code>	Gibt eine Meldung aus.
<code>confirm()</code>	Erfragt eine Entweder-oder-Entscheidung.
<code>prompt()</code>	Erfragt einen Eingabewert.

Tabelle 3.1: Die modalen Fenstertypen, die mit Standardfunktionen in JavaScript erstellt werden können.

2. Betrachten Sie das Ergebnis im Browser.

Es entsteht eine durch JavaScript erzeugte Dialogbox mit dem Inhalt Hallo Welt und einem OK-Button. Die Dialogbox wird erst wieder geschlossen, wenn Sie auf den OK-Button klicken, dann wird auch erst das HTML-Dokument weiter geladen.



3. Bereiten Sie das Script vor.

```
<HEAD>
  <TITLE>
    JavaScript
  </TITLE>
<SCRIPT language="JavaScript" type="text/javascript">

</SCRIPT>
</HEAD>
```

Angabe der Scriptsprache für den Browser. So erkennt der Browser, dass er den folgenden Text als JavaScript zu interpretieren hat.

- ▶ JavaScript-Code wird traditionell zwischen dem Title-End-Tag und dem Head-End-Tag geschrieben.
- ▶ Mit dem Script-Element leiten Sie ein JavaScript-Programm ein.
- ▶ Der eigentliche Script-Code wird zwischen Start- und End-Tag des Script-Elements innerhalb eines Kommentar-Elements eingegeben.

4. Definieren Sie Ihre erste Funktion

Es reicht natürlich nicht aus, auf die wenigen Standardfunktionen von JavaScript zurückzugreifen. Sie wollen schließlich ganz spezielle Probleme mit JavaScript lösen. Dazu müssen Sie eigene Funktionen definieren. Hier unterscheidet sich JavaScript nicht von anderen Programmiersprachen.

Achtung Achten Sie darauf, jede Programmzeile mit einem Semikolon zu beenden, dies ist zwingend notwendig.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Die erste eigene Funktion
    </TITLE>
    <SCRIPT language="JavaScript" type="text/javascript">
      <!--

          function MeineErsteFunktion()
          {
            alert("Die Welt ist groesser");
          }

      // -->
    </SCRIPT>
  </HEAD>
  <BODY onload="MeineErsteFunktion()">
  </BODY>
</HTML>
```

Im Kommentar-Tag befindliche Funktion, die eine Standardfunktion alert() aufruft und ausführt. Diesmal wird nicht die Standardfunktion geladen, sondern Ihre eigene.

Die im Head-Element definierte Funktion wird im Body-Element aktiviert und ausgeführt.

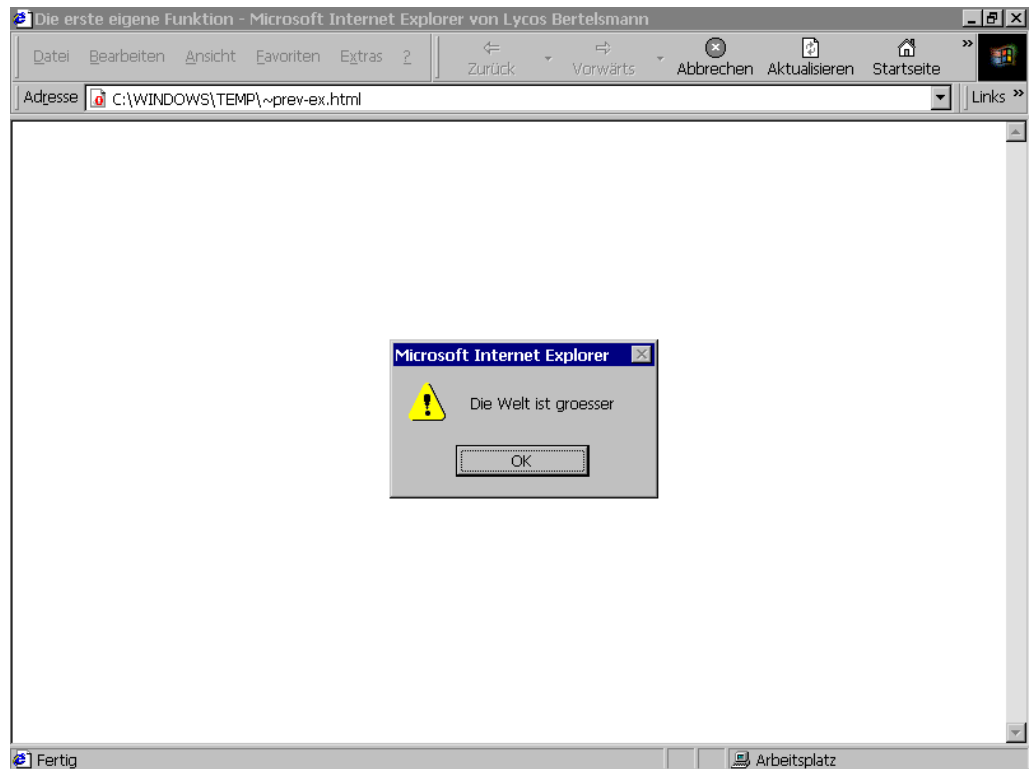
5. Probieren Sie die Funktion im Browser aus.

Die Funktion MeineErsteFunktion() wird beim Laden des Body-Elements ausgeführt. Diese Dialogbox wird durch eine Funktion gestartet, und Sie müssen auf OK klicken, um die Box wieder zu schließen.

Die runden Klammern nach dem Namen der Funktion enthalten die Übergabeparameter (in diesem Fall sind keine vorhanden). Zwischen den geschweiften Klammern {} befindet sich der Code der Funktion.

Die Funktion `alert` erzeugt eine Dialogbox mit einer *OK*-Schaltfläche und einem Ausrufezeichen. Der in der Dialogbox ausgegebene Text wird als Parameter übergeben. Da es sich um einen Text handelt, wird er in Anführungszeichen ('text') eingeschlossen.

Der JavaScript-Code wird durch die schließende HTML-Markierung `</SCRIPT>` abgeschlossen. Der Vorteil einer selbst definierten Funktion ist, dass Sie mehr als nur eine Aktion, Berechnung oder Funktion auf einmal abarbeiten können.



Variablen

Ohne Variablen ist das Programmieren nicht denkbar, sie sind unverzichtbare Datenspeicher. Variablen können Sie sich wie kleine Aufbewahrungsdosen in der Werkzeugkiste vorstellen. Je nach Bedarf legen Sie dort die verschiedensten Dinge so lange ab, bis sie benötigt werden. Variablen müssen immer zuerst definiert werden. Den Namen einer Variablen können Sie frei wählen.

1. Fügen Sie eine Variable ein, die den Text an die Funktion übergibt.

- ▶ Ändern Sie den Namen der Funktion.
- ▶ Definieren Sie die Variable Ausgabe mit dem JavaScript-Schlüsselwort `var`.

Unzulässige Zeichen

In Funktionen und Variablen sind einige Zeichen unzulässig. Es handelt sich dabei um die Zeichen:

Ä ä Ü ü Ö ö ß * + - ! " : \$ % & / () = ? \ @ ~ # , . ; ^ " - < > | ,

Tipp Buchstaben werden bei der Wertübergabe immer in Anführungszeichen gesetzt. Variablen werden nicht in Anführungszeichen gesetzt.

- ▶ Weisen Sie der Variablen den Wert "Die Welt ist noch groesser" zu.
- ▶ Übergeben sie den Wert der Variablen der Funktion alert() als Parameter.

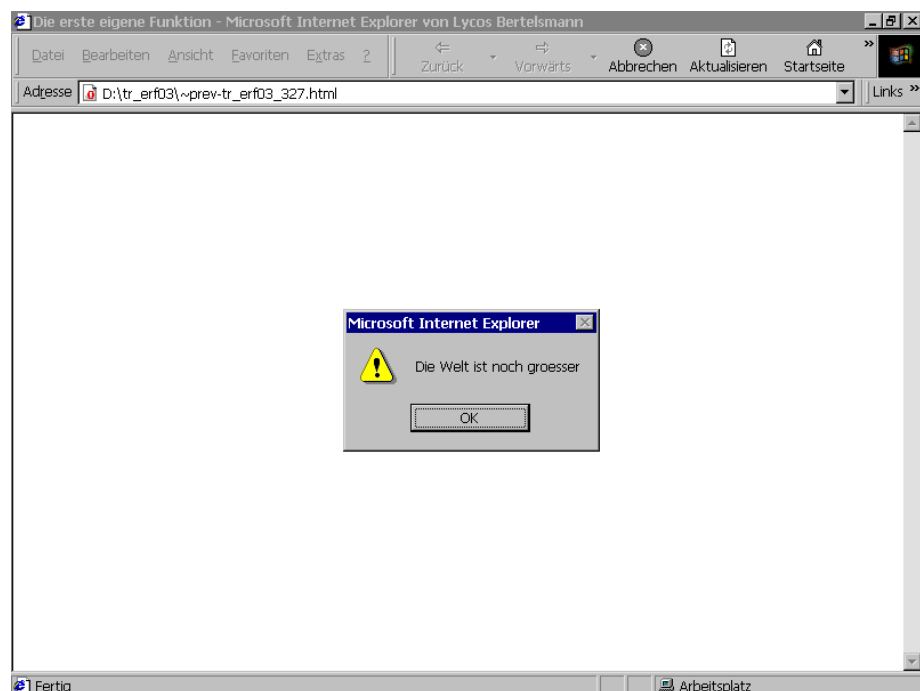
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Die erste eigene Funktion
    </TITLE>
    <SCRIPT language="JavaScript" type="text/javascript">
      <!--

        function MeineZweiteFunktion()
        {
          var Ausgabe;
          Ausgabe="Die Welt ist noch groesser";
          alert(Ausgabe);
        }

      // -->
    </SCRIPT>
  </HEAD>
  <BODY onload="MeineZweiteFunktion()">
  </BODY>
</HTML>
```

Eine Funktion mit Variable, die den Text der Dialogbox aufnimmt und später an die Funktion alert() übergibt.

2. Analysieren Sie das Ergebnis im Browser.



Im Browser ist kein Unterschied zu sehen, obwohl die Dialogbox mit variablem Inhalt erstellt wird. Wie Sie der Funktion `alert()` die benötigten Werte zukommen lassen, spielt bei der Darstellung der Dialogbox keine Rolle.

Lektion 3

Eingabe- und Meldeboxen

Die Möglichkeit, vom Besucher Daten zu erfragen und zur Laufzeit eines Programms auszuwerten, ist eines der Einsatzgebiete von JavaScript.

JavaScript stellt zur Datenabfrage die Standardfunktion `prompt()` zur Verfügung. Die daraus resultierende Dialogbox besitzt zwei Buttons: *OK* und *Abbrechen*.

1. Erstellen Sie ein JavaScript zur Datenabfrage.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Die erste eigene Funktion
    </TITLE>
  <SCRIPT language="JavaScript" type="text/javascript">
    <!--

        function DatenAbfrageFunktion()
        {
            var Eingabe;
            Eingabe = prompt("Bitte geben Sie Ihr Alter ein >");
            alert("Sie haben Ihr Alter mit: " + Eingabe +
                " angegeben");
        }

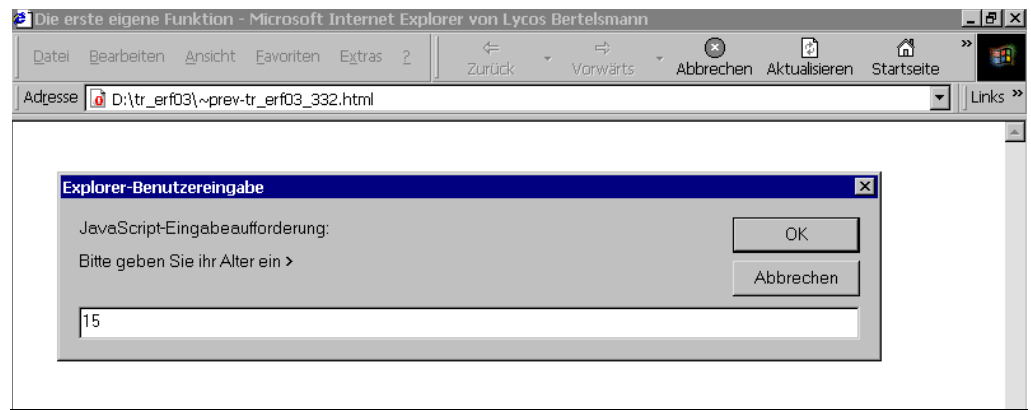
    // -->
  </SCRIPT>
  </HEAD>
  <BODY onload="DatenAbfrageFunktion()">
  </BODY>
</HTML>
```

HTML-Dokument mit Datenabfrage und ergebnisanzeigender Dialogbox.

2. Laden Sie das HTML-Dokument in den Browser.

Es erscheint eine Dialogbox der Funktion `prompt()` zur Abfrage des Alters. In der Abbildung ist bereits ein Wert eingegeben.

Tipp Die Standardfunktion `prompt()` öffnet eine Dialogbox zur Dateneingabe.



3. Bestätigen Sie mit **OK** und betrachten Sie das Ergebnis.



Die Funktion `alert()` gibt die zusammengebaute Antwort zurück. Diese Antwort besteht aus den eingegebenen Daten und festgelegtem Text.

Verzweigungen oder die If-Anweisung.

Eine der häufigsten Anwendungen für Programme ist die Auswahl zwischen zwei oder mehr Zuständen: »Wenn dies so ist, dann tue das, ansonsten mache etwas anderes.«

1. Erstellen Sie ein HTML-Dokument mit einer JavaScript-Funktion zum automatischen Erkennen des Browsers.

Dabei gehen Sie in folgenden Schritten vor:

- ▶ Funktion `Auswahl()` erstellen.
- ▶ Variable zum Aufnehmen der Browser-Namens definieren.
- ▶ Variable zum Aufnehmen der URL für die Netscape-Seite definieren.

Zusammengesetzte Strings

Sie haben bereits gesehen, dass einzelne Textblöcke aus Variablen und Texten in Reinschrift miteinander kombiniert werden können. Hier noch ein paar Grundregeln zum Umgang mit Strings. Strings sind aus Buchstaben und Zahlen bestehende Daten, die keiner mathematischen Berechnung unterliegen.

- ▶ Text, den Sie direkt im Script eingeben, wird immer in Anführungszeichen gesetzt.
- ▶ Variablen erhalten keine Anführungszeichen.
- ▶ Einzelne Textblöcke können mit dem Pluszeichen `+` hintereinander gestellt werden.

Beispiel:

```
alert("Sie haben Ihr Alter mit: " + Eingabe " angegeben");
```

- ▶ Variable zum Aufnehmen der URL für die Internet Explorer-Seite definieren.
- ▶ If-Anweisung, um zu überprüfen, ob es sich um einen Netscape-Browser handelt.
- ▶ Aktion definieren, die ausgeführt wird, wenn die If-Bedingung zutrifft.
- ▶ URL-Angabe zu der zu ladenden Datei in die Variablen setzen.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Die erste eigene Funktion
    </TITLE>
  <SCRIPT language="JavaScript" type="text/javascript">
    <!--
```

```
      function Auswahl()
      {
        var Browser_name=navigator.appName;
        var Netscape_seite="netscape.html";
        var IExplorer_seite="ie.html";
        if (Browser_name=="Netscape")
          {
            window.location.href=Netscape_seite;
          }
        else
          {
            window.location.href=IExplorer_seite;
          }
      }
    -->
```

```
  </SCRIPT>
</HEAD>
<BODY onload="Auswahl()">
</BODY>
</HTML>
```

JavaScript-Funktion in einem HTML-Dokument, welches den Browser erkennt und ein entsprechendes HTML-Dokument aufruft.

2. Erstellen Sie die Seite für den Microsoft Internet Explorer.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
```

```

        Ms Internet Explorer
    </TITLE>
</HEAD>
<BODY>
    <H1>
        Ms Internet Explorer Seite
    </H1>
</BODY>
</HTML>

```

HTML-Dokument für die Anzeige mit dem Microsoft Internet Explorer.

Achtung Achten Sie unbedingt auf die richtige Schreibweise, um Fehler zu vermeiden, die sonst erst langwierig gesucht werden müssen.

3. Erstellen Sie die Netscape-Seite.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Netscape
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      Netscape Seite
    </H1>
  </BODY>
</HTML>

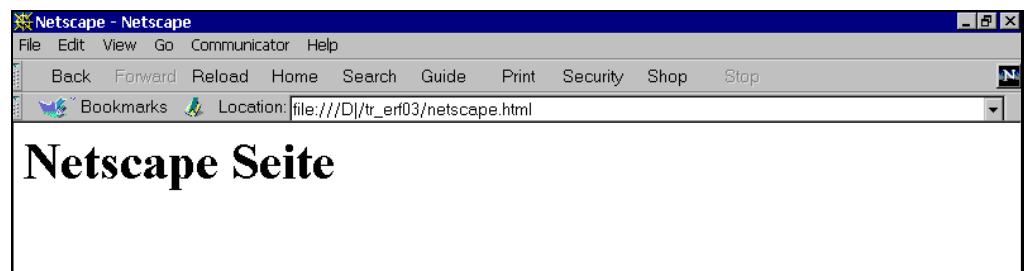
```

HTML-Dokument, das angezeigt wird, wenn Sie den Netscape-Browser benutzen.

Achtung Achten Sie darauf, dass alle drei HTML-Dokumente im selben Verzeichnis vorhanden sein müssen.

4. Betrachten Sie die Script-Seite mit dem Netscape-Browser.

Das JavaScript hat den Browser richtig erkannt und das speziell für den Netscape-Browser entworfene Dokument geöffnet.



5. Betrachten Sie nun dasselbe Dokument mit dem Internet Explorer.



Da die erste Bedingung nicht Wahr ist, wird die mit `else` eingeleitete Alternative ausgeführt: Das JavaScript hat den Browser korrekt erkannt und die richtige Datei in das Window-Objekt geladen.

Lektion 4

Benutzerinformationen und andere nützliche Scripts

So schön das Arbeiten mit einer Programmiersprache wie JavaScript auch ist: JavaScript läuft auf dem Client-Rechner, der die Seite darstellt. Informationen und die Berechnungen lassen sich daher einfach im Browser darstellen. Der umgekehrte Weg ist dagegen nicht so leicht.

Ganz wichtig ist es, dass der Benutzer Ihrer Webseiten seinem Browser erlaubt hat, JavaScript auszuführen. Es gibt immer mehr Menschen, die Plug-Ins wie Java oder JavaScript deaktivieren. Deshalb sollten Sie nicht zu viel JavaScript in den wichtigen Navigationsebenen einsetzen. Trotzdem einige Beispiele.

1. Erstellen Sie ein Script, das die aktuellen Werte von Uhrzeit und Datum ermittelt.

Dabei gehen Sie folgendermaßen vor:

- ▶ Definieren Sie die Funktion `DatumZeit()`.
- ▶ Erstellen Sie das neue Objekt `zeit`.
- ▶ Definieren Sie Variablen für alle Zeit- und Datumswerte.
- ▶ Fügen Sie Anweisungen ein, die es dem JavaScript ermöglichen, in das HTML-Dokument zu schreiben, und geben Sie den zu schreibenden Text an.
- ▶ Rufen Sie die Funktion `DatumZeit()` beim Laden des Body-Elements auf.

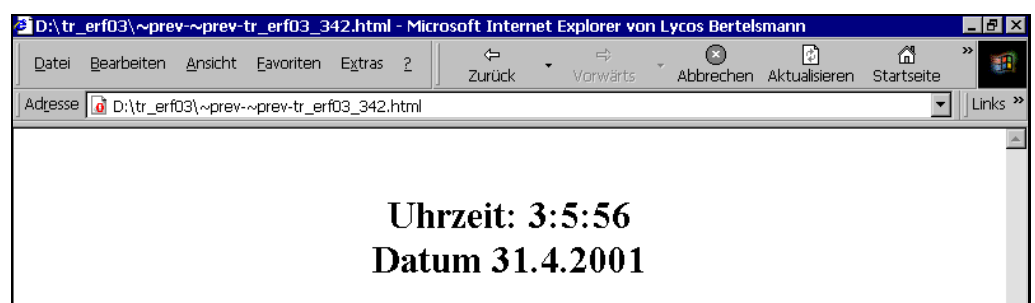
Tipp Sonderzeichen wie Anführungsstriche, Schrägstriche usw. gehören zu den unzulässigen Zeichen. Wenn Sie diese Zeichen trotzdem eingeben müssen, so setzen Sie einen Backslash \ direkt vor das Sonderzeichen. Der Backslash hebt die Sonderfunktion eines Zeichens auf und ermöglicht das Anzeigen dieses Zeichens.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Datum und Uhrzeit ermitteln.
    </TITLE>
    <SCRIPT language="JavaScript" type="text/javascript">
      <!--
        function DatumZeit()
        {
          var zeit = new Date();
          var stunden = zeit.getHours();
          var minuten = zeit.getMinutes();
          var sekunden = zeit.getSeconds();
          var tag = zeit.getDate();
          var monat = zeit.getMonth();
          var jahr = zeit.getFullYear();
          var datum = tag+"."+monat+"."+jahr;
          var uhrzeit = stunden +":"+minuten+":"+sekunden;
          window.document.write("<div align=\"center\">
            <H2> <br>");
          window.document.write("Uhrzeit: "+uhrzeit+"<br>");
          window.document.write("Datum "+datum);
          window.document.write("<\/h2><\/div>");
        }
      // -->
    </SCRIPT>
  </HEAD>
  <BODY onload="DatumZeit()">
  </BODY>
</HTML>
```

Dieses HTML-Dokument zeigt das Datum und die Uhrzeit an.

2. Betrachten Sie das Ergebnis im Browser.

Tipp Da JavaScript auf dem Clientrechner ausgeführt wird, handelt es sich hierbei immer um die Systemzeit des Besuchers, nicht des Servers.



Die aktuelle Systemzeit und das Datum werden ausgelesen. Beide Werte werden mittels HTML-Tags zentriert als Überschrift zweiter Ordnung dargestellt.

3. Erzeugen Sie wechselnde Bilder.

Sicher sind Sie schon häufig diesen animierten Hyperlinks im Internet begegnet. Wenn Sie mit der Maus über einen Link fahren, ändert der Button z.B. seine Farbe, oder eine Zusatzinformation wird so lange eingeblendet, wie die Maus über dem Hyperlink verweilt.

Dahinter steckt ein JavaScript.

- ▶ Erstellen Sie die neuen Objekte `overbild1` und `outbild1` usw.
- ▶ Weisen Sie den Objekten die Eigenschaft `src` (source, engl. Quelle) zu. Der Name wird in Anführungszeichen angegeben und beinhaltet den kompletten URL des Bilds.
- ▶ Definieren Sie die Funktion `Bildwechsel` mit den Übergabeparametern `Bildnr` und `Bildobjekt`.
- ▶ Rufen Sie das ausgewählte Image-Objekt auf.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Wechselnde Grafiken
    </TITLE>
  <SCRIPT language="JavaScript" type="text/javascript">
    <!--
      overbild1 = new Image();
      overbild1.src = "center2.jpg";

      outbild1 = new Image();
      outbild1.src = "center1.jpg";

      function Bildwechsel(Bildnr,Bildobjekt)
      {
        window.document.images[Bildnr].src = Bildobjekt.src;
      }
    //-->

  </SCRIPT>
</HEAD>
<BODY>
  <A href="index.shtml" onmouseover="Bildwechsel(0,overbild1)"
    onmouseout="Bildwechsel(0,outbild1)"><IMG src="center1.jpg"
    width="342" height="266" border="0" alt="Extra Info Bilder">
  </A>
</BODY>
</HTML>
```

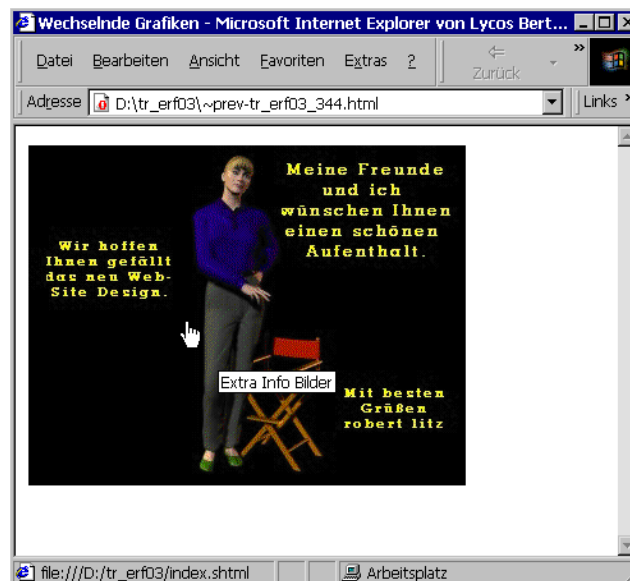
Ein HTML-Dokument mit wechselndem Bild bei mouseover als Hyperlinkanimation.

4. Betrachten Sie das HTML-Dokument im Browser.



Die HTML-Seite mit JavaScript-animiertem Bildwechsel bei mouseover. Hier befindet sich die Maus neben der Grafik.

5. Fahren Sie mit der Maus auf die Grafik.



Achtung Achten Sie auf übereinstimmende Bildgrößen, sonst wirkt der Wechsel eher wie ein Wackeln.

Tipp Denken Sie daran, dass JavaScript-Funktionen nur ausgeführt werden, wenn der Besucher JavaScript aktiviert hat.

Sie sehen, wie die Grafiken ausgetauscht werden: Das JavaScript hat das Original ersetzt. Der Mauszeiger ändert sein Aussehen beim Überfahren des Hyperlinks und löst den Bildwechsel aus.

Zwei Frames gleichzeitig ändern

Ähnlich wie die Mausaktionen können Sie auch andere Ereignisse auf den Webseiten steuern. Sie benötigen einige HTML-Dokumente für diese Aufgabe.

1. Erstellen Sie ein Verzeichnis, in das Sie alle Dokumente speichern.**2. Erstellen und speichern Sie RAHMEN.HTML.**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
<HTML>
  <HEAD>
    <TITLE>
      Frameset Datei
    </TITLE>
  </HEAD>
  <FRAMESET cols="30%,*">
    <FRAME src="verweise.html">
    <FRAMESET rows="25%,*">
      <FRAME src="hallo1.html">
      <FRAME src="hallo2.html">
    </FRAMESET>
  </FRAMESET>
</HTML>
```

Die Datei *rahmen.html* stellt das Gerüst der Frames dar.

Tipp Frame- und auch Image-Zählungen beginnen immer bei Null, also ist der Frame 1 der zweite Frame ab Anfang des Dokuments.

3. Erstellen Sie die Datei VERWEISE.HTML.

- ▶ Erstellen Sie eine Funktion mit nachfolgenden Parametern.
 - URL1 ist die Datei, die in den zweiten Frame geladen werden soll.
 - F1 ist der zu ladende Frame, in diesem Fall handelt es sich um den zweiten Frame.
 - URL2 ist die Datei, die in den dritten Frame geladen werden soll.
 - F2 ist der mit der Datei URL2 zu ladenden Frame (hier der dritte Frame).
- ▶ Weisen Sie jedem Frame das neue Dokument zu.
- ▶ Erstellen Sie die benötigten Anchor-Elemente.
- ▶ Anstelle eines Pfades rufen Sie ein Java-Script als Wert des href-Attributs auf.
- ▶ Geben Sie die benötigten Übergabeparameter URL1, F1, URL2 und F2 an.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Verweise
    </TITLE>
  <SCRIPT language="JavaScript" type="text/javascript">
    <!--
    function ZweiFrames(URL1,F1,URL2,F2)
    {
      parent.frames[F1].location.href=URL1;
```

```

        parent.frames[F2].location.href=URL2;
    }
    //-->
</SCRIPT>
</HEAD>
<BODY text="#000000" link="#0000CC" vlink="#000099"
alink="#0000FF">
    <H1>
        Verweise
    </H1>
    <A href="javascript:ZweiFrames('elfriede1.
html',1,'elfriede2.html',2)">Elfriede</A> <BR>
    <A href="javascript:ZweiFrames('erwin1.html',1,
'erwin2.html',2)">Erwin</A>
</BODY>
</HTML>

```

Die Datei verweise.html befindet sich im rechten Rahmen und beinhaltet das Java-Script sowie die Navigationsleiste.

4. Erstellen und speichern Sie HALLO1.HTML.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <TITLE>
            Hallo 1
        </TITLE>
    </HEAD>
    <BODY>
        <H1>
            Hallo, hier ist das Topfenster
        </H1>
    </BODY>
</HTML>

```

Die Datei hallo1.html wird beim Laden des Framesets oben rechts angezeigt.

5. Erstellen und speichern Sie HALLO2.HTML.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <TITLE>
            Hallo 2
        </TITLE>
    </HEAD>
    <BODY>
        <H1>

```

```
Hallo, hier ist das Hauptfenster
</H1>
</BODY>
</HTML>
```

Die Datei hallo2.html wird beim Laden des Framesets rechts unten angezeigt.

6. Erstellen und speichern Sie Die Datei ERWIN1.HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Erwin 1
    </TITLE>
  </HEAD>
  <BODY>
    <IMG src="erwin1.jpg" width="150" height="156" border="0">
  </BODY>
</HTML>
```

Die Datei erwin1.html beinhaltet ein Foto und wird oben rechts angezeigt.

7. Erstellen und speichern Sie die Datei ERWIN2.HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Erwin 2
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      das ist Erwin im Alter von 2 Jahren
    </H1>
  </BODY>
</HTML>
```

Die Datei erwin2.html wird in den unteren rechten Rahmen geladen.

8. Erstellen und speichern Sie die Datei ELFRIEDE1.HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      elfriede 1
    </TITLE>
  </HEAD>
```

```

<BODY>
  <IMG src="elfriede1.jpg" width="150" height="208" border="0">
</BODY>
</HTML>

```

Die Datei *elfriede1.html* wird in den Toprahmen geladen.

9. Erstellen und speichern Sie die Datei ELFRIEDE2.HTML.

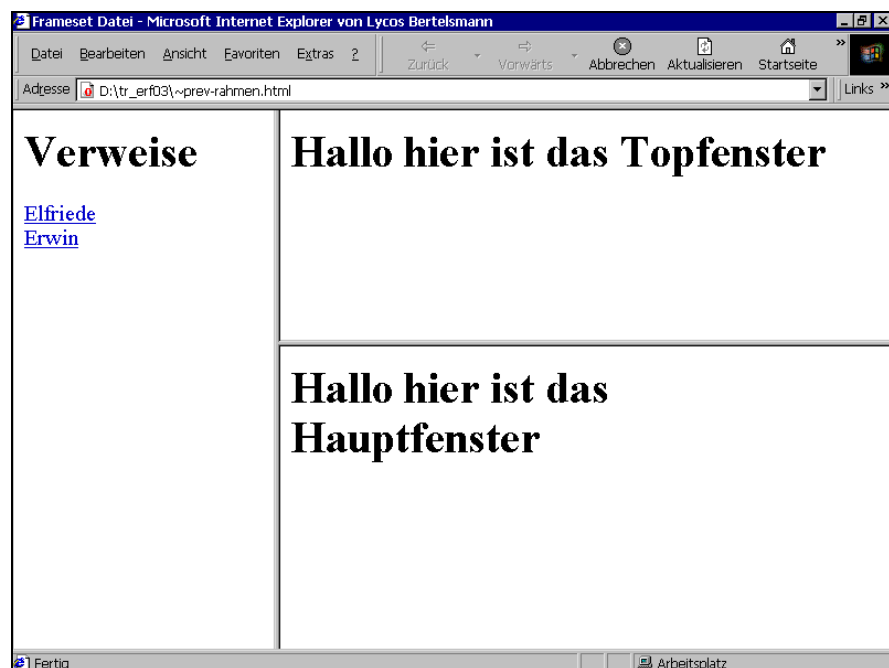
```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>
      Elfriede 2
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      Hier ist Elfriede im Alter von 105 Jahren
    </H1>
  </BODY>
</HTML>

```

Die Datei *elfriede2.html* wird im unteren rechten Rahmen geladen.

10. Betrachten Sie das erstellte Frameset im Browser.



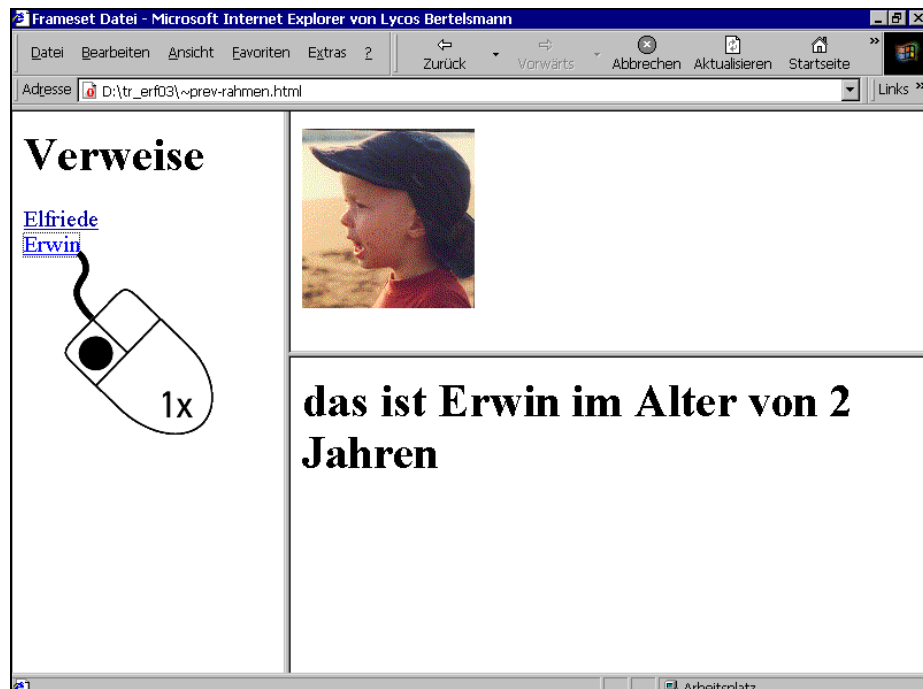
Sie sehen das Frameset-Dokument, bei dem der Hyperlink beide rechten Rahmen neu lädt.

11. Klicken Sie auf *Elfriede*.



Nach dem Klick auf den Hyperlink *Elfriede* ist das Dokument mit dem Bild sowie das Dokument mit dem Text geladen worden.

12. Klicken Sie auf *Erwin*.



Der Klick auf den Hyperlink *Erwin* bringt eine andere Bild-Textkombination in die beiden rechten Rahmen.



Testen Sie Ihr Wissen!

1. Ist JavaScript nur auf Linux-Betriebssystemen nutzbar?
2. Welche Standardfunktionen von JavaScript kennen Sie?
3. Welche Aufgabe haben Funktionen?
4. Was macht Variablen variabel?
5. Wie lassen sich Zeichenketten (Strings) erstellen?
6. Müssen Sie immer den Quellcode der Java-Scripts selber tippen?
7. Lassen sich Animationsaufgaben mit JavaScript bewerkstelligen?

