

Unterschiede Java - Javascript¹

JavaScript wird anders als Java direkt in den HTML-Code geschrieben. Für Java ist es notwendig, mit dem JDK (Java Development Kit) oder einem grafischen Tool zu programmieren. Dann wird der Code kompiliert und das Applet liegt als Bytecode in einem so genannten class-File vor.

- ⇒ Für JavaScript ist es nicht notwendig, den Code zu kompilieren. Es werden keine separaten Dateien wie bei Java (*.java und *.class) benötigt.
- ⇒ Ein JavaScript läuft immer nur in einem Web-Browser ab. Dies ist ähnlich wie bei Java-Applets. Eigenständige Applikationen, die auch mit Java möglich sind, sind derzeit mit JavaScript nicht möglich.
- ⇒ JavaScript kann ganz einfach analysiert werden. Der JavaScript-Code steht direkt im HTML-Code der Web-Seite. Er steht dort im Klartext, während ein Java-Applet oder eine Java-Applikation als class-Datei überspielt wird, die man nicht analysieren kann. Hat man dazu nicht den originalen Java-Sourcecode, kann man sich die Tricks der Programmierer nicht anschauen.
- ⇒ Im Gegensatz zu Java ist kein JDK oder Ähnliches erforderlich. Selbst ein grafisches Entwicklertool ist nicht notwendig; die Codes sind relativ einfach und können direkt in den HTML-Code geschrieben werden.
- ⇒ JavaScript ist in vielen Teilen einfacher konzipiert als Java. Beispielsweise werden Variablen pauschal über *var* definiert - unabhängig davon, ob in dieser Variable eine Zahl, ein Zeichen oder ein Wahrheitswert gespeichert wird. Ebenso sind die Funktionsdefinitionen einfacher als in Java. Dort werden keine komplizierten Klassen, Schnittstellen und Methoden erzeugt. Es gibt keine Unterscheidung der Sichtbarkeit nach *public*, *protected*, *private* und *private protected*.

Dies sind die wichtigsten Unterschiede zwischen Java und JavaScript. Wie bei Java können Sie auch in JavaScript Variablen benutzen, Methoden aufrufen, Objekte einsetzen usw. Sie werden dies in diesem Kapitel noch sehen. Dabei lernen Sie, dass es im Prinzip ganz einfach ist, ein JavaScript zu erstellen.

Was kann man nicht mit Javascript machen?

JavaScript besitzt einige deutliche Beschränkungen. Mit JavaScript ist Folgendes nicht möglich:

- ⇒ JavaScript kann keine Dateien dauerhaft auf der Festplatte des Internet-Surfers speichern (Ausnahme: Cookies).
- ⇒ JavaScript kann auch keine Dateien dauerhaft auf dem Server speichern. Bekannte Anwendungen wie Gästebücher, Chatrooms oder Besucherzähler sind mit JavaScript nicht möglich.
- ⇒ JavaScript kann nicht dauerhaft den Inhalt von HTML-Dateien ändern. Es ist nur möglich, dynamisch den Inhalt von HTML-Dateien zu verändern.
- ⇒ Eine Anbindung an eine Datenbank ist mit JavaScript nicht möglich. Es können keine Datenbanken via JavaScript abgefragt werden.

¹ aus: Michael Seeboerger-Weichselbaum, Javascript, S. 26 ff

Was kann man mit JavaScript alles machen?

Sie können Farben verändern, Text ausgeben und Fenster anzeigen lassen, die Menüs und Dialogboxen enthalten. Mit JavaScript ist sehr vieles möglich. Typische Anwendungen, die in JavaScript erstellt werden, sind:

- Buttons verändern sich, wenn die Maus darüber gefahren wird (so genannter Rollover-Effekt).
- Laufschriften, die in der Statusleiste oder in einem Formularfeld durchlaufen.
- Kleine Popup-Fenster, die separaten Inhalt enthalten können.
- Dialogboxen, die zur Eingabe von Daten genutzt werden können.
- Auswertung von Formularen
- Automatische Erkennung des Browsers.

[Ein Beispiel für die Möglichkeiten von Javascript]:

Der Windows-Explorer zeigt die Verzeichnisstruktur der Festplatte und der darin enthaltenen Dateien an. Dies gibt es auch als freies JavaScript für die Homepage. Hier wird jedoch nicht der Inhalt der Festplatte angezeigt, sondern dieser »Homepage-Explorer« dient der besseren Navigation in Web-Seiten (<http://www.ivanpeters.com/>).

Vorsicht beim Addieren von Zahlen, die keine sind!

Eine weitere teuflische Fehlerquelle lauert, wo Zahlen als Text gespeichert sind und mit „+“ addiert werden sollen. Lassen Sie den Benutzer beispielsweise per Formular Zahlen a und b eingeben, dann ist diese Eingabe immer erst einmal als Text gespeichert. Schreiben Sie nun a+b, dann werden die Zahlen einfach hintereinander geschrieben statt richtigerweise addiert - siehe oben das Beispiel 5+“7”=“57”. Wenn Sie das wiederum auf dem Formular anzeigen, sehen Sie nicht einmal, dass der Browser es als Text betrachtet - und kommen so natürlich zu völlig falschen Ergebnissen. Deshalb müssen Sie Zahleneingaben aus Formularen immer vor der Weiterverarbeitung mit **parseInt()** oder **parseFloat()** in echte Zahlvariablen umwandeln.

(Driescher S. 119)

```
<html>
<head>
  <title>Multiplizieren</title>
<script language="JavaScript" type="text/javascript">
<!--
var a, b, ergebnis;
a=0
b=0
a=prompt("Bitte tragen Sie Zahl a ein!","");
b=prompt("Bitte tragen Sie Zahl b ein, sie wird zu a addiert","");
a=parseFloat(a);
b=parseFloat(b);
ergebnis=(a+b);
//-->
</script>
</head>
<body bgcolor="white">
<h1>Multiplizieren</h1>
<script language="JavaScript" type="text/javascript">
<!--
document.write("Das korrekte Ergebnis lautet " + ergebnis + "!");
//-->
</script>
</body>
</html>
```

Bei der Addition **5+6** kommt jetzt **11** heraus!

Aufgabe: Erstellen Sie ein HTML-Dokument mit Javaskript-Elementen.

- Der Benutzer soll beim Betreten der Website mit einer Meldung begrüßt werden (Eventhandler onload)
- Statuszeilenmeldung bei Berührung eines Links
- Die Seite soll einen Button enthalten, mit dem der Benutzer $12 \cdot 28$ rechnen kann
- Bauen Sie zwei Austausch-Grafiken ein.

Aufgabe: Erstellen Sie das folgende Dokument namens **Var_test_01.html** ²

```
<html>
<head>
<title>Form-Test</title>

<script language="JavaScript" type="text/javascript">
function addieren(v1,v2)
    {
    var sum=0;
    var va1, va2;
    va2=parseInt(v2); /* Werte als Integer */
    va1=parseInt(v1);
    sum= va1 + va2;
    document.write("Ergebnis =" +sum);
    }
</script>
</head>
<body>
<form name="Formular" action="">
Geben Sie die erste Zahl ein: <input type="text" name="val1" size="3">
Geben Sie die zweite Zahl ein: <input type="text" name="val2" size="3">
<input type="button" value="Addieren" onClick="addieren(val1.value,val2.value)">
<br>
</form>
</body>
</html>
```

Dieses Beispiel dient dazu, zu zeigen, wie die Eingabe von Variablen über ein Formular entgegengenommen werden kann. Statt in die Eingabeaufforderungsfenster zu schreiben, wird der Wert direkt in ein Formularfeld eingegeben.

Durch den Button werden die Werte an die Funktion übergeben, in der festgelegt ist, was mit den Zahlen geschehen soll.

```
onClick="addieren(val1.value,val2.value)
```

Durch diese Zeile werden die Werte an die Funktion übergeben. Die Funktion **addieren** hat zwei Parameter, die oben bei der Deklaration der Funktion genannt wurden.

```
function addieren(v1,v2)
```

Durch die onClick-Zeile wird übergeben: **val1.value** und **val2.value** .

Das sind dann die **v1** und **v2**, die in der Funktion verwendet werden. Sie müssen zuerst zu Zahlenwerten umgewandelt werden (**va2=parseInt(v2)**), weil die Werte, die aus Formularen stammen, zuerst automatisch Zeichenketten sind.

² Vielen Dank, Herr Kangowski!

Beispiel aus Selfhtml Werte aus Formularen übernehmen

```
<html>
<head>
<title>Test</title>
<script type="text/javascript">
<!--
function Quadrat() {
    var Ergebnis = document.Formular.Eingabe.value * docu-
ment.Formular.Eingabe.value;
    alert("Das Quadrat von " + document.Formular.Eingabe.value + " =
" + Ergebnis);
}
//-->
</script>
</head>
<body>
<form name="Formular" action="">
<input type="text" name="Eingabe" size="3">
<input type="button" value="Quadrat errechnen" onClick="Quadrat()">
</form>
</body>
</html>
```

Wenn das Ergebnis statt in einem neuen Fenster in einem weiteren Formularfeld stehen soll, dann nehmen Sie bitte die folgenden Änderungen an der Datei **var_test_01.html** vor:

```
<html>
<head>
<title>Form-Test</title>

<script language="JavaScript" type="text/javascript">
function addieren(v1,v2)
    {
        var sum=0;
        var val, va2;
        va2=parseInt(v2); /* Werte als Integer */
        val=parseInt(v1);
        sum= val + va2;
        document.Formular.Ergebnisfeld.value=sum;
    }
</script>
</head>
<body>
<form name="Formular" action="">
Geben Sie die erste Zahl ein: <input type="text" name="val1" si-
ze="3">
Geben Sie die zweite Zahl ein: <input type="text" name="val2" si-
ze="3">
<input type="button" value="Addieren" on-
Click="addieren(val1.value,val2.value)">
<br><br>
Das Ergebnis ist:<input type="text" name="Ergebnisfeld">
</form></body></html>
```

Wenn Sie erreichen wollen, dass ein Ereignis direkt beim Laden ausgeführt wird, dann kann der Event-Handler **onLoad** direkt in das body-tag gepackt werden.

```
<body onLoad="alert('Willkommen auf meiner Homepage')">
```

Das komplette Beispiel-Skript:

gruss.html

```
<html>
<head>
<title>JavaScript-Test</title>
<script language="JavaScript">
<!--

//-->
</script>
</head>

<body onload="alert('Willkommen auf meiner Homepage')">
<h1>Die neue Website</h1>
</body>
</html>
```

Wenn Sie eine Umleitung zu einer anderen Seite erreichen wollen, dann reicht dazu der Javascript –Befehl:

```
location.href="ziel.htm";
```

Erzeugen Sie dazu die beiden Dokumente. Wenn Sie die Datei **umleitung.html** aktivieren, geht es direkt zum Dokument **ziel.html** weiter.

umleitung.html

```
<html>
<head>
  <title>Einfache Umleitung</title>
<script language="JavaScript" type="text/javascript">
<!--
location.href="ziel.htm";
//-->
</script>
</head>
<body bgcolor="white">
<h1>Umleitung 1</h1>
</body>
</html>
```

ziel.html

```
<html>
<head>
  <title>Zielseite</title>
</head>
<body>
<h1>Das ist die Zielseite</h1>
</body>
</html>
```

Wenn dies zeitverzögert laufen soll, dann können Sie es mit und ohne Javascript erreichen.

Die Javaskript-Lösung:

```
setTimeout("location.href='ziel.htm'",5000);
```

umleitung2.html

```
<html>
<head>
  <title>Einfache Umleitung</title>
<script language="JavaScript" type="text/javascript">
<!--
setTimeout("location.href='ziel.htm'",5000);
//-->
</script>
</head>
<body bgcolor="white">
<h1>Wir sind umgezogen!</h1>
<p>Falls Sie nicht in 5 Sekunden automatisch weitergeleitet werden, klicken
Sie bitte auf <a href="ziel.htm">Zielseite</a>.</p>
</body>
</html>
```

umleitung3.html

```
<html>
<head>
<meta http-equiv="refresh" content="5;URL=ziel.htm" />
  <title>Einfache Umleitung</title>
</head>
<body bgcolor="white">
<h1>Wir sind umgezogen!</h1>
<p>Falls Sie nicht in 5 Sekunden automatisch weitergeleitet werden, klicken
Sie bitte auf <a href="ziel.htm">Zielseite</a>.</p>
</body>
</html>
```

Den Befehl : **location.href="datei.html";**

können Sie auch brauchen, um Hyperlinks direkt auf Buttons zu legen. Erstellen Sie dazu die folgende Datei:

Linkbutton.html

```
<html>
<head>
  <title>Link-Buttons</title>
</head>

<body>

<h1>Hyperlink-Buttons mit JavaScript</h1>

<form>
<input type="button" value="Google " onClick="location.href='http://www.google.de'" />
<input type="button" value="Altavista" onClick="location.href='http://www.altavista.de'" />
</form>

</body>
</html>
```

Bei den ersten Funktionen (Skript S. 6/7) haben Sie jeweils eine neue neue Funktion erstellt, wenn Sie einen Button mit einer anderen Farbe definiert hatten. Für diesen Zweck eignet sich auch die Parameterübergabe:

Die Funktion wird so definiert:

Parameter

```
function hintergrund(farbe) {
document.bgColor=farbe
}
```

Dort, wo der Button steht, wird über den Event-Handler der parameter (z. B. rot) gesetzt:
`<input type="button" value="rot" onClick="hintergrund('red')" />`

parameter.html

```
<html>
<head>
  <title>Funktion mit Parameterübergabe</title>
<script language="JavaScript" type="text/javascript">
<!--
function hintergrund(farbe) {
document.bgColor=farbe
}
//-->
</script>
</head>
<body bgcolor="white">
<h1>Funktion mit Parameterübergabe</h1>
<form>
<input type="button" value="rot" onClick="hintergrund('red')" />
<input type="button" value="gelb" onClick="hintergrund('yellow')" />
<input type="button" value="hellgrau" onClick="hintergrund('silver')" />
</form>
</body>
</html>
```

Aufgaben:

1. Erstellen Sie eine Datei, die direkt beim Aufruf eine Dialogbox mit einem Wilkommens-Gruß zeigt. (Datei: willkommen.html)
2. Ergänzen Sie die Datei so, dass zuerst nach dem Vornamen gefragt wird, dann erfolgt der Gruß mit Nennung des Namens. (z. B. Willkommen, Liebe Elke ...) (Datei: willkommen2.html)
3. Erstellen Sie eine Datei mit einer Umleitung zu einer weiteren Datei. Verwenden Sie dazu eine Zeitverzögerung.
4. Erstellen Sie eine Datei, die den Benutzer nach einer Zahl fragt. Die Zahl soll zuerst in einen Satz (ohne Dialogbox, normaler Text) ausgegeben werden. (Du hast 8 eingegeben). Dann soll es der Klick auf einen Button ermöglichen, das Quadrat dieser Zahl zu berechnen.

Das folgende Beispiel³ ermöglicht die Berechnung eines Bruttobetrages.

Nettobetrag:	<input type="text" value="400"/>
Prozentsatz:	<input type="text" value="20"/>
Kommabetrag mit Punkt eingeben!	
<input type="button" value="Brutto ermitteln"/>	
Ergebnis:	<input type="text" value="480"/>

Erstellen Sie die Datei und speichern Sie mit dem Namen **brutto.html**

```
<html><head>
<title>Test</title>
<script type="text/javascript">
<!--
function BruttoBetrag(Netto, Prozente) {
  var Ergebnis = Netto * (1 + (Prozente / 100));
  return Ergebnis;
}

function SchreibeBrutto(Betrag, Prozentsatz) {
  var Wert = BruttoBetrag(Betrag, Prozentsatz);
  document.BruttoForm.Ergebnisfeld.value = Wert;
}
//-->
</script>
</head>
<body>
<form name="BruttoForm" action="">
<pre>
Nettobetrag: <input type="text" name="NettoEingabe">
Prozentsatz: <input type="text" name="ProzentEingabe"></pre>
<p>Kommabetrag mit Punkt eingeben!</p>
<input type="button" value="Brutto ermitteln"
onClick =
"SchreibeBrutto(document.BruttoForm.NettoEingabe.value,document.BruttoForm.ProzentEingabe.value)">
<pre>
Ergebnis:      <input type="text" name="Ergebnisfeld"></pre>
</form>
</body>
</html>
```

Aufgabe: Ändern Sie die Datei so ab, dass zu einem Nettobetrag nur die Mehrwertsteuer von 16% und der Mehrwertsteuer-Betrag ausgerechnet wird.

³ siehe Stefan Münz' SelfHTML

Aufgabe: Das folgende Beispiel dient dazu, noch einmal die Berechnung von Werten in einem kleinen Formular zu zeigen.⁴
 Dabei wird der Event-Handler **onChange()** verwendet. Dabei braucht man keinen Button zum Auslösen der Aktion zu erstellen, sondern es reicht, irgendwo hinzuklicken, um die Funktion auszulösen.
 Die Formatierung über das HTML-Tag `<pre> ... </pre>` erspart einem die Tabelle, weil damit eine nicht-proportionale Schrift wie Courier New erzielt wird, die einem die Ausrichtung erleichtert.

Ihre Rechnung

Bitte tragen Sie die gewünschte Anzahl ein. Klicken Sie dann irgendwo anders hin und alles wird berechnet!

Anzahl	Artikel	Einzelpreis	Preis
<input style="width: 50px;" type="text"/>	Pizza Primavera a´	5.30 Euro	= <input style="width: 50px;" type="text"/> Euro

Der Code zu der Datei ist: **rechnung_01.html**

```

<html>
  <head>
    <title>B. Rechnung mit Rechnung</title>
    <SCRIPT language="JavaScript">
      <!--
        //----- B. Rechnung -----
        function Rechnung()
        {
          var p1 = document.b.Anz1.value * 5.3;
          document.b.Preis1.value = p1;
        }
      //-->
    </SCRIPT>
  </head>

  <body>
    <h2>Ihre Rechnung</h2>
    Bitte tragen Sie die gewünschte Anzahl ein. Klicken Sie
    dann irgendwo anders hin und alles wird berechnet!<br>

    <form name="b">
      <pre><!-- erzeugt Monotype-Schrift, schreibt Mehrfach-Leerzeichen und
      Zeilenenden wie hier-->
      Anzahl Artikel          Einzelpreis  Preis
      -----
      <input type="Text" name="Anz1" size="5" onChange="Rechnung()"> Pizza
      Primavera a´ 5.30 Euro = <input type="Text" name="Preis1" size="10"> Euro
      -----
      </pre>
    </form>
  </body>
</html>
    
```

⁴ dazu wurde hier ein umfangreicheres Beispiel aus www.r-krell.de herangezogen und abgewandelt.

Zur Wiederholung noch einmal die Wertübergabe aus dem Formular:

```
<form name="b">
.....
.....
<input type="Text" name="Anz1" size="5" onChange="Rechnung()"> Piz-
za Primavera a´ 5.30 Euro = <input type="Text" name="Preis1" si-
ze="10"> Euro
```

Das Formular hat den Namen **b**
Die Formularfelder heißen **Anz1** und **Preis1**

In der Funktion wird der Name des Formulars und dann der Name des Feldes verwendet:

```
var p1 = document.b.Anz1.value * 5.3
```

allgemeine Form: **Document.Formularname.Formularfeld.value**

Trotz einfacher Multiplikation erscheinen bei bestimmten Werten falsche Berechnungen:

Anzahl	Artikel	Einzelpreis	Preis
<input type="text" value="3"/>	Pizza Primavera a´	5.30 Euro	= <input type="text" value="15.8999999999"/> Euro

Um dies zu verhindern, muss der Wert gerundet werden. Dazu dient eine Methode des sogenannten **Math**-Objekts.

```
var x=Math.round(Zahl)
```

Aufgabe: Um das Runden auszuprobieren, ändern wir zunächst das vorige Beispiel ab. Das Ergebnis soll sein:

Runden

Bitte tragen Sie einen Wert mit drei Kommastellen in das erste Feld ein. Verwenden Sie den Punkt statt Komma!

gerundet Euro

Sie müssen dazu wissen, dass die Methode **Math.round(Zahl)** nur ganze Zahlen liefert. Um auf zwei Kommastellen zu kommen, muss deshalb zunächst mal 100 gerechnet werden, um dann außerhalb des Rundens wieder durch 100 zu teilen.

```
p1=Math.round(p1*100)/100;
```

Weil bei dem vorliegenden Beispiel jetzt die Multiplikation (oder eine andere Rechnung) fehlt, muss die Variable, die aus dem Formularfeld stammt, erst wieder in eine Zahl umgewandelt werden, da sie ansonsten als string (Zeichenkette) vorliegt. Also muss zuerst die Umwandlung erfolgen:

```
p1=parseFloat(p1)
```

Aufgabe: Versuchen Sie, das Beispiel selber so abzuändern, dass nur das Ergebnis aus der Abbildung oben erscheint. Dateiname: **runden.html**

Der Code zum Vergleich ist auf der nächsten Seite zu sehen.

Anschließend soll dann die Rundung in die Datei **rechnung_01** eingebaut werden.

runden.html

```
<html>
  <head>
    <title>B. Rechnung mit Rechnung</title>
    <SCRIPT language="JavaScript">
      <!--
        //----- B. Rechnung -----
        function Runden()
        {
          var p1 = document.b.Anz1.value;
          p1=parseFloat(p1)
          p1=Math.round(p1*100)/100;
          document.b.Preis1.value = p1;
        }
      //-->
    </SCRIPT>
  </head>

  <body>
    <h2>Runden</h2>
    Bitte tragen Sie einen Wert mit drei Kommastellen in das erste Feld
    ein. Verwenden Sie den Punkt statt Komma!
    <form name="b">
      <pre>
-----
<input type="Text" name="Anz1" size="5" onChange="Runden()"> gerundet
<input type="Text" name="Preis1" size="10"> Euro
-----
      </pre>
    </form>
  </body>
</html>
```

Und hier noch die komplette Funktion mit der Rundung aus der Datei **rechnung_01.html**:

```
function Rechnung()
{
  var p1 = document.b.Anz1.value * 5.3;
  p1=Math.round(p1*100)/100;
  document.b.Preis1.value = p1;
}
```

Aufgabe: Jetzt soll das Formular so erweitert werden: (Datei **rechnung_04.html**)

Ristorante Napoli				
Bitte tragen Sie nur die Mengen ein. Klicken Sie dann irgendwo anders hin und alles wird berechnet!				
Anzahl	Artikel	Einzelpreis	Preis	
<input type="text" value="4"/>	Pizza Funghi	à 5.30 Euro	= <input type="text" value="21.2"/>	Euro
<input type="text" value="1"/>	Rigatoni	à 4.80 Euro	= <input type="text" value="4.8"/>	Euro
<input type="text" value="2"/>	Lasagne	à 6.50 Euro	= <input type="text" value="12.2"/>	Euro
Summe			= <input type="text" value="38.2"/>	Euro

Der komplette Code: **rechnung_04.html**

```

<html>
  <head>
    <title>B. Rechnung mit Rechnung</title>
    <SCRIPT language="JavaScript">
      <!--
        //----- B. Rechnung -----
        function Gesamt()
        {
          var p = 1*document.b.Preis1.value + 1*document.b.Preis2.value
            + 1*document.b.Preis3.value ;
          document.b.Rechnung.value = p;
        }

        function Rechnung1()
        {
          var p1 = document.b.Anz1.value * 5.3;
          document.b.Preis1.value = p1;
          Gesamt(); //Aufruf einer anderen Funktion
        }

        function Rechnung2()
        {
          var p2 = document.b.Anz2.value * 4.8;
          document.b.Preis2.value = p2;
          Gesamt();
        }

        function Rechnung3()
        {
          var p3 = document.b.Anz3.value * 6.1;
          document.b.Preis3.value = p3;
          Gesamt();
        }
      </SCRIPT>
    </head>

    <body>

```

bitte wenden!



```
<h2>Ristorante Napoli</h2>
Bitte tragen Sie nur die Mengen ein.
Klicken Sie
dann irgendwo anders hin und alles wird berechnet!<br>

<form name="b">
  <pre><!-- erzeugt Monotype-Schrift, schreibt Mehrfach-Leerzeichen und
Zeilenenden wie hier-->
Anzahl  Artikel                Einzelpreis  Preis
-----
<input type="Text" name="Anz1" size="5" onChange="Rechnung1()">  Pizza
Funghi   à 5.30 Euro = <input type="Text" name="Preis1" size="10"> Euro
<input type="Text" name="Anz2" size="5" onChange="Rechnung2()">  Rigatoni
à 4.80 Euro = <input type="Text" name="Preis2" size="10"> Euro
<input type="Text" name="Anz3" size="5" onChange="Rechnung3()">  Lasagne
á 6,50 Euro = <input type="Text" name="Preis3" size="10"> Euro
-----
                                Summe = <input type="Text" name="Rechnung"
size="10"onChange="Gesamt()"> Euro
  </pre>
</form>
</body>
</html>
```

Aufgabe: Im Code fehlt noch die Rundung, die auf den vorigen Seiten erklärt wurde. Ergänzen Sie das Beispiel noch um die erforderlichen Schritte.

Im folgenden Beispiel wird nur getestet, ob ein Besucher einer Site seinen Namen in ein Formularefeld eingetragen hat. Wenn ja, dann wird er zur Datei **start.html** weitergeleitet (Die Datei **start.html** können Sie aus dem Netzordner kopieren)

Anmeldung

Bitte geben Sie Ihren Namen ein:

Wenn nichts eingegeben wurde, dann erscheint die Meldung:



besuchercheck_01.html

```

<html>
<head>
<title>JavaScript-Testdatei</title>
<script language="JavaScript" type="text/javascript">
<!--
function besuchercheck(b)
    {
    if(b=="")
    alert("Sie haben Ihren Namen nicht eingegeben!");
    else location.href="start.html";
    }
//-->
</script>
</head>
<body>
<h2>Anmeldung</h2>
<form name="beispiel">
Bitte geben Sie Ihren Namen ein:
<input name="besucher" />
<input type="button" value="Klick mich"
onclick="besuchercheck(this.form.besucher.value)" />
</form>
</body>
</html>

```

Sie sehen fettgedruckt eine andere Form der Referenz auf das Formular. Die bisher verwendete Angabe war: (funktioniert genauso):

```

onclick="besuchercheck(document.beispiel.besucher.value)" />

```

Beispiel zu if ... else

In diesem Beispiel sollen noch einmal Zahlen nach der Eingabe in der Größe verglichen werden. Der Unterschied zu den einleitenden if .. else-Beispielen im Skript ist der, dass hier die Werte in Formularfeldern und nicht über prompt-Dialogboxen erfragt werden.

Zahl 1 eingeben: Zahl 2 eingeben:

Datei if_form_01.html

```
<html>
<head>
<title>JavaScript-Test</title>
<script language="JavaScript">
<!--
var ausgab,zahl1,zahl2
function pruefen(zahl1,zahl2)
{
var zahl1=document.wenn_dann.zahl1.value;
var zahl2=document.wenn_dann.zahl2.value;
if(zahl1<zahl2)
{
ausgab="Zahl 1 ist kleiner";
}
else if (zahl1>zahl2)
{
ausgab="Zahl 1 ist größer";
}
else
{
ausgab="beide Zahlen sind gleich";
}
ausgabe();
}
function ausgabe()
{
document.write(ausgab)
}
//-->
</script>
</head>
<body>
<form name="wenn_dann">
Zahl 1 eingeben: <input type="Text" name="zahl1" size="5">
Zahl 2 eingeben: <input type="Text" name="zahl2" size="5"><br><br>
<input type="button" value="Prüfen" onClick="pruefen(zahl1,zahl2)">
<input type="reset">
</form></body>
</html>
```

Aufgabe: Überlegen Sie, was Sie ändern müssen, wenn auch die Ergebnisse in ein Formularfeld gebracht werden sollen. So soll das Ergebnis z. B. werden:

Zahl 1 eingeben: Zahl 2 eingeben:
 Das Ergebnis ist:

Noch ein Beispiel zu if else:

Guten Tag!

Nach der Eingabe von Geschlecht und Vornamen bitte den Knopf drücken:

Geschlecht: männlich weiblich

Name:

Das Beispiel "Guten Tag" soll die Formularauswertung mit if ..else an einem Beispiel zeigen, in dem auch die Radio-Buttons mit einbezogen werden.

```

<html>
<head>
<title> </title>
<SCRIPT language="JavaScript">
<!--
//----- F. Gruss -----
var sex = "w"; //globale Variable speichert das Geschlecht

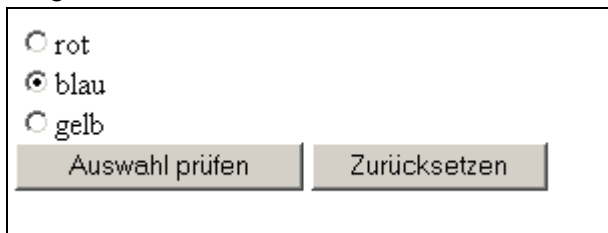
function Gruss()
{
var vn = document.f.VName.value;
var aus = "Guten Tag, ";
if (sex=="w")
{
aus = aus + "liebe"
}
else
{
aus = aus + "lieber"
}
aus = aus + " " + vn + "!" // setzt Text zusammen
document.f.Ausgabe.value = aus;
}
//-->
</SCRIPT>
</head>
<body>
<h2>Guten Tag!</h2>
Nach der Eingabe von Geschlecht und Vornamen bitte den Knopf drücken:<br>

<form name="f">
<P ALIGN="CENTER">
Geschlecht:
<input type="radio" name="Geschlecht" onclick="sex='m'"> männlich
<input type="radio" name="Geschlecht" onclick="sex='w'" checked>
weiblich<br>
Name:<input type="Text" name="VName" size="25"><br><br>
<input type="button" value="Eingabe komplett?" onClick="Gruss()"><br><br>
<input type="Text" name="Ausgabe" size="40">
</P>
</form>
</body></html>

```

Beispiel zu if ..else:⁵

Im folgenden Beispiel sehen Sie ein kleines Formular mit "Radiobuttons". Es soll zunächst nur der Demonstration dienen, wie die Werte aus dem Formular ausgelesen und an JavaScript übergeben werden.



Gedacht ist das Beispiel so, dass eine der Farben durch Anklicken ausgewählt wird. Drücken Sie dann auf den Button "Auswahl prüfen". Es erscheint ein Fenster, in dem die gewählte Farbe angezeigt wird. Durch Anklicken von "Zurücksetzen" wird die Auswahl wieder gelöscht.

Code:

Datei: **if_radio.html**

```
<html><head>
<title>JavaScript-Test</title>
<script language="JavaScript">

</script>
</head>
<body>

<form>
<input type="radio" name="Auswahl">rot<br>
<input type="radio" name="Auswahl">blau<br>
<input type="radio" name="Auswahl">gelb<br>
<input type="button" value="Auswahl prüfen"

onClick="if(Auswahl[0].checked)
        {alert('rot')}
        else if(Auswahl[1].checked)
        {alert('blau')}
        else if(Auswahl[2].checked)
        {alert('gelb')}
        else
        {alert('Bitte eine Farbe auswählen')}">

<input type="reset">
</form>
</body>
</html>
```

Auswahl[0].checked

Hier wird abgefragt, ob der entsprechende Radiobutton angeklickt wurde. Der erste Button hat die Nummer 0, der nächste die Nummer 1 usw.

Sie sehen hier, dass die if-Abfrage in den **onClick**-Eventhandler eingebaut ist. Die Anführungszeichen beginnen direkt nach **onclick** und werden erst hinter der geschweiften Klammer nach dem letzten **alert** geschlossen. Danach steht noch das **>**, mit dem erst der input-tag zuende ist.

⁵ Aus: http://www.kipanya.com/JavaScript/js_2-3.html

Switch ...case:

Das Beispiel von der vorigen Seite, das bei der **if ... else if ... else** - Bedingung verwendet wurde, lässt sich auch mit der Fallunterscheidung durchführen.

```
<html>
<head>
<title>Test</title>

<form>
<skript language="JavaScript">
<!--
var Wert='0';
// --></skript>
<input type="radio" name="Auswahl"
onClick="Wert='1' ">rot<br>
<input type="radio" name="Auswahl" onClick="Wert='2' ">blau<br>
<input type="radio" name="Auswahl" onClick="Wert='3' ">gelb<br>
<input type="button" VALUE=" Auswahl prüfen "
onClick="switch(parseInt(Wert))
{
case 1:
    alert('rot');
    break;
case 2:
    alert('blau');
    break;
case 3:
    alert('gelb');
    break;
default:
    alert('Bitte eine Farbe auswählen')
}">
<input type="reset" value="Löschen" onClick=" Wert='0' ">
</form>
```


Sie sehen, dass die Auswahl hier über die Variable Wert vorgenommen wird. Die switch-case-Unterscheidung braucht Werte, um **case 1**, **case 2**, **case 3** usw. unterscheiden zu können.

Das Ergebnis ist dasselbe wie im Beispiel vorher.

Das Beispiel mit der if-Abfrage und den Radio-Buttons soll jetzt verwendet werden, um je nach Wunsch verschiedene Grafiken anzuzeigen.

Dabei handelt es sich um **bank.jpg**, **biergarten.jpg** und **boot.jpg**

Bank
 Biergarten
 Boot



Die gewünschte Wirkungsweise ist klar: Wenn zuerst der Radiobutton bei z. B. **Boot** angeklickt wird und dann auf die Schaltfläche Bild anzeigen, dann soll die Datei **boot.jpg** angezeigt werden.

Datei **bild_wechsel_01.html**

Die Grafiken befinden sich im Unterordner **image**

```

<html><head>
<title>JavaScript-Test</title>
<script language="JavaScript">
</script>
</head>
<body>
<form>
<input type="radio" name="Auswahl">Bank<br>
<input type="radio" name="Auswahl">Biergarten<br>
<input type="radio" name="Auswahl">Boot<br>
<input type="button" value="Bild anzeigen"
onClick="if(Auswahl[0].checked)
    {
        document.images.bild.src='image/bank.jpg';
    }
    else if(Auswahl[1].checked)
    {
        document.images.bild.src='image/biergarten.jpg';
    }
    else if(Auswahl[2].checked)
    {
        document.images.bild.src='image/boot.jpg';
    }
    else
    {
        alert('Bitte etwas auswählen')
    }">
<input type="reset">
</form>

</body>
</html>

```

Erläuterungen zur Datei:

Ganz unten erst findet sich der HTML-Befehl, mit dem das erste Bild zur Anzeige gebracht wird.

```

```

Darauf kann dann verwiesen werden. Sehen Sie, wie derselbe Name **bild** immer wieder gebraucht wird, um dann jeweils eine der drei jpg-Dateien damit zu verlinken.

```
onClick="if(Auswahl[0].checked)
        {
            document.images.bild.src='image/bank.jpg';
        }
        else if(Auswahl[1].checked)
        {
            document.images.bild.src='image/biergarten.jpg';
        }
        else if(Auswahl[2].checked)
        {
            document.images.bild.src='image/boot.jpg';
        }
    "
```

Die folgenden vier HTML-Dokumente werden zunächst gebraucht, um verschiedene Arten und Weisen der Navigation auszuprobieren:

- ⇒ über Listenfelder mit Go-Button
- ⇒ über Listenfelder mit Direktauswahl
- ⇒ über Frames

Aufgabe: Erstellen Sie diese vier Seiten (am besten in einem separaten Ordner; in dem auch die folgenden Dateien zur Navigation untergebracht werden:

seite_a.html	seite_b.html	seite_c.html	seite_d.html
<pre><html> <head> <title>Seite A</title> </head> <body> <h1>Seite A</h1> </body> </html></pre>	<pre><html> <head> <title>Seite B</title> </head> <body> <h1>Seite B</h1> </body> </html></pre>	<pre><html> <head> <title>Seite C</title> </head> <body> <h1>Seite C</h1> </body> </html></pre>	<pre><html> <head> <title>Seite D</title> </head> <body> <h1>Seite D</h1> </body> </html></pre>

Navigation über einen Go-Button (Auswahl und Aufruf der neuen Seite in einem Dokumentfenster)

Datei **Go.html**

```
<html><head>
<title>Go!</title>
<script language="JavaScript" type="text/javascript">
<!--
function wechsel()
{
    var auswahl, seite;
    auswahl=document.listel.wahl.selectedIndex;
    seite=document.listel.wahl.options[auswahl].value;
    if (seite=="nichts")
    {
        document.listel.reset();
        return false;
    }
    else
    {
        location.href=seite;
    }
}
//-->
</script>
</head>
<body bgcolor="white">
<h1>Pull-Down-Menü mit Go-Button</h1>
<form name="listel">
<select size="1" name="wahl">
<option value="nichts" selected="selected">Bitte wählen!</option>
<option value="seite_a.html">Seite A</option>
<option value="seite_b.html">Seite B</option >
<option value="seite_c.html">Seite C</option >
<option value="nichts">-----ich bin ein Trennstrich-----</option >
<option value="seite_d.html">Seite D</option >
</select>
<input type="button" value="Go!" onClick="wechsel()" />
</form></body></html>
```

Das folgende Beispiel ist vom Code her sehr ähnlich (am besten nur Änderungen vornehmen). Im Unterschied zur Variante vorher brauchen Sie hier keinen Go-Button, weil die Auswahl über **onChange()** vorgenommen wird:

Change.html

```
<html>
<head>
<title>onChange</title>
<script language="JavaScript" type="text/javascript">
<!--
function wechsel() {
    var auswahl, seite;
    auswahl=document.listel.wahl.selectedIndex;
    seite=document.listel.wahl.options[auswahl].value;
    if (seite=="nichts")
        {
        document.listel.reset();
        return false;
        }
    else
        {
        location.href=seite;
        }
    }
//-->
</script>
</head>
<body bgcolor="white">
<h1>Pull-Down-Menü mit onChange</h1>
<form name="listel">
<select size="1" name="wahl" onChange="wechsel()">
<option value="nichts" selected="selected">Bitte wählen!</option>
<option value="seite_a.html">Seite A</option>
<option value="seite_b.html">Seite B</option >
<option value="seite_c.html">Seite C</option >
<option value="nichts">-----ich bin ein Trennstrich-----</option >
<option value="seite_d.html">Seite D</option >
</select>
</form></body></html>
```

Frame mit Pull-Down-Feld zum Navigieren

Hier soll ein zweigeteiltes Frameset aufgebaut werden. Im oberen Teil befindet sich das Auswahlfeld, in den unteren Frame wird zuerst die Seite A (seite_a.html) geladen, dann je nach Auswahl die anderen Seiten.

Seite A

Aufgabe: Erstellen Sie zu dem Zweck zuerst das Frameset :

Datei **pull_frame_01.html**

```
<html>
<head>
    <title>Das ist das Frameset</title>
</head>
<frameset rows="50,*">
<frame src="navi.html" name="navi" >
<frame src="seite_a.html" name="main" >
</frameset>
</html>
```

Besonders wichtig ist hier natürlich die Datei **navi.html**, über die die Steuerung erfolgt:

```
<html> <head>
<title>onChange</title>
<script language="JavaScript" type="text/javascript">
<!--
function wechsel()
    {
    var auswahl, seite;
    auswahl=document.listel.wahl.selectedIndex;
    seite=document.listel.wahl.options[auswahl].value;
    if (seite=="nichts")
        {
        document.listel.reset();
        return false;
        }
    else
        {
        parent.main.location.href=seite;
        }
    }
//-->
</script>
</head>
<body>
<form name="listel">
<select size="1" name="wahl" onChange="wechsel()">
<option value="nichts" selected="selected">Bitte wählen!</option>
<option value="seite_a.html">Seite A</option>
<option value="seite_b.html">Seite B</option >
<option value="seite_c.html">Seite C</option >
<option value="nichts">-----ich bin ein Trennstrich-----</option >
<option value="seite_d.html">Seite D</option >
</select>
</form>
</body> </html>
```

Wichtig ist hier diese Anweisung:

parent.main.location.href=seite;

Arrays

Im Unterschied zu den bisher verwendeten Variablen sind Arrays mehrdimensional. Das bedeutet, dass man in einem Array mehr als einen Wert speichern kann.

Aufgabe: Erstellen Sie die folgende Datei. Geben Sie ihr den Namen **array_01.html**

```
<html>
<head>
<title>Array-Testdatei</title>
</head>
<body>
<h2>Wochentag</h2>
<script language="JavaScript" type="text/javascript">
<!--
var wochentage = new Array("Montag", "Dienstag", "Mittwoch", "Donnerstag",
"Freitag", "Samstag", "Sonntag");
document.write(wochentage[0])
//-->
</script>
</body>
</html>
```

Die Zeile belegt die Werte im Array:

```
var wochentage = new Array("Montag", "Dienstag", "Mittwoch",
"Donnerstag", "Freitag", "Samstag", "Sonntag");
```

Der Array enthält die Werte jetzt in nummerierter Reihenfolge. Der erste Wert ist 0. Deshalb wird über die Zeile

```
document.write(wochentage[0])
```

Der Tag **Montag** (der den ersten Wert im Array belegt) ausgegeben.

Das Initialisieren des Arrays erfolgt über das Schlüsselwort **new Array**, gefolgt von den Werten, die der Array enthalten soll. Im Beispiel oben haben Sie die Kurzform kennengelernt. Die ausführlichere Schreibweise sehen Sie im nächsten Beispiel:

Datei **array_02.html**

```
<html><head>
<title>Array-Testdatei</title>
</head>
<body>
<h2>Monate</h2>
<script language="JavaScript" type="text/javascript">
<!--
var monate = new Array(7)
monate[0]="vorherleer00";
monate[1]="Januar";
monate[2]="Februar";
monate[3]="März";
monate[4]="April";
monate[5]="Mai";
monate[6]="Juni";
document.write(monate[5])
//-->
</script>
</body>
</html>
```

Hier wird zuerst gesagt, wieviel Werte der Array bekommt.

```
var monate = new Array(7)
```

Dann folgt in den einzelnen Zeilen die Zuweisung. Um mit Januar=1 zu beginnen, habe ich hier **monate[0]="vorherleer00";** belegt.

Die Eigenschaft `length` des array-Objekts

Die folgende Eigenschaft des arrays, die Länge (`length`) wird gebraucht, um die Anzahl gespeicherter Elemente aus dem Array zu lesen.

Aufgabe: Speichern Sie die vorhin erstellte Datei `array_01.html` mit dem neuen Namen `array_01a.html`. Ändern Sie dort die `document.write`- Zeile so ab:

```
document.write("es sind " +wochentage.length + " Tage gespeichert")
```

Das Ergebnis ist:

<p style="text-align: center;">Wochentag</p> <p style="text-align: center;">es sind 7 Tage gespeichert</p>

die `length`-Eigenschaft des Array lässt sich gut zur Steuerung von Schleifen einsetzen.

Aufgabe: Speichern Sie die Datei noch einmal mit dem zusätzlichen Namen `array_01_schleife.html` und ändern Sie den Code folgendermaßen ab:

```
<html>
<head>
<title>Array-Testdatei</title>
</head>
<body>
<h2>Wochentag</h2>
<script language="JavaScript" type="text/javascript">
<!--
var wochentage = new Array("Montag", "Dienstag", "Mittwoch", "Donnerstag",
"Freitag", "Samstag", "Sonntag");

document.write("es sind " +wochentage.length + " Tage gespeichert"+"<br>")
for (var i = 0; i < wochentage.length; i++)
{
    document.write(wochentage[i]+"<br>");
}
//-->
</script>
</body>
</html>
```

Erklärung der `for`-Schleife: Der Zähler `i` wird auf 0 gesetzt. Solange `i` kleiner ist als die Anzahl gespeicherter Werte im Array "Wochentage", wird die Schleife durchlaufen. Bei jedem Schleifendurchlauf wird `i` um 1 erhöht. Bei jedem Schleifendurchlauf trifft Javascript auf die Anweisung:

```
document.write(wochentage[i]+"<br>");
```

Das heißt, dass in die eckigen Klammern der jeweilige Wert von `i` eingesetzt wird.

for ... in-Schleifen

Eine Besonderheit der `for`-Schleife ist die `for in` Schleife. Sehen Sie dazu folgendes Beispiel:

```
<html>
<head>
<title>For..in-Testdatei</title>
</head>
<body>
<h2>Wochentage</h2>
<script language="JavaScript" type="text/javascript">
<!--
```

```
var wochentage = new Array("Montag", "Dienstag", "Mittwoch", "Donnerstag",  
"Freitag", "Samstag", "Sonntag");  
for (i in wochentage) {  
  document.write(wochentage[i]+"<br>");  
}  
//-->  
</script>  
</body>  
</html>
```

führt zu:

Wochentage

Montag
Dienstag
Mittwoch
Donnerstag
Freitag
Samstag
Sonntag

Artikelsucher

Unsere neue Kollektion umfasst Jacken, Hosen, Hemden und Krawatten.

Wählen Sie den Artikel, dessen Artikelnummer Sie anzeigen wollen:

- Jacke
 Hose
 Hemd
 Krawatte

A1002

Ihre Bestellung

Artikelnummer: Menge:

```

<html>
<head>
<title>JavaScript-Testdatei</title>
<script language="JavaScript1.1" type="text/javascript">
<!--
function anr_anzeigen(nr) {
var anr = new Array("A1001","A1002","A1003", "A1004");

for (i=0; i<document.beispiel.artikel.length;i++) {
  if(document.beispiel.artikel[i].checked)
    document.beispiel.artikelnr.value=anr[i];
  }
}
//-->
</script>
</head>

<body>
<form name="beispiel">
<h2>Artikelsucher</h2>
<p>Unsere neue Kollektion umfasst Jacken, Hosen, Hemden und Krawatten.</p>
Wählen Sie den Artikel, dessen Artikelnummer Sie anzeigen wollen:<br>
<input type="radio" name="artikel" onclick="anr_anzeigen()" />Jacke<br />
<input type="radio" name="artikel" onclick="anr_anzeigen()" />Hose<br />
<input type="radio" name="artikel" onclick="anr_anzeigen()" />Hemd<br />
<input type="radio" name="artikel" onclick="anr_anzeigen()" />Krawatte</p>
<input readonly name="artikelnr" value="0" /></p>
<p>Ihre Bestellung</p>
Artikelnummer: <input name="bestellung" /> Menge: <input name="bestellung"
/>
<input type="submit" value="Bestellen" />
</form>
</body>
</html>

```

Am folgenden Beispiel sehen Sie, wie ein Array neu definiert wird, mit vier Texten gefüllt wird, um dann einen wechselnden Werbetext in die Statuszeile zu bringen:

```
<html><head>
  <title>Wechselnder Text</title>
<script language="JavaScript" type="text/javascript">
<!--
var i, botschaft;
i=0;
botschaft=new Array();
botschaft[0]="Neu: Buch zu Windows 2000";
botschaft[1]="Nachauflage: Die besten Office 2000 Geheimnisse";
botschaft[2]="Sonderangebot: PC-Seminar Word 2000";
botschaft[3]="Bestseller: SchnellAnleitung Outlook 2000";
function wechseltext() {
window.status=botschaft[i];
if (i<botschaft.length-1) {
i++;
}
else {
i=0;
}
setTimeout("wechseltext()",5000);
}
//-->
</script>
</head>
<body bgcolor="white" onLoad="wechseltext()">
<h1>Wechseltext in der Statuszeile!</h1>
</body>
</html>
```

Bilder per Schleife und Array vorladen

Im folgenden Beispiel finden Sie eine verbesserte Version der Diashow von Seite 26. Wenn es sich um eine große Menge Bilder handelt, wäre es zu mühsam, jedes einzelne Bild im Array aufzuführen. Die Grafiken heißen hier alle bildnn.jpg, wobei nn für eine fortlaufende Zahl steht.

Zu beachten ist dabei, dass die Bilder alle die gleiche Größe haben müssen, denn sie werden in den Platzhalter geladen, der im HTML-Teil mit Größenangaben versehen wird:

```

```

```
<html>
<title>Bildwechsel</title>
<script language="JavaScript" type="text/javascript">
<!--
var nr=0;
var bilder=new Array()
for (i=0;i<13;i++) { // Vorladen per Schleife
bilder[i]=new Image();
bilder[i].src="bild"+i+".jpg"; // Deklaration
}
function wechsel() {
nr++;
if (document.images) {
if (nr==12) { // 12 ist Anzahl der Bilder
nr=0;
}
document.bild.src=bilder[nr].src;
setTimeout("wechsel()",5000); // alle 5 Sek.
}
}
//-->
</script>
</head>
<body bgcolor="white" onLoad="wechsel()">
<h1>Bildwechsel</h1>



</body>
</html>
```

Aufgabe: Eine Schwachstelle ist noch, dass die Anzahl (12) hier nicht automatisch ermittelt wird. Suchen Sie dazu eine Lösung über die **length**-Eigenschaft des Arrays!

Im folgenden Beispiel für eine for-Schleife wird berechnet, wie sich ein Kapital vermehrt, wenn Zins und Zinseszins Jahr für Jahr zusammen addiert werden.

Kapital eingeben:	<input type="text" value="1000"/>	Nach 1 Jahren 1030 Euro
		Nach 2 Jahren 1060.9 Euro
Prozentsatz eingeben:	<input type="text" value="3"/>	Nach 3 Jahren 1092.73 Euro
		Nach 4 Jahren 1125.51 Euro
Anzahl Jahre eingeben:	<input type="text" value="5"/>	Nach 5 Jahren 1159.28 Euro

zins.html

```
<html><head>
<title>Zinsen</title>
<script language="JavaScript">
<!--
  var k,zins,jahre
  function rechnen(k,p,jahre)
    {
      var k=document.zinsen.k.value;
      var p=document.zinsen.p.value;
      var jahre=document.zinsen.jahre.value;
      for(i=1;i<=jahre;i++)
        {
          zins=(k*p/100);
          k=parseFloat(k)
          zins=parseFloat(zins);
          k=Math.round((k+zins)*100)/100
          document.write("Nach " + i + " Jahren "+k+" Euro <br>");
        }
    }
  //-->
</script>
</head><body>
<pre>
<form name="zinsen">
Kapital eingeben:      <input type="Text" name="k" size="5"><br>
Prozentsatz eingeben: <input type="Text" name="p" size="5"><br>
Anzahl Jahre eingeben: <input type="Text" name="jahre" size="5"><br>
<input type="button" value="Rechnen"
onClick="rechnen(k.value,p.value,jahre.value)"><br>
<input type="reset">
</form>
</pre></body></html></html>
```

Übungsaufgaben zur Testvorbereitung:

1. Erstellen Sie eine Datei mit einem Array, in dem 10 beliebige Vornamen gespeichert sind.
Über eine Schleife sollen alle Vornamen ausgegeben werden.
2. Dann soll eine Benutzerfrage nach einer Zahl von 1 bis 10 fragen, und der entsprechende Vorname soll ausgegeben werden.
- 3) Schreiben Sie ein Programm mit einer Schleife, die das kleine Einmaleins der Zahl 5 ausgibt
einmalfünf.html
0 mal 5 ist 0
1 mal 5 ist 5
2 mal 5 ist 10
3 mal 5 ist 15
4 mal 5 ist 20
5 mal 5 ist 25
6 mal 5 ist 30
7 mal 5 ist 35
8 mal 5 ist 40
9 mal 5 ist 45
10 mal 5 ist 50
4. Erweitern Sie das Programm so, dass das kleine Einmaleins der Zahlen von 1 bis 5 ausgegeben wird.
5. Erstellen Sie eine Datei mit einem Button, der beim Anklicken zur TAW-Site weiterleitet.
6. Erstellen Sie eine HTML-Datei mit einem Formular, das nach Namen und Vornamen fragt. Dann soll es einen OK-Button geben.

Wenn kein Vorname oder kein Name eingegeben wurde, soll dies als Meldung auf dem Bildschirm gemeldet werden. (alert-box)
7. Erstellen Sie eine Datei mit einer case-Fallunterscheidung. Der Benutzer soll zuerst gefragt werden, welcher Tarifgruppe (Gehaltsklasse) er angehört. Dann soll das Bruttogehalt angezeigt werden.
8. Fragen Sie den Benutzer, wieviele Kinder er hat. Dann soll das Kindergeld ausgegeben werden: Für das erste Kind 135 Euro, für das zweite 150 Euro, und ab dem dritten 180 Euro.

vornamen1_schleife.html

```
<html><head>
<title>JavaScript-Testdatei</title>
</head><body>
<h2>Vornamen</h2>
<script language="JavaScript" type="text/javascript">
<!--
var vornamen = new Array("Ute", "Maria", "Cornelia", "Sarah", "Gerlinde",
"Paul", "Freddie", "Hans", "Peter", "Michael");
for (i in vornamen)
{
    document.write(vornamen[i]+"<br>");
}
//-->
</script></body></html>
```

vornamen1_nr.html

```
<html><head>
<title>JavaScript-Testdatei</title>
</head><body>
<h2>Vornamen</h2>
<script language="JavaScript" type="text/javascript">
<!--
var vn
var vornamen = new Array("Ute", "Maria", "Cornelia", "Sarah", "Gerlinde",
"Paul", "Freddie", "Hans", "Peter", "Michael");
vn=prompt("Bitte die Nummer eingeben",0);
document.write(vornamen[vn]);
//-->
</script></body></html>
```

einmalfünf.html

```
<html><head>
<title>Schleife</title>
<script language="JavaScript" type="text/javascript">
<!--
var i, z,x;
for(i=0;i<=10;i++)
{
    x=i*5
    document.write(i+" mal 5 ist "+x +"<br>");
}
//-->
</script></head><body></body></html>
```

einmaleins.html

```
<html><head>
<title>Schleife</title>
<script language="JavaScript" type="text/javascript">
<!--
var i, e,x;
for(k=1;k<=5;k++)
{
    for(i=1;i<=10;i++)
    {
        e=i*k
        document.write(i+" mal "+k+" ist "+e +"<br>");
    }
}
//-->
</script></head><body></body></html>
```

Das folgende Beispiel von J. Krell⁶ arbeitet mit ein paar kleinen Tricks, um zu erreichen, dass nur die angeklickten Werte addiert werden.

Was wollen Sie essen?

Stellen Sie sich durch Anklicken Ihr Menü zusammen. Klicken Sie zum Schluss in das Betragsfeld und halten Sie bitte den angezeigten Betrag für die Kasse bereit!

<input type="checkbox"/>	Suppe	(1.50 Euro)
<input checked="" type="checkbox"/>	kleiner gem. Salat	(2.10 Euro)
<input checked="" type="checkbox"/>	Hauptgericht	(3.50 Euro)
<input type="checkbox"/>	Nachtisch	(1.80 Euro)

Ihr Menü kostet Euro

kantine.html

```

<html>
  <head>
    <title>Kantinen-Menü</title>
    <SCRIPT language="JavaScript">
      <!--
        //Abfrage von Checkbox-Value funktioniert nicht. Deshalb globale
        //Variablen. Die Werte bedeuten: -1 = ohne Haken; +1 = mit Haken
        var d_1 = -1; //Suppe (am Anfang ohne Haken)
        var d_2 = -1; //Salat
        var d_3 = -1; //Hauptgericht
        var d_4 = -1; //Nachtisch

        function Rechnung()
        {
          var summe = 0;
          if (d_1 > 0) //falls angehakt, Preis zur Summe dazu
            { summe = summe + 1* 1.50; } //1*, damit nicht als Text
hintendran
          if (d_2 > 0)
            { summe = summe + 1* 2.10; }
          if (d_3 > 0)
            { summe = summe + 1* 3.50; }
          if (d_4 > 0)
            { summe = summe + 1* 1.80; }
          document.d.Summenfeld.value = summe;
        }

        //-->
      </SCRIPT>
    </head>

    <body>
      <h2>Was wollen Sie essen?</h2>
      Stellen Sie sich durch Anklicken Ihr Menü zusammen. Klicken Sie zum Schluss
      in das Betragsfeld und halten Sie bitte den angezeigten Betrag für die
      Kasse bereit!

      <form name="d">
      <pre>
      <!-- Hinweis: Multiplikation mit (-1) schaltet zwischen -1 und +1 um! -->

```

⁶ www.r-krell.de

```
<input type="Checkbox" onClick="d_1 = (-1)*d_1"> Suppe (1.50
Euro)
<input type="Checkbox" onClick="d_2 = (-1)*d_2"> kleiner gem. Salat (2.10
Euro)
<input type="Checkbox" onClick="d_3 = (-1)*d_3"> Hauptgericht (3.50
Euro)
<input type="Checkbox" onClick="d_4 = (-1)*d_4"> Nachtisch (1.80
Euro)
-----
Ihr Menü kostet <input type="Text" name="Summenfeld" size="8"
onClick="Rechnung()" onChange="Rechnung()"> Euro
</pre>
</form>

</body>
</html>
```

DHTML

Die Abkürzung DHTML steht für "dynamic HTML". ES geht dabei um die Verbindung von HTML, CSS-Anweisungen und Javascript, mit dem erwünschten Effekt, dass man die Eventhandler von Javascript einsetzt, um die Website dynamisch zu machen, z. B. indem man Menüs definiert, die man über Mausberührung steuern kann.

Das Beispiel aus dem Skript zu Javascript ab S. 43 soll hier etwas übersichtlicher dargestellt werden. Zunächst ein Ausschnitt, aus dem nur das Wichtigste klar werden soll:

Hier ist ein geschlossener Menüpunkt zu sehen.



Wenn die Maus darüberfährt, öffnet er sich, und die Links aus dem Menü können bedient werden.



[Merkmale einer Datenbank](#)

[Einführung II](#)

[Arbeiten mit Tabellen](#)

Zunächst einmal die reinen CSS-Befehle in einer separaten Datei:

format1.css

```
/* Zuerst wird die Leiste angelegt. */
div.leiste {
    position: absolute;
    top: 50px;
    left: 20px;
    height: 20px;
    width: 240px;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt; background-color: #99ccff;
    layer-background-color: #99ccff;
}
/* Der nächste Stil gestaltet das Menü */

div.menu1 {
    position: absolute;
    top: 70px;
    left: 30px;
    height: 82px;
    width: 170px;
    visibility: hidden;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt;
    background-color: #ffff99;
    layer-background-color: #ffff99;}
```

Die HTML-Datei mit Javascript sieht so aus (zunächst nur für den Internet Explorer, funktioniert nicht in Netscape)

menu_01.html

```
<html><head>
<title>Datenbank</title>
<link rel="stylesheet" href="format1.css" type="text/css">
<script language="JavaScript" type="text/javascript">
<!--
function auf(menu) {
    if (document.all) {
        document.all[menu].style.visibility="visible";
    }
}
function zu(menu) {
```


Hier folgt jetzt die verbesserte Version, in der a) zwei Menüpunkte nebeneinander gestellt werden und b) die Links mit css-Angaben ansprechender formatiert werden.

format.css

```
/* Zuerst wird die Leiste angelegt. */
div.leiste {
    position: absolute;
    top: 50px;
    left: 20px;
    height: 20px;
    width: 600px;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt; background-color: #99ccff;
    layer-background-color: #99ccff;
}
/* Die nächsten beiden Stile gestalten die Menüs */

div.menu1 {
    position: absolute;
    top: 70px;
    left: 30px;
    height: 82px;
    width: 170px;
    visibility: hidden;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt;
    background-color: #ffff99;
    layer-background-color: #ffff99;}

div.menu2 {
    position: absolute;
    top: 70px;
    left: 225px;
    height: 82px;
    width: 200px;
    visibility: hidden;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt;
    background-color: #ffff99;
    layer-background-color: #ffff99;
}

/* Diese 4 Hover-Stile steuern das Aussehen der Links */

A:link {
    text-decoration: none;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt; color: black;
}

A:hover {
    text-decoration: none;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt; color: red;
}

A:active {
    text-decoration: none;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt;
    color: black;
}
```

```

A:visited {
    text-decoration: none;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 10pt;
}

p {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 11pt;
    color: #004284;
    background-color: #FFFBEA;
}

```

Von der Datei, die die Menusteuerung enthält, hier zunächst nur von <head> bis </head>, um zu zeigen, wie die Javascript-Anweisungen aussehen. Der Teil, der die Besonderheiten von Netscape behandelt, ist hier weggelassen.

Die Anweisung

```
<link rel="stylesheet" href="format.css" type="text/css">
```

ruft die externe Datei format.css auf.

menu.html

```

<html>
<head>
<title>Datenbank</title>
<link rel="stylesheet" href="format.css" type="text/css">

<script language="JavaScript" type="text/javascript">
<!--
function auf(menu) {
    if (document.all) {
        document.all[menu].style.visibility="visible";
    }
    if (document.layers) {
        document.layers[menu].visibility="visible";
    }
}

function zu(menu) {
    if (document.all) {
        document.all[menu].style.visibility="hidden";
    }
    if (document.layers) {
        document.layers[menu].visibility="hidden";
    }
}
//-->
</script></head>

```

Nun noch der body-Teil der Datei **menu.html**

Hier sind jetzt hauptsächlich die Links enthalten, und es wird gesteuert, welcher Menüteil auf- und zu geht.

Die Dateien, die hier verlinkt sind, können Sie kopieren,. Sie sind in den Ordnern **Access** und **Mysql** zu finden. Achten Sie auf die Pfadangaben: Sie müssen dafür sorgen, dass Ihre Menu-Datei eine Ordnerstufe höher ist.

Hier ein Beispiel aus dem Javascript –Workshop, Verlag Addison Wesley:
Hier wird ein Schreibmaschinen-Effekt erzeugt, der Text erscheint Buchstabe für Buchstabe:

schreib.html

```
<html>
<head>
<title>JavaScript-Workshop Beispiel 09/02</title>
<script>
var text = "Immerhin besitzt doch ein Mensch die Freiheit, "
  + "seine Hand auf den Kopf zu legen. (G.W. Leibniz, "
  + "Neue Abhandlung über den menschlichen Verstand, "
  + "Leipzig: Meiner 1915, S. 108)";
var zaehler=0;
function schreib(){
  document.body.innerHTML += text.charAt(zaehler);
  zaehler++;
  if (zaehler<text.length) setTimeout("schreib()",100);
}
</script>
</head>
<body onload="schreib()">
</body>
</html>
</body>
</html>
```

Es folgt ein Beispiel für das Sichtbar- und unsichtbar-Machen von Mechtild Käufer (GO To Javascript)

Bitte wählen Sie die gewünschte Versandart:

Normale Post

Express

Erst, wenn auf Express geklickt wird, wird ein Zusatz sichtbar:

The screenshot shows a form with two radio buttons. The 'Express' radio button is selected, indicated by a small circle next to it. Below the radio buttons, a text box is visible containing the text 'Zusatzkosten: 15,00 DM'.

express.html

```
<html><head>
<title>JavaScript-Testdatei</title>
<script language="JavaScript" type="text/javascript">
<!--
function zeigen(n) {
  if(n=="1") {
    document.all.div2.style.visibility="visible";
  }
  else {
    document.all.div2.style.visibility="hidden";
  }
}
```

```
}
//-->
</script>

</head>
<body>
<div id="div1" style="position:absolute;top:50;left:50;">
<form>
<h2>Bitte wählen Sie die gewünschte Versandart:</h2>
<input type="radio" name="versandart" checked="checked"
onclick="zeigen(0)">Normale Post</p>
<input type="radio" name="versandart" onclick="zeigen(1)">Express</p>
</div>
<div id="div2" style="position:absolute;top:200;left:50;
visibility:hidden">
<form>
<input type="text" readonly name="kosten" value="Zusatzkosten: 15,00 DM">
</form>
</div>
</body></html>
```