
ITG und JavaScript

Im Lehrplanentwurf "Informationstechnische Grundbildung (Gymnasium)" wird gefordert, dass im Sekundarbereich I eine informationstechnische Grundbildung vermittelt wird. Die Inhalte werden dabei in das Lernangebot der bestehenden Fächer eingebettet und integriert. In der Mathematik werden u.a. einfache Algorithmen in lauffähige Programme umgesetzt. Die Entscheidung über die Programmiersprache liegt beim Fachlehrer.

HTML und JavaScript sind für die Umsetzung der Algorithmen in Lauffähige Programme besonders geeignet. Eine Hinführung an die objektorientierte Programmierung und an die Erstellung von Web-Seiten kann hier erfolgen.

Java ist eine objektorientierte vollwertige Programmiersprache, die gleichwertig mit bekannten Programmiersprachen wie C++, Pascal oder FORTRAN ist. JavaScript ist dagegen nur eingeschränkt objektorientiert. JavaScript ist objektbasiert. In JavaScript stehen weder Klassen noch Vererbung zur Verfügung. JavaScript macht Gebrauch von Objekten. Diese Objekte sind im wesentlichen fertig vordefinierte Objekte, nämlich das HTML-Dokument und dessen Elemente. Das Formular ist das wichtigste Objekt. Es besteht auch in JavaScript die Möglichkeit, eigene Objekte mit Eigenschaften und Methoden zu definieren. Diese Möglichkeiten sind aber aufgrund des geringen Sprachumfangs eingeschränkt.

Ein wesentlicher Unterschied zwischen Java und JavaScript liegt darin, dass Java-Programme, die im Quelltext vorliegen, erst kompiliert und dann als Bytecode abgespeichert werden. Der Bytecode wird von einem Bytecode-Interpreter beim Client ausgeführt. JavaScript ist dagegen eine interpretierte Sprache. Durch die fehlende Kompilierung findet keine Prüfung umfassende Prüfung auf Programmfehler statt.

JavaScript ist eine Scriptsprache, die von Netscape eingeführt wurde und von Netscape lizenziert ist. Netscape 2.0 interpretiert den JavaScript-Sprachstandard 1.0. Der MS-Internet-Explorer versteht diesen Sprachstandard seit Version 3.0.

Netscape interpretiert seit Version 3.0 den JavaScript-Standard 1.1. Der MS-Internet-Explorer interpretiert diesen Sprachumfang von JavaScript seit der Produktversion 4.0.

Netscape interpretiert seit Version 4.0 den JavaScript-Standard 1.2. Einige Befehle dieser Sprachversion werden auch vom MS-Internet-Explorer 4.0 interpretiert. Für die Umsetzung mathematischer Inhalte stellt JavaScript das Objekt Math mit entsprechenden Eigenschaften und Methoden zur Verfügung.

Objekt Math

Mit dem Objekt Math können Berechnungen, auch komplexe wissenschaftlicher oder kaufmännischer Art, durchgeführt werden. Man kann Eigenschaften und Methoden von Math direkt verwenden. Es ist darauf zu achten, dass in JavaScript Groß- und Kleinschreibung unterschieden wird. Die Eigenschaften und Methoden müssen in folgender Schreibweise formuliert werden.

`x = Math.Eigenschaft;`

`x = Math.Methode(Parameter);`

a) Eigenschaften

Eulersche Zahl	E	x = Math.E;
natürlicher Logarithmus von 2	LN2	x = Math.LN2;
natürlicher Logarithmus von 10	LN10	x = Math.LN10;
konstanter Logarithmus von 2	LOG2E	x = Math.LOG2E;
konstanter Logarithmus von 10	LOG10E	x = Math.LOG10E;
Zahl PI	PI	x = Math.PI;
Konstante für Quadratwurzel aus 0.5	SQRT1_2	x = Math.SQRT1_2;
Konstante für Quadratwurzel aus 2	SQRT2	x = Math.SQRT2;

b) Methoden

Betrag einer Zahl	abs()	x = Math.abs(Wert);
Arcus Cosinus	acos()	x = Math.acos(Wert);
Arcus Sinus	asin()	x = Math.asin(Wert);
Arcus Tangens	atan()	x = Math.atan(Wert);
nächsthöhere ganze Zahl	ceil()	x = Math.ceil(Wert);
Kosinus	cos()	x = Math.cos(Wert);
Exponentialwert	exp()	x = Math.exp(Wert);
nächstniedrige ganze Zahl	floor()	x = Math.floor(Wert);
Maximum zweier Zahlen	max()	x = Math.max(Wert1,Wert2);
Minimum zweier Zahlen	min()	x = Math.min(Wert1,Wert2);
Potenz (Zahl hoch Exponent)	pow()	x = Math.pow(Basis,Exponent);
Zufallszahl aus (0,1)	random()	x = Math.random();
kaufmännische Rundung einer Zahl	round()	x = Math.round(Wert);
Sinus	sin()	x = Math.sin(Wert);
Quadratwurzel	sqrt()	x = Math.sqrt(Wert);
Tangens	tan()	x = Math.tan(Wert);

Die Typkonvertierung in JavaScript erfolgt in der ersten Version automatisch. Ab der Version 1.2 muss gegebenenfalls die Typkonvertierung programmiert werden (siehe Kapitel 6).

Umwandlung in ganze Zahl: parseInt(zahl)
Umwandlung in reelle Zahl: parseFloat(zahl)

1. Lineare Programme

In der Prozentrechnung geht es stets um die Berechnung einer der drei Größen Prozentwert, Grundwert oder Prozentsatz. Dabei ergeben sich für die Berechnung der einzelnen Größen die folgenden Gesetzmäßigkeiten:

$$\text{Prozentwert} = \text{Grundwert} \cdot \frac{\text{Prozentsatz}}{100}$$

$$\text{Grundwert} = \text{Prozentwert} \cdot \frac{100}{\text{Prozentsatz}}$$

$$\text{Prozentsatz} = \frac{\text{Prozentwert}}{\text{Grundwert}} \cdot 100$$

Nach einer unterrichtlichen Behandlung (Klassenstufe 7) eignet sich die Prozentrechnung zur Erstellung linearer Programme. In der ITG soll an einfachen Beispielen exemplarisch an die Erstellung von Algorithmen und deren Programmierung herangeführt werden.

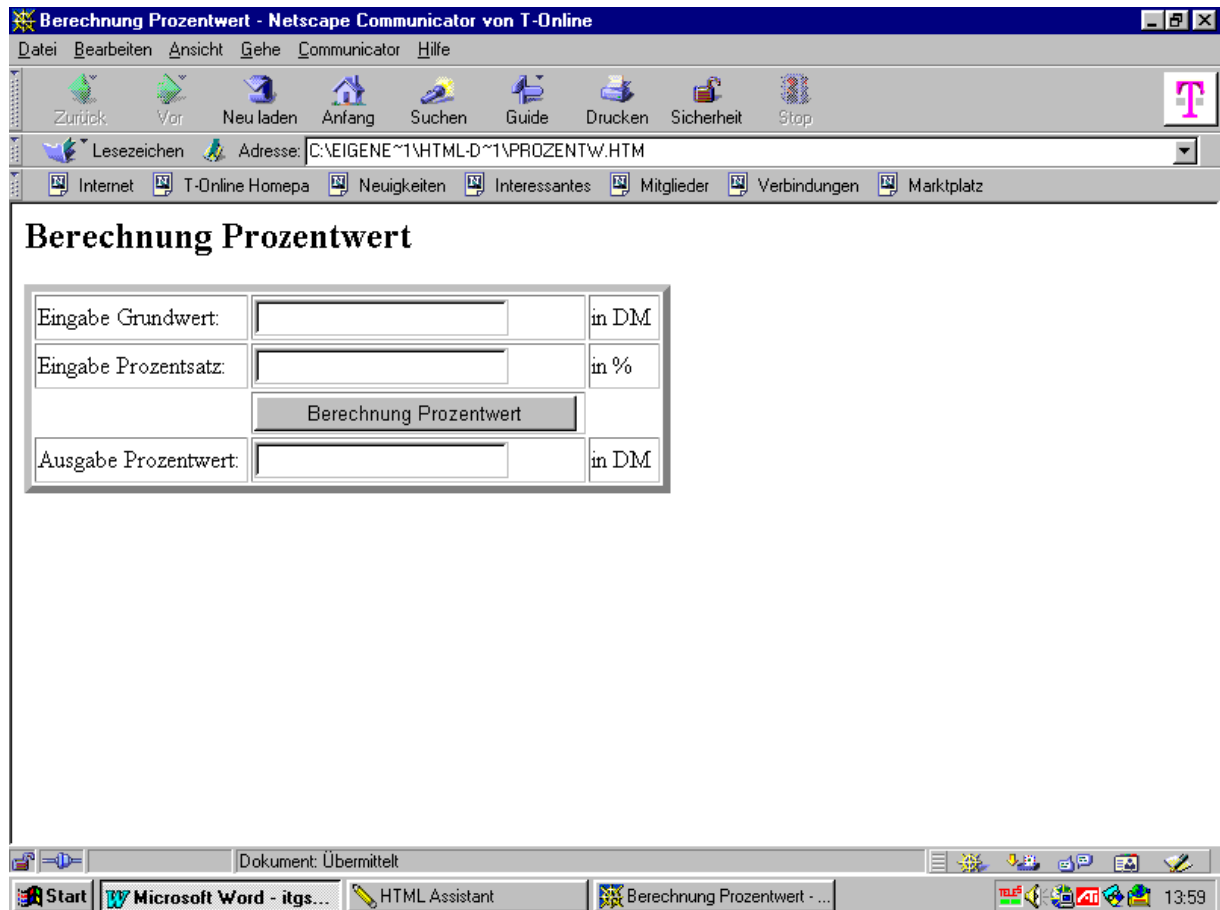
1.1. Algorithmus Prozentwert

Eingabe Grundwert
Eingabe Grundwert
$\text{Prozentwert} \leftarrow \text{Grundwert} \cdot \frac{\text{Prozentsatz}}{100}$
Ausgabe Prozentwert

1.1.1. Programm Prozentwert

Das Formular ist ein Objekt der HTML-Datei. Die einzelnen Ein- und Ausgabefelder sind Objekte mit der Eigenschaft einen Wert zu besitzen. Der Zugriff auf die Eigenschaft eines Feldes geschieht über die bei Objekten übliche Notation mit Hilfe der Bezeichnung über Objektvariablenname, gefolgt von einem Punkt und der Objekteigenschaft.

form.feldname.value bzw. feldname.value oder formularname.feldname.value



```

<HTML>
<HEAD>
<TITLE>Berechnung Prozentwert</TITLE>
</HEAD>
<BODY>
<H2>Berechnung Prozentwert</H2>

<FORM>
  <TABLE BORDER="5">
    <TR>
      <TD> Eingabe Grundwert: </TD>
      <TD><INPUT TYPE="Text" NAME="Grundwert" VALUE="" size="20"> </TD>
      <TD> in DM </TD>
    </TR>
    <TR>
      <TD> Eingabe Prozentsatz: </TD>
      <TD> <INPUT TYPE="Text" NAME="Prozentsatz" VALUE="" size="20"></TD>
      <TD> in % </TD>
    </TR>
    <TR>
      <TD> </TD>
      <TD> <INPUT TYPE="Button" VALUE="Berechnung Prozentwert"
      onClick="Prozentwert.value=Grundwert.value*Prozentsatz.value/100"> </TD>
      <TD> </TD>
    </TR>
    <TR>
      <TD> Ausgabe Prozentwert: </TD>
      <TD> <INPUT TYPE="Text" NAME="Prozentwert" VALUE="" size="20" ></TD>
  </TABLE>

```

```

        <TD> in DM </TD>
    </TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

1.1.2. Programm Prozentwert (mit Hilfe einer JavaScript-function)

Die Berechnung des Prozentsatzes kann auch mit Hilfe einer JavaScript-Funktion ausgeführt werden. Dabei sind die übergebenen Parameter Objekte. In der Klassenstufe 7 steht der Funktionsbegriff noch nicht zur Verfügung. Deshalb wird man die JavaScript-Funktion erst in späteren Klassenstufen einführen.

```

<HTML>
<HEAD>
<TITLE>Berechnung Prozentwert</TITLE>

<SCRIPT language="JavaScript">
    function Prozentwert(Grundwert, Prozentsatz, W)
    {
        W.value=Grundwert.value*Prozentsatz.value/100;
    }
</SCRIPT>

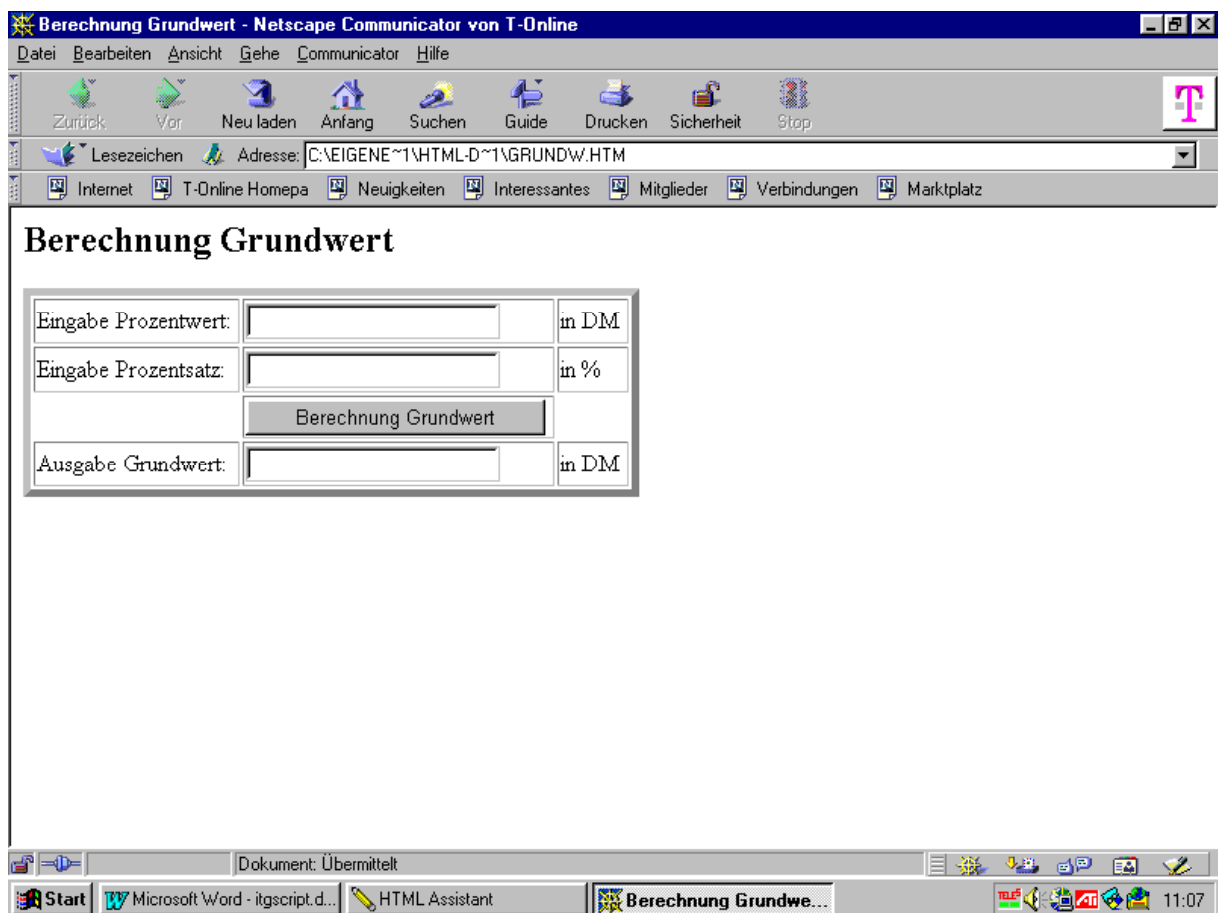
</HEAD>
<BODY>
<H2>Berechnung Prozentwert</H2>
<FORM>
    <TABLE BORDER="5">
        <TR>
            <TD> Eingabe Grundwert: </TD>
            <TD><INPUT TYPE="Text" NAME="Grundwert" VALUE="" size="20"> </TD>
            <TD> in DM </TD>
        </TR>
        <TR>
            <TD> Eingabe Prozentsatz: </TD>
            <TD> <INPUT TYPE="Text" NAME="Prozentsatz" VALUE="" size="20"></TD>
            <TD> in % </TD>
        </TR>
        <TR>
            <TD> </TD>
            <TD> <INPUT TYPE="Button" VALUE="Berechnung Prozentwert"
            onClick="Prozentwert(Grundwert,Prozentsatz,W)"> </TD>
            <TD> </TD>
        </TR>
        <TR>
            <TD> Ausgabe Prozentwert: </TD>
            <TD> <INPUT TYPE="Text" NAME="W" VALUE="" size="20" ></TD>
            <TD> in DM </TD>
        </TR>
    </TABLE>
</FORM>
</BODY>
</HTML>

```

1.2. Algorithmus Berechnung Grundwert

Eingabe Prozentwert
Eingabe Prozentsatz
$\text{Grundwert} \leftarrow \text{Prozentwert} \cdot \frac{100}{\text{Prozentsatz}}$
Ausgabe Grundwert

1.2.1 Programm Grundwert



```

<HTML>
<HEAD>
<TITLE>Berechnung Grundwert</TITLE>

<SCRIPT language="JavaScript">
    function Grundwert(Prozentwert, Prozentsatz,G)
    {
        G.value=Prozentwert.value*100/Prozentsatz.value;
    }
</SCRIPT>

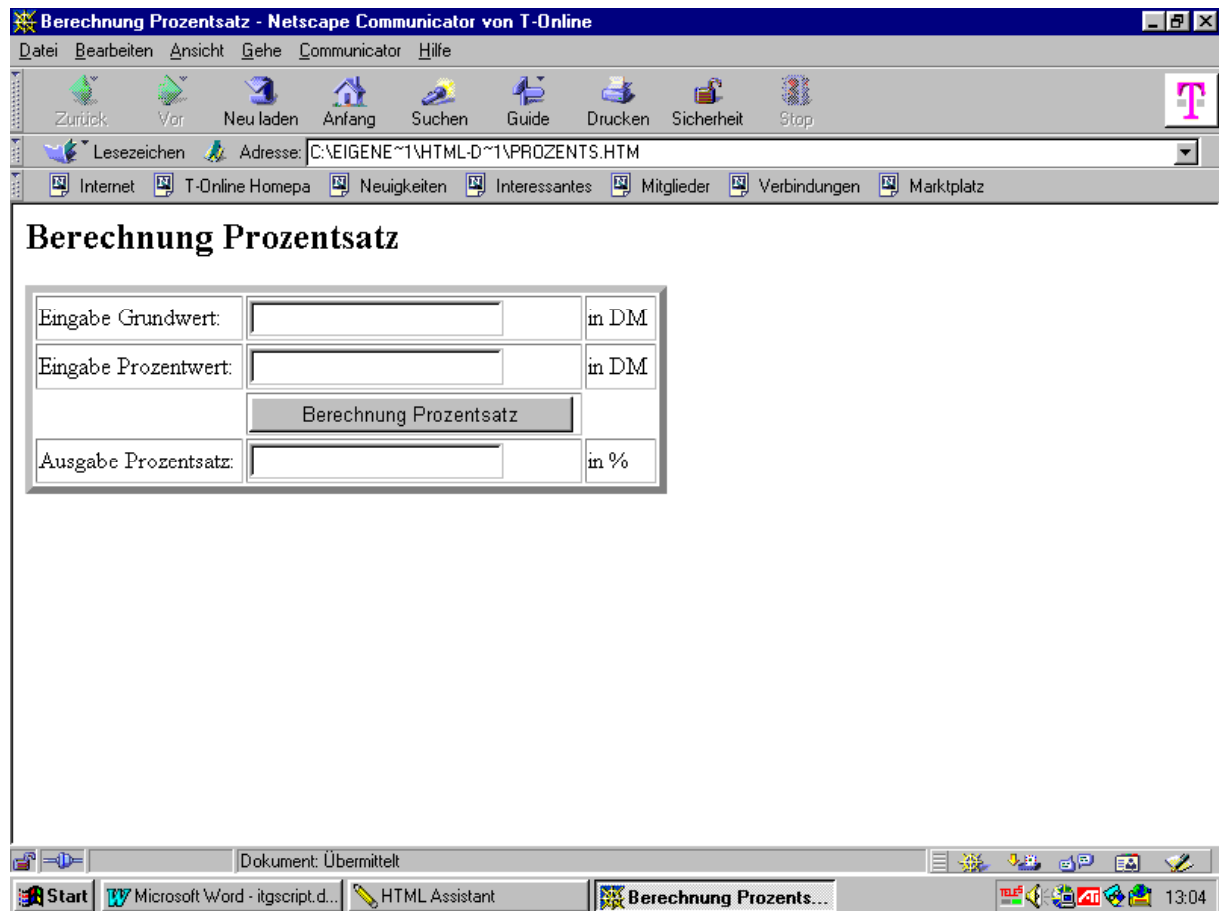
</HEAD>
<BODY>
<H2>Berechnung Grundwert</H2>
<FORM>
<TABLE BORDER="5">
    <TR>
        <TD> Eingabe Prozentwert: </TD>
        <TD><INPUT TYPE="Text" NAME="Prozentwert" VALUE="" size="20"> </TD>
        <TD> in DM </TD>
    </TR>
    <TR>
        <TD> Eingabe Prozentsatz: </TD>
        <TD> <INPUT TYPE="Text" NAME="Prozentsatz" VALUE="" size="20"></TD>
        <TD> in % </TD>
    </TR>
    <TR>
        <TD> </TD>
        <TD> <INPUT TYPE="Button" VALUE="Berechnung Grundwert"
            onClick="Grundwert(Prozentwert, Prozentsatz, G)"> </TD>
        <TD> </TD>
    </TR>
    <TR>
        <TD> Ausgabe Grundwert: </TD>
        <TD> <INPUT TYPE="Text" NAME="G" VALUE="" size="20" ></TD>
        <TD> in DM </TD>
    </TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

1.3. Berechnung Prozentsatz

Eingabe Grundwert
Eingabe Prozentwert
$\text{Prozentsatz} \leftarrow \text{Prozentwert} \cdot \frac{100}{\text{Grundwert}}$
Ausgabe Prozentsatz

1.2.2 Programm Prozentsatz



```
<HTML>
<HEAD>
<TITLE>Berechnung Prozentsatz</TITLE>

<SCRIPT language="JavaScript">
    function Prozentsatz(Grundwert,Prozentwert,P)
    {
        P.value=Prozentwert.value*100/Grundwert.value;
    }
</SCRIPT>

</HEAD>
<BODY>
<H2>Berechnung Prozentsatz</H2>
<FORM>
<TABLE BORDER="5">
    <TR>
        <TD> Eingabe Grundwert: </TD>
        <TD><INPUT TYPE="Text" NAME="Grundwert" VALUE="" size="20"> </TD>
        <TD> in DM </TD>
    </TR>
    <TR>
        <TD> Eingabe Prozentwert: </TD>
        <TD> <INPUT TYPE="Text" NAME="Prozentwert" VALUE="" size="20"></TD>
```



```

        <TD> in DM </TD>
</TR>
<TR>
    <TD> </TD>
    <TD> <INPUT TYPE="Button" VALUE="Berechnung Prozentsatz"
        onClick="Prozentsatz(Grundwert, Prozentwert, P)"> </TD>
    <TD> </TD>
</TR>
<TR>
    <TD> Ausgabe Prozentsatz: </TD>
    <TD> <INPUT TYPE="Text" NAME="P" VALUE="" size="20" ></TD>
    <TD> in % </TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

2. Verzweigte Programme

In der Klassenstufe 9 kann am Beispiel der quadratischen Gleichung an verzweigte Algorithmen herangeführt werden. Normalerweise wird im Unterricht die p-q-Formel behandelt, aber auch die Lösungsformel der allgemeinen quadratischen Gleichung ist für die algorithmische Umsetzung geeignet.

einseitige Verzweigung: if (Bedingung)
 {
 Anweisungen (then)
 }

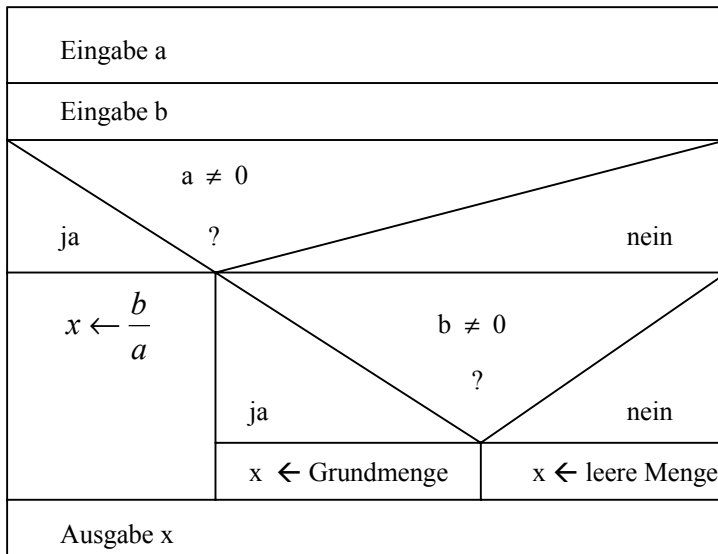
zweiseitige Verzweigung: if (Bedingung)
 {
 Anweisungen (then)
 }
 else
 {
 Anweisungen (else)
 }

Vergleichsoperatoren: < kleiner als
 <= kleiner oder gleich
 > größer als
 >= größer oder gleich
 != ungleich
 == gleich

Boolesche Operatoren: && und (and)
 || oder (or)
 ! nicht (not)

2.1. Lineare Gleichung $ax = b$

2.1.1. Algorithmus



2.1.2. Programm

The screenshot shows a Netscape Communicator window titled "Lineare Gleichun $ax = b$ - Netscape Communicator von T-Online". The address bar shows "file:///C:/beispiele/javascript/Gleich1.htm". The main content area displays the title "Lineare Gleichun $ax = b$ " and a form with the following fields and buttons:

Eingabe a:	<input type="text" value="6"/>
Eingabe b:	<input type="text" value="24.2"/>
<input type="button" value="Lösung der Gleichung"/>	
Lösung x:	<input type="text" value="4.033333333333333"/>

The Windows taskbar at the bottom shows the Start button, Microsoft Word, and the Netscape Communicator window. The system tray shows the time as 17:32.

```

<HTML>
<HEAD>
<TITLE>Lineare Gleichung ax = b</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function Gleichung(a,b,x)
{
  if (a.value != 0)
  {
    x.value = b.value/a.value;
  }
  else
  {
    if (b.value == 0)
    {
      x.value = "Grundmenge";
    }
    else
    {
      x.value = "leere Menge";
    }
  }
}
</SCRIPT>

</HEAD>
<BODY>
<CENTER>
<H2>Lineare Gleichung ax = b</H2>
<FORM>
<TABLE BORDER="5">
  <TR>
    <TD> Eingabe a: </TD>
    <TD> <INPUT TYPE="Text" NAME="a" VALUE=" " SIZE="20"></TD>
  </TR>
  <TR>
    <TD> Eingabe b: </TD>
    <TD> <INPUT TYPE="Text" NAME="b" VALUE=" " SIZE="20"></TD>
  </TR>
  <TR>
    <TD> </TD>
    <TD> <INPUT TYPE="Button" VALUE="Lösung der Gleichung"
      onClick="Gleichung(a,b,x)"> </TD>
  </TR>
  <TR>
    <TD> Lösung x </TD>
    <TD> <INPUT TYPE="Text" NAME="x" VALUE=" " SIZE="20"></TD>
  </TR>
</TABLE>
</FORM>
</CENTER>
</BODY>
</HTML>

```

2.2. Quadratische Gleichung (p-q-Formel)

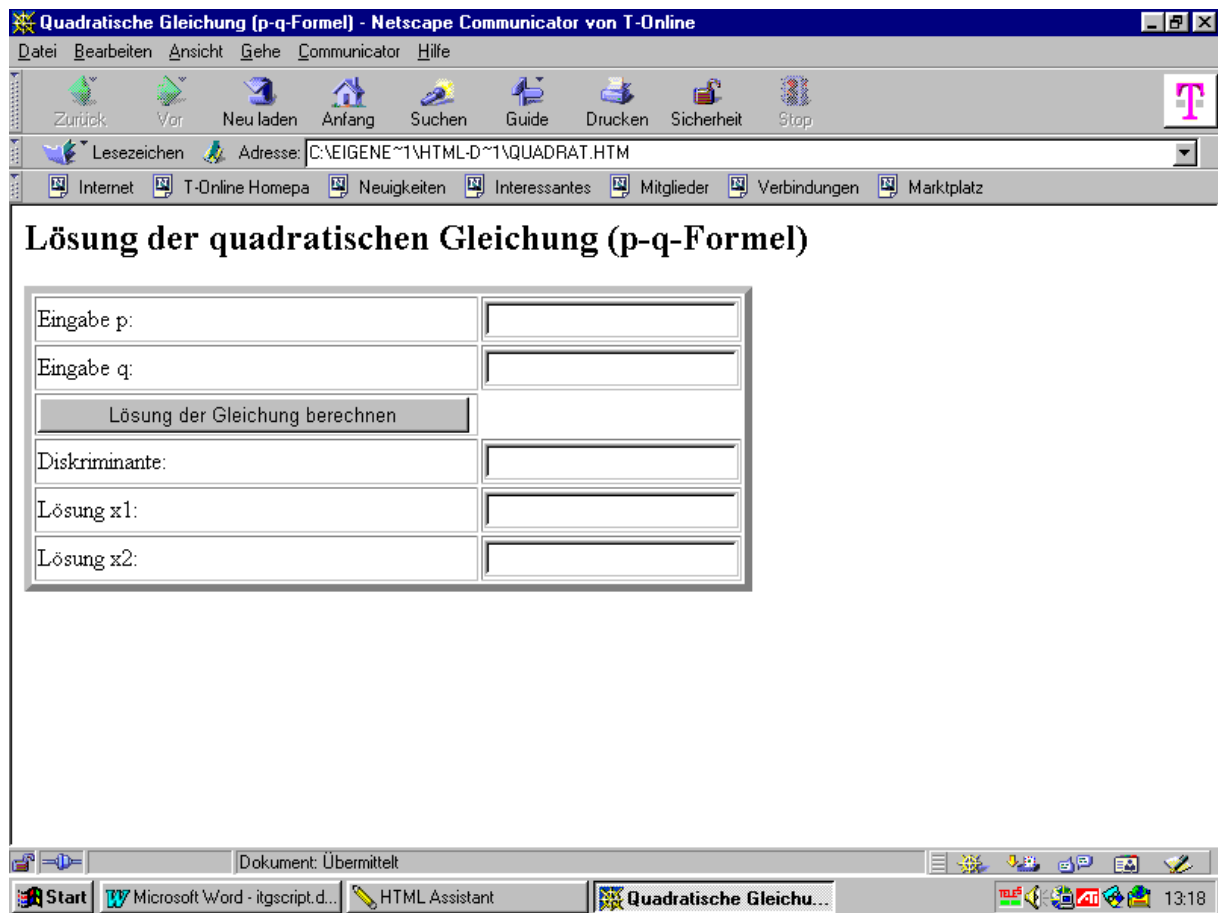
Quadratische Gleichung: $x^2 + px + q = 0$

Lösungsformel: $x_1 = -\frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - q}$ $x_2 = -\frac{p}{2} - \sqrt{\left(\frac{p}{2}\right)^2 - q}$

2.2.1. Algorithmus

Eingabe p		
Eingabe q		
$D \leftarrow \left(\frac{p}{2}\right)^2 - q$		
D > 0		
ja	nein	
$x_1 = -\frac{p}{2} + \sqrt{D}$	D = 0	
$x_2 = -\frac{p}{2} - \sqrt{D}$	ja	nein
	$x = -\frac{p}{2}$	
Ausgabe x ₁		
Ausgabe x ₂	Ausgabe x	Ausgabe keine Lösung

2.2.2. Programm p-q-Formel



```
<HTML>
<HEAD>
<TITLE>Quadratische Gleichung (p-q-Formel)</TITLE>
```

```
<SCRIPT language="JavaScript">
function loesung(p,q,D,x1,x2)
{
    D.value=p.value*p.value/4-q.value;
    if (D.value>0)
    {
        x1.value=-p.value/2+Math.sqrt(D.value);
        x2.value=-p.value/2-Math.sqrt(D.value);
    }
    else
    {
        if (D.value==0)
        {
            x1.value=-p.value/2;
            x2.value=" ";
        }
        else
        {
            x1.value="keine Lösung";
            x2.value=" ";
        }
    }
}
```

```

    }
  }
</SCRIPT>

</HEAD>
<BODY>
<H2>Lösung der quadratischen Gleichung (p-q-Formel)</H2>
<FORM>
<TABLE BORDER="5">
  <TR>
    <TD>Eingabe p:</TD>
    <TD><INPUT TYPE="Text" NAME="p" VALUE="" SIZE="20"></TD>
  </TR>
  <TR>
    <TD>Eingabe q:</TD>
    <TD><INPUT TYPE="Text" NAME="q" VALUE="" SIZE="20"></TD>
  </TR>
  <TR>
    <TD><INPUT TYPE="Button" VALUE="Lösung der Gleichung berechnen"
onClick="loesung(p,q,D,x1,x2)" </TD>
    <TD></TD>
  </TR>
  <TR>
    <TD> Diskriminante: </TD>
    <TD><INPUT TYPE="text" NAME="D" VALUE="" SIZE="20"></TD>
  </TR>
  <TR>
    <TD>Lösung x1:</TD>
    <TD><INPUT TYPE="Text" NAME="x1" VALUE="" SIZE=20"></TD>
  </TR>
  <TR>
    <TD>Lösung x2:</TD>
    <TD><INPUT TYPE="Text" NAME="x2" VALUE="" SIZE=20"></TD>
  </TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

2.3. Lineares Gleichungssystem mit zwei Variablen

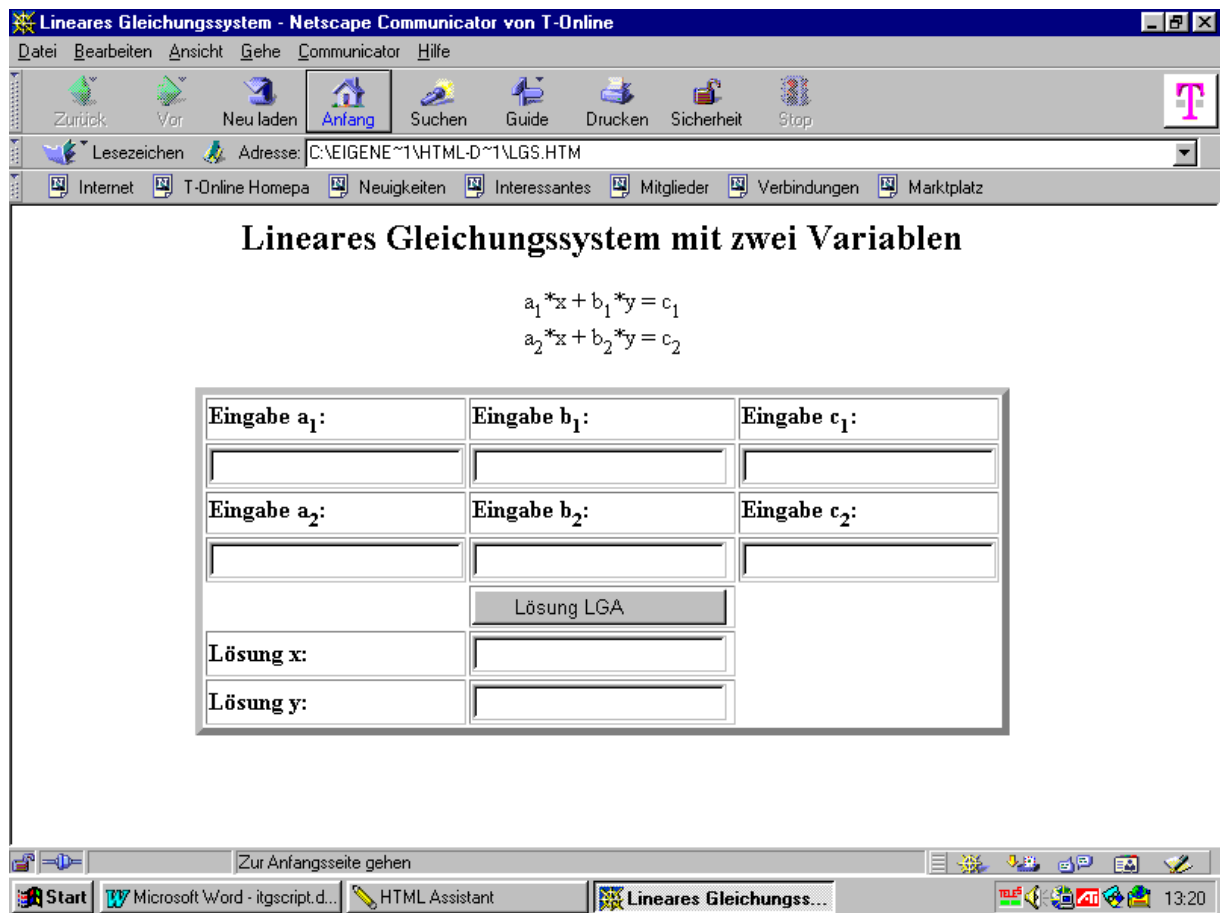
Lineare Gleichungssysteme bilden einen weiteren Themenschwerpunkt in der 9. Klassenstufe. Die Lösungsformel kann zum Ende der Unterrichtseinheit allgemein hergeleitet werden und in ein lauffähiges Programm umgesetzt werden.

$$\begin{array}{l}
 a_1x + b_1y = c_1 \\
 a_2x + b_2y = c_2
 \end{array}
 \quad
 x = \frac{b_2c_1 - b_1c_2}{a_1b_2 - a_2b_1}
 \quad
 y = \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}$$

2.3.1. Algorithmus

Eingabe $a_1, a_2, b_1, b_2, c_1, c_2$		
Nenner $\leftarrow a_1 b_2 - a_2 b_1$		
Zähler_x $\leftarrow b_2 c_1 - b_1 c_2$		
Zähler_y $\leftarrow a_1 c_2 - a_2 c_1$		
Nenner $\neq 0$		
ja	nein	
$x \leftarrow \frac{\text{Zähler}_x}{\text{Nenner}}$	(Zähler_x=0) und (Zähler_y=0)	
$y \leftarrow \frac{\text{Zähler}_y}{\text{Nenner}}$	ja	nein
Ausgabe x	unendliche viele Lösungen	keine Lösung
Ausgabe y		

2.3.2 Programm Lineares Gleichungssystem



```

<HTML>
<HEAD>
<TITLE>Lineares Gleichungssystem</TITLE>

<SCRIPT language="JavaScript">
function loesung(a1,a2,b1,b2,c1,c2,x,y)
{
    var Nenner=a1.value*b2.value-a2.value*b1.value;
    var Zaehler_x=b2.value*c1.value-b1.value*c2.value;
    var Zaehler_y=a1.value*c2.value-a2.value*c1.value;
    if (Nenner != 0)
    {
        x.value=Zaehler_x/Nenner;
        y.value=Zaehler_y/Nenner;
    }
    else
    {
        if ((Zaehler_x == 0) & (Zaehler_y == 0))
        {
            x.value="Unendlich viele L&ouml;sungen";
            y.value=" ";
        }
        else
        {
            x.value="Keine L&ouml;sung";
            y.value=" ";
        }
    }
}

```



```

    }
  }
}
</SCRIPT>

</HEAD>
<BODY>
<CENTER>
<H2>Lineares Gleichungssystem mit zwei Variablen</H2>
<P><P>
a<SUB>1</SUB>*x + b<SUB>1</SUB>*y = c<SUB>1</SUB><BR>
a<SUB>2</SUB>*x + b<SUB>2</SUB>*y = c<SUB>2</SUB><BR>
<P><P>
<FORM>
<TABLE BORDER="5">
  <TR>
    <TD><B>Eingabe a<SUB>1</SUB>:</B></TD>
    <TD><B>Eingabe b<SUB>1</SUB>:</B></TD>
    <TD><B>Eingabe c<SUB>1</SUB>:</B></TD>
  </TR>
  <TR>
    <TD><INPUT TYPE="Text" NAME="a1" VALUE="" SIZE="20"></TD>
    <TD><INPUT TYPE="Text" NAME="b1" VALUE="" SIZE="20"></TD>
    <TD><INPUT TYPE="Text" NAME="c1" VALUE="" SIZE="20"></TD>
  </TR>
  <TR>
    <TD><B>Eingabe a<SUB>2</SUB>:</B></TD>
    <TD><B>Eingabe b<SUB>2</SUB>:</B></TD>
    <TD><B>Eingabe c<SUB>2</SUB>:</B></TD>
  </TR>
  <TR>
    <TD><INPUT TYPE="Text" NAME="a2" VALUE="" SIZE="20"></TD>
    <TD><INPUT TYPE="Text" NAME="b2" VALUE="" SIZE="20"></TD>
    <TD><INPUT TYPE="Text" NAME="c2" VALUE="" SIZE="20"></TD>
  </TR>
  <TR>
    <TD></TD>
    <TD><INPUT TYPE="Button" VALUE="L&ouml;sung LGA "
      onClick="loesung(a1,a2,b1,b2,c1,c2,x,y)" </TD>
    <TD></TD>
  </TR>
  <TR>
    <TD><B>L&ouml;sung x:</B></TD>
    <TD><INPUT TYPE="Text" NAME="x" VALUE="" SIZE=20"></TD>
    <TD></TD>
  </TR>
  <TR>
    <TD><B>L&ouml;sung y:</B></TD>
    <TD><INPUT TYPE="Text" NAME="y" VALUE="" SIZE=20"></TD>
    <TD></TD>
  </TR>
</TABLE>
</FORM>
</CENTER>
</BODY>
</HTML>

```

3. Wiederholungen in JavaScript (gezählte Wiederholung)

In der 10. Klassenstufe kann an die Kontrollstrukturen der gezählten Wiederholung, der annehmenden und der abweisenden Wiederholung herangeführt werden.

for (Initialisierung; Bedingung; Schrittweite)

```
{  
  Schleifenkörper  
}
```

for (k=1; k <=100; k++)

```
{  
  summe=summe + k  
}
```

3.1. Berechnung der Zahl PI (Untersumme)

Im Unterricht wird die Summenformel für die Untersumme hergeleitet. Die Anzahl der Wiederholungen ist bekannt und die Realisierung erfolgt mit Hilfe der gezählten Wiederholung.

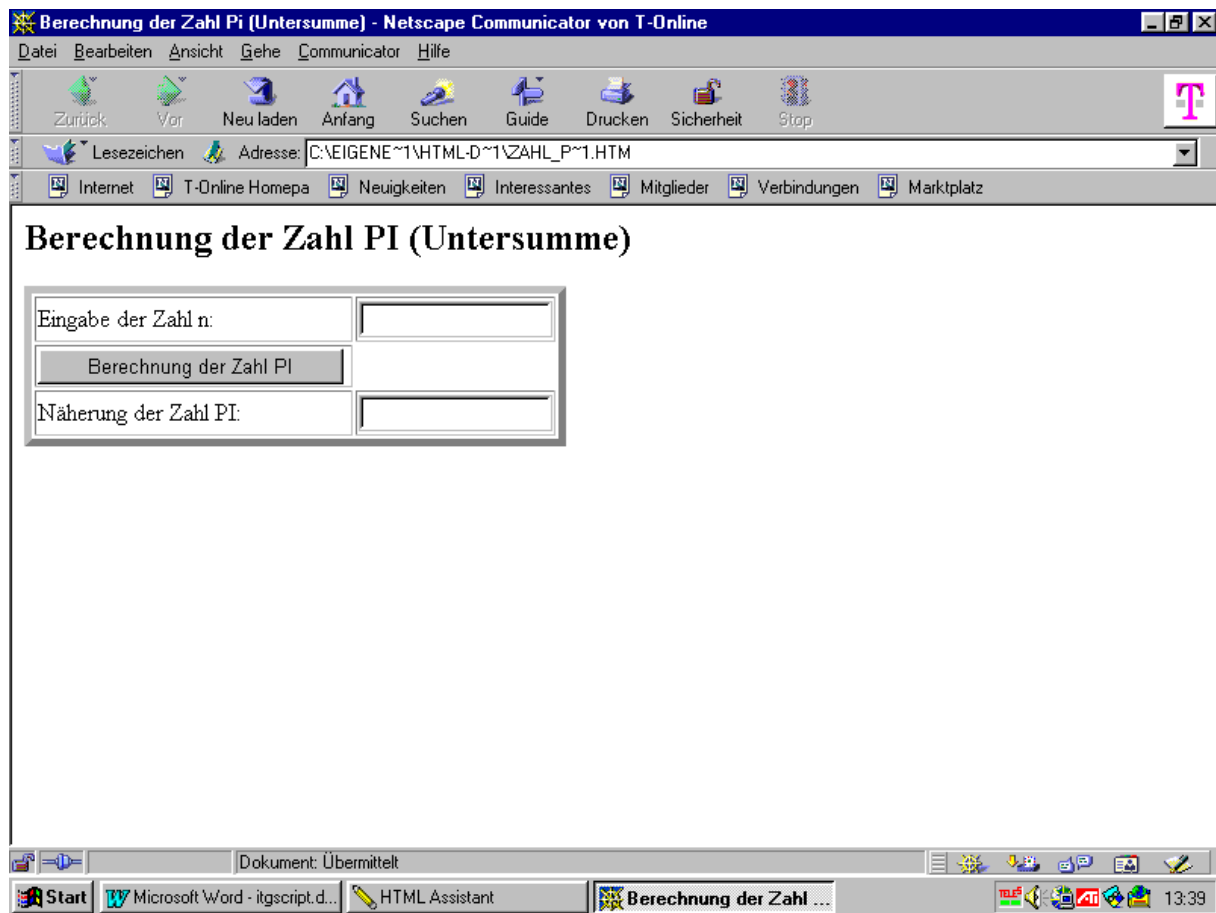
allgemeine Formel (Untersumme):

$$A_u(n) = \frac{4}{n} \left(\sqrt{1 - \left(\frac{1}{n}\right)^2} + \sqrt{1 - \left(\frac{2}{n}\right)^2} + \sqrt{1 - \left(\frac{3}{n}\right)^2} + \dots + \sqrt{1 - \left(\frac{n-1}{n}\right)^2} \right)$$

3.1.1. Algorithmus

Eingabe n
Summe ← 0
für k ← 1 bis n - 1
tue $Summe \leftarrow Summe + \sqrt{1 - \left(\frac{k}{n}\right)^2}$
$PI \leftarrow \frac{4 \cdot Summe}{n}$
Ausgabe PI

3.1.2. Programm zur Untersumme



```
<HTML>
<HEAD>
<TITLE>Berechnung der Zahl Pi (Untersumme)</TITLE>

<SCRIPT language="JavaScript">
  function ZahlPI(n,PI)
  {
    var summe, k,m;
    summe=0;
    m=n.value-1;
    for (k=1; k<=m; k++)
    {
      summe=summe+Math.sqrt(1-(k/m)*(k/m));
      // Das Mathematik-Objekt muss Math geschrieben werden
    }
    summe=4*summe/m;
    PI.value=summe;
  }
</SCRIPT>

</HEAD>
<BODY>
<H2>Berechnung der Zahl Pi (Untersumme)</H2>
<P>
<FORM>
<TABLE BORDER="5">
```

```

<TR>
  <TD> Eingabe der Zahl n:</TD>
  <TD><INPUT TYPE="Text" NAME="n" VALUE="" SIZE="15"></TD>
</TR>
<TR>
  <TD> <INPUT TYPE="Button" VALUE="Berechnung der Zahl PI" onClick="ZahlPI(n,PI)">
  </TD>
  <TD> </TD>
</TR>
<TR>
  <TD> N&auml;herung der Zahl PI: </TD>
  <TD> <INPUT TYPE="Text" NAME="PI" VALUE="" SIZE="15"></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

3.2. Berechnung der Zahl PI (Obersumme)

Für die Berechnung der Obersumme kann im Unterricht folgende Formel hergeleitet werden:

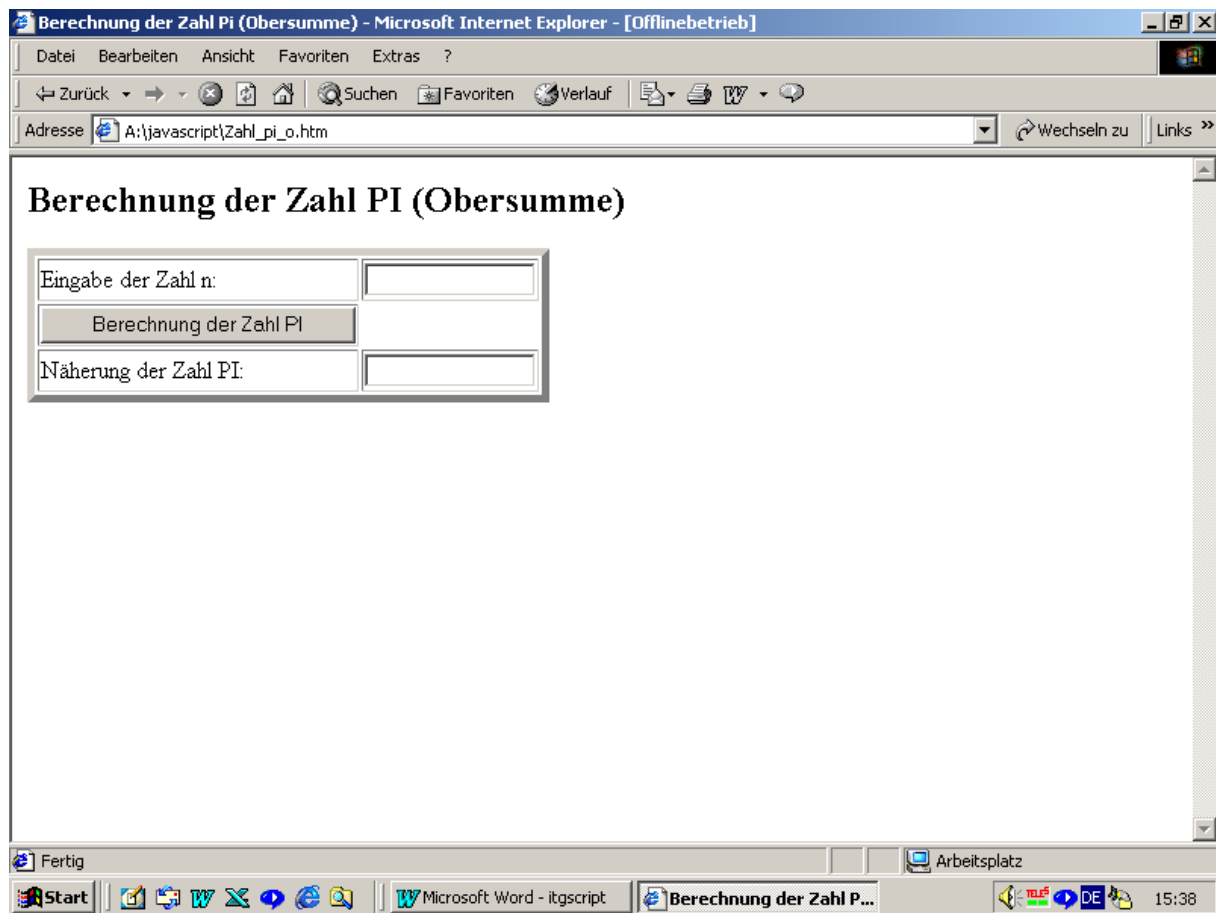
allgemeine Formel (Obersumme):

$$A_o(n) = \frac{4}{n} \left(1 + \sqrt{1 - \left(\frac{1}{n}\right)^2} + \sqrt{1 - \left(\frac{2}{n}\right)^2} + \sqrt{1 - \left(\frac{3}{n}\right)^2} + \dots + \sqrt{1 - \left(\frac{n-1}{n}\right)^2} \right)$$

3.2.1. Algorithmus

Eingabe n	
Summe ← 1	
für k ← 1 bis n - 1	
tue	$summe \leftarrow summe + \sqrt{1 - \left(\frac{k}{n}\right)^2}$
$PI \leftarrow \frac{4 \cdot summe}{n}$	
Ausgabe PI	

3.2.2. Programm Berechnung der Zahl PI (Obersumme)



```
<HTML>
<HEAD>
<TITLE>Berechnung der Zahl Pi (Untersumme)</TITLE>

<SCRIPT language="JavaScript">
    function ZahlPI(n,PI)
    {
        var summe, k,m;
        summe=1;
        m=n.value-1;
        for (k=1; k<=m; k++)
        {
            summe=summe+Math.sqrt(1-(k/m)*(k/m));
            // Das Mathematik-Objekt muss Math geschrieben werden
        }
        summe=4*summe/m;
        PI.value=summe;
    }
</SCRIPT>

</HEAD>
<BODY>
<H2>Berechnung der Zahl PI (Untersumme)</H2>
<P>
<FORM>
<TABLE BORDER="5">
```

```

<TR>
  <TD> Eingabe der Zahl n:</TD>
  <TD><INPUT TYPE="Text" NAME="n" VALUE="" SIZE="15"></TD>
</TR>
<TR>
  <TD> <INPUT TYPE="Button" VALUE="Berechnung der Zahl PI" onClick="ZahlPI(n,PI)">
  </TD>
  <TD> </TD>
</TR>
<TR>
  <TD> Näherung der Zahl PI: </TD>
  <TD> <INPUT TYPE="Text" NAME="PI" VALUE="" SIZE="15"></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

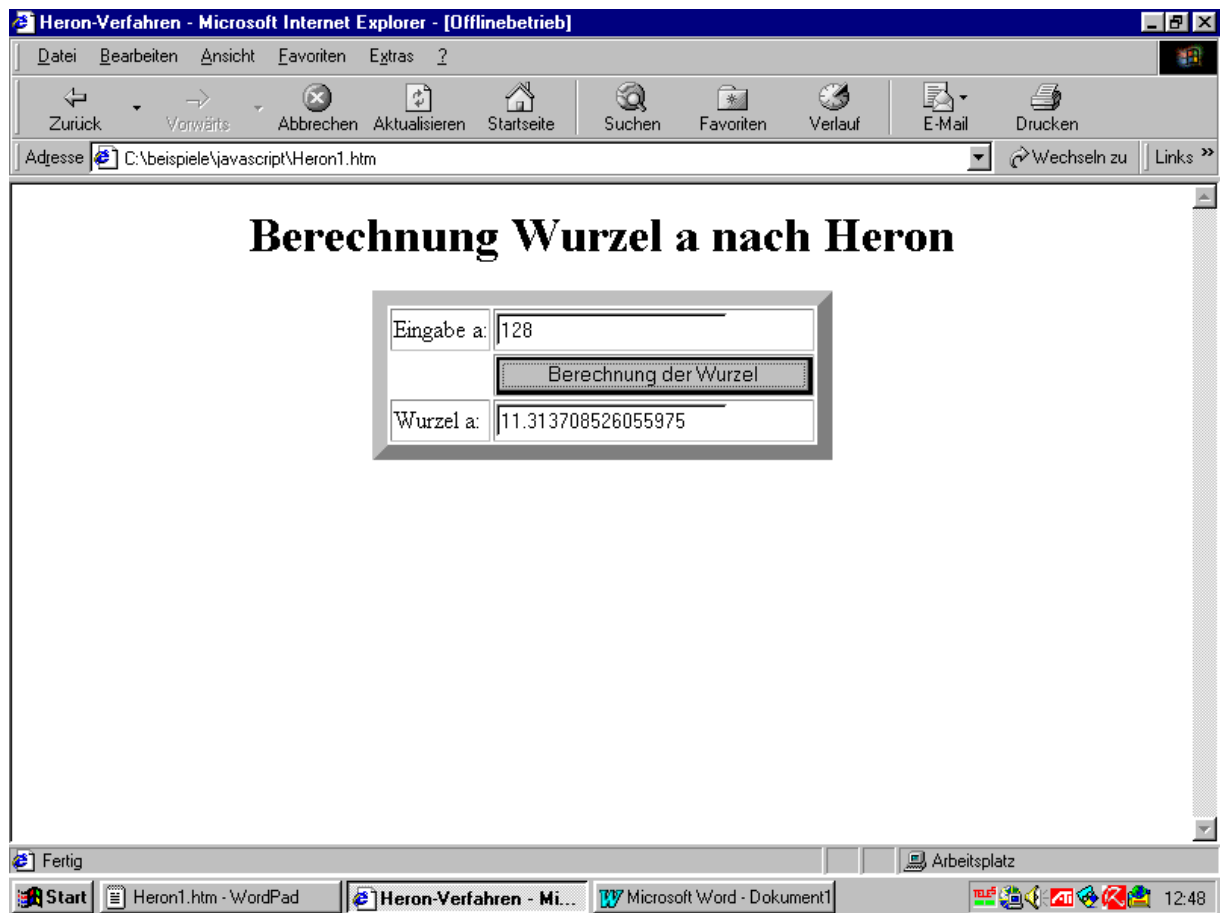
3.3. Heron-Verfahren \sqrt{a}

Mit Hilfe einer gezählten Wiederholung kann die Wurzel näherungsweise mit hinreichender Genauigkeit berechnet werden.

3.3.1 Algorithmus

Eingabe a	
xalt \leftarrow 1	
yalt \leftarrow a	
für k \leftarrow 1 bis 10	
tue	$x_{neu} \leftarrow \frac{1}{2}(x_{alt} + y_{alt})$
	$y_{neu} \leftarrow \frac{a}{x_{neu}}$
	$x_{alt} \leftarrow x_{neu}$
	$y_{alt} \leftarrow y_{neu}$
Wurzel(a) \leftarrow xneu	

3.3.2. Programm Heron



```
<HTML>
<HEAD>
<TITLE>Heron-Verfahren</TITLE>
<SCRIPT language="JavaScript">
function heron(a,x)
{
    var xalt, xneu, yalt, yneu;
    xalt=1;
    yalt=a.value;
    for(k=1;k<=10;k++)
    {
        xneu=(xalt+yalt)/2;
        yneu=a.value/xneu;
        xalt=xneu;
        yalt=yneu;
    }
    x.value=xneu;
}
</SCRIPT>
</HEAD>
<BODY >
<CENTER>
<H1>Berechnung Wurzel a nach Heron</H1>
<FORM>
<TABLE BORDER="10" >
    <TR>
```

```

        <TD>Eingabe a:  </TD>
        <TD><INPUT TYPE="Text" NAME="a" VALUE="" SIZE="20"></TD>
    </TR>
    <TR>
        <TD></TD>
        <TD><INPUT TYPE="Button" VALUE="Berechnung der Wurzel"
            onClick="heron(a,x)"></TD>
    </TR>
    <TR>
        <TD>Wurzel a:  </TD>
        <TD><INPUT TYPE="Text" NAME="x" VALUE="" SIZE="20"></TD>
    </TR>
</TABLE>
</FORM>
</CENTER>
</BODY>
</HTML>

```

4. Wiederholungen mit JavaScript (abweisende Wiederholung)

Mit Hilfe entsprechender Beispiele können im Mathematikunterricht auch die abweisende bzw. die annehmende Wiederholung behandelt werden.

Am Beispiel der näherungsweise iterativen Berechnung der Quadratwurzel nach Heron können beide Schleifen eingeführt werden. Die Näherungsformel kann schon in der 9. Klasse bei der Einführung der reellen Zahlen thematisiert werden.

Das Verfahren führt überraschend schnell zu genauen Näherungswerten. Es wurde von dem griechischen Mathematiker, Physiker und Ingenieur Heron (1. Jahrhundert nach Chr.) entwickelt.

$$x_{neu} \leftarrow \frac{1}{2} \left(x_{alt} + \frac{a}{x_{alt}} \right) \text{ mit Startwert } x_{alt} = a$$

```

while (Bedingung)
{
    Anweisungsfolge
}

```

Das Schlüsselwort while leitet die Wiederholung ein. Die Bedingung steht in runden Klammern. Ist die Bedingung erfüllt, wird die Anweisungsfolge ausgeführt. Ist die Bedingung nicht erfüllt, wird der Programmblock nach while ausgeführt, d.h. das Programm geht mit den Befehlen weiter, die nach der geschlossenen geschweiften Klammer der while-Schleife stehen.

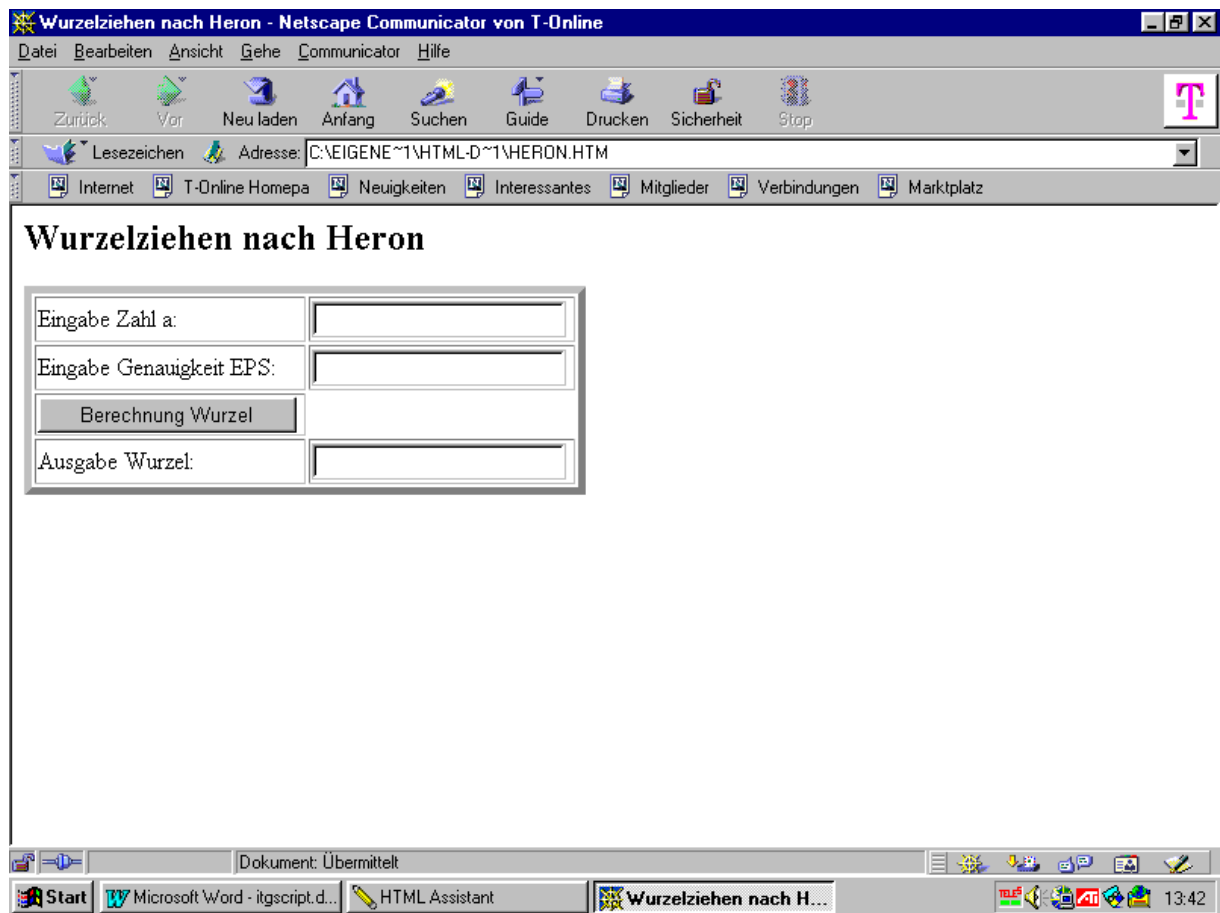
4.1. Wurzelziehen nach Heron $x = \sqrt{a}$

$$x_{neu} \leftarrow \frac{1}{2} \left(x_{alt} + \frac{a}{x_{alt}} \right) \text{ mit Startwert } x_{alt} = a$$

4.1.1. Algorithmus

Eingabe Zahl a	
Eingabe Genauigkeit EPS	
$x_{neu} \leftarrow a$	
$x_{alt} \leftarrow 0$	
solange $ x_{neu} - x_{alt} \geq EPS$	
tue	$x_{alt} \leftarrow x_{neu}$
	$x_{neu} \leftarrow \frac{1}{2} \left(x_{alt} + \frac{a}{x_{alt}} \right)$
Ausgabe	xneu

4.1.2. Programm



```
<HTML>
<HEAD>
<TITLE>Wurzelziehen nach Heron</TITLE>

<SCRIPT language="JavaScript">
  function Wurzel(a, EPS, x)
  {
    var xalt=0;
    var xneu=parseFloat(a.value);           //Typkonvertierung
    var genau=parseFloat(EPS.value);        //Typkonvertierung
    while (Math.abs(xneu-xalt) >= genau)
    {
      xalt=xneu;
      xneu=0.5*(xalt+a.value/xalt);
    }
    x.value=xneu;
  }
</SCRIPT>

</HEAD>
<BODY>
<H2>Wurzelziehen nach Heron</H2>
<FORM>
<TABLE BORDER="5">
  <TR>
    <TD> Eingabe Zahl a: </TD>
```

```

        <TD><INPUT TYPE="Text" NAME="a" VALUE="" size="20"> </TD>
    </TR>
    <TR>
        <TD> Eingabe Genauigkeit EPS: </TD>
        <TD> <INPUT TYPE="Text" NAME="EPS" VALUE="" size="20"></TD>
    </TR>
    <TR>
        <TD> <INPUT TYPE="Button" VALUE="Berechnung Wurzel"
            onClick="Wurzel(a,EPS,x)"> </TD>
    </TR>
    <TR>
        <TD> Ausgabe Wurzel: </TD>
        <TD> <INPUT TYPE="Text" NAME="x" VALUE="" size="20" ></TD>
    </TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

5. Wiederholungen mit JavaScript (annehmende Wiederholung)

```

do
{
    Anweisungsfolge
}
while (Bedingung)

```

Das Schlüsselwort `do` leitet die Wiederholung ein. Die Bedingung steht in runden Klammern. Ist die Bedingung erfüllt, wird die Anweisungsfolge ausgeführt. Ist die Bedingung nicht erfüllt, wird die Schleife verlassen.

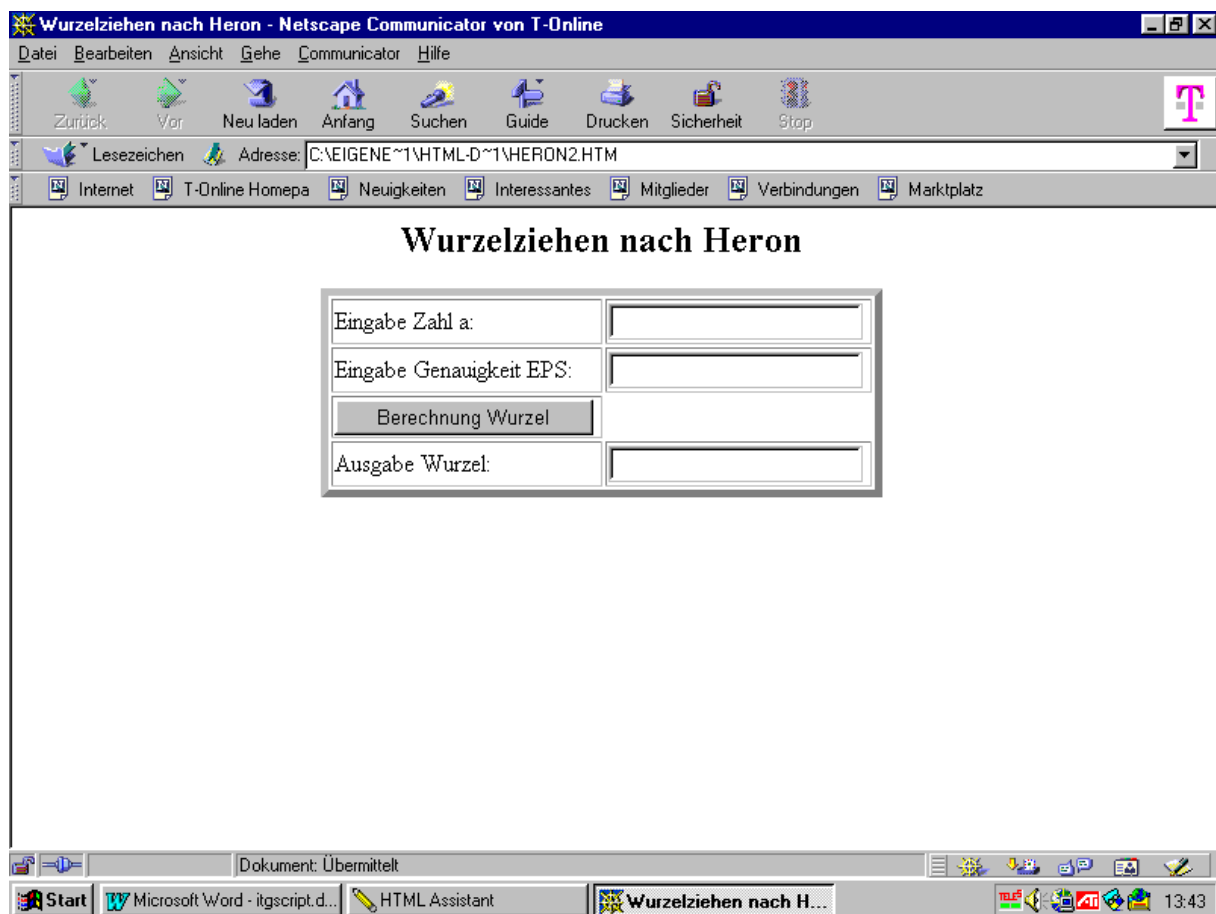
5.1. Wurzelziehen nach Heron $x = \sqrt{a}$

$x_{neu} \leftarrow \frac{1}{2} \left(x_{alt} + \frac{a}{x_{alt}} \right)$ mit Startwert $x_{alt} = a$

5.1.1. Algorithmus

Eingabe Zahl a	
Eingabe Genauigkeit EPS	
$x_{neu} \leftarrow a$	
wiederhole	$x_{alt} \leftarrow x_{neu}$
	$x_{neu} \leftarrow \frac{1}{2} \left(x_{alt} + \frac{a}{x_{alt}} \right)$
bis $ x_{neu} - x_{alt} < EPS$	
Ausgabe xneu	

5.1.2. Programm



```

<HTML>
<HEAD>
<TITLE>Wurzelziehen nach Heron</TITLE>

<SCRIPT language="JavaScript">
function Wurzel(a,EPS,x)
{
    var xneu=parseFloat(a.value);           //Typkonvertierung
    var xalt;
    var genau=parseFloat(EPS.value);       //Typkonvertierung
    do
    {
        xalt=xneu;
        xneu=0.5*(xalt+a.value/xalt);
    }
    while (Math.abs(xneu-xalt)>=genau);
    x.value=xneu;
}
</SCRIPT>

</HEAD>
<BODY>
<CENTER>
<H2>Wurzelziehen nach Heron</H2>
<FORM>
<TABLE BORDER="5">
    <TR>
        <TD> Eingabe Zahl a: </TD>
        <TD><INPUT TYPE="Text" NAME="a" VALUE="" size="20"> </TD>
    </TR>
    <TR>
        <TD> Eingabe Genauigkeit EPS: </TD>
        <TD><INPUT TYPE="Text" NAME="EPS" VALUE="" size="20"></TD>
    </TR>
    <TR>
        <TD><INPUT TYPE="Button" VALUE="Berechnung Wurzel"
            onClick="Wurzel(a,EPS,x)"> </TD>
    </TR>
    <TR>
        <TD> Ausgabe Wurzel: </TD>
        <TD><INPUT TYPE="Text" NAME="x" VALUE="" size="20" ></TD>
    </TR>
</TABLE>
</FORM>
</CENTER>
</BODY>
</HTML>

```

6. Top-Level-Funktionen

6.1. Typkonvertierung

Die Typkonvertierung in JavaScript erfolgt in der ersten Version automatisch. Ab der Version 1.2 muss gegebenenfalls die Konvertierung programmiert werden.

Mit dem typeof-Operator kann der Typ des Operanden festgestellt werden.

Syntax: typeof Operand oder typeof(Operand) Klammern optional

Die Ausgabe Operandentyp kann wie folgt erfolgen: document.write(typeof(Operand))

Zur Typkonvertierung stehen folgende Funktionen zur Verfügung:

eval(String) Wertet den String als numerischen Ausdruck und berechnet den Wert.

parseInt(String [,Basis]) Interpretiert den String als ganze Zahl und liefert diesen Zahlenwert zurück. Die Basis ist optional, für Basen größer als 10 werden die Buchstaben A .. Z benutzt

parseFloat(String) Interpretiert den String als Gleitkommazahl und liefert diesen Zahlenwert zurück. Das erste Zeichen im String muss, wie bei parseInt, eine gültige Ziffer sein, um Fehlermeldungen auszuschließen.

Number(String) Wandelt das Objekt in eine Zahl um.

String(Objekt) Wandelt das Objekt in eine Zeichenkette um.

6.2. Ein- und Ausgabeanweisungen

Die Ausgabe von Texten in einem HTML-Dokument ist mit dem document-Objekt und den Methoden write() und writeln() möglich.

document.write(ausdruck1 [,ausdruck2] ...)

document.writeln(ausdruck1 [,ausdruck2] ...)

document.write schreibt einen oder mehrere Ausdrücke in ein zuvor geöffnetes Dokument. Strings werden in normaler Form, Zahlen dezimal und boolesche Werte in den Zuständen true und false ausgegeben. document.writeln erzeugt zusätzlich einen Zeilenvorschub.

Die Eingabe von Werten in ein HTML-Dokument ist mit dem window-Objekt und der Methode prompt() möglich.

window.prompt(Ausgabertext [,Vorgabe])

Im Ausgabertext sollte die Eingabeaufforderung formuliert werden. Die optionale Vorgabe ist ebenfalls ein String. Fehlt dieser, erscheint undefined im Eingabefeld. Soll kein Wert vorselektiert werden, notiert man einfach zwei Anführungszeichen hintereinander. Damit wird die Ausgabe undefined unterdrückt.