


Der Autor hat für interessierte Leser ein vollständiges Lehrbuch zum Programmieren mit JavaScript (***jspau.pdf***) verfasst. Das Lehrbuch hat 436 Seiten und ist in 8 Teile gegliedert. In jedem Teil werden zu allen wichtigen Themen zuerst die Theorie erklärt und dann dazu die Programmcodes entsprechender Übungsprojekte aufgelistet.

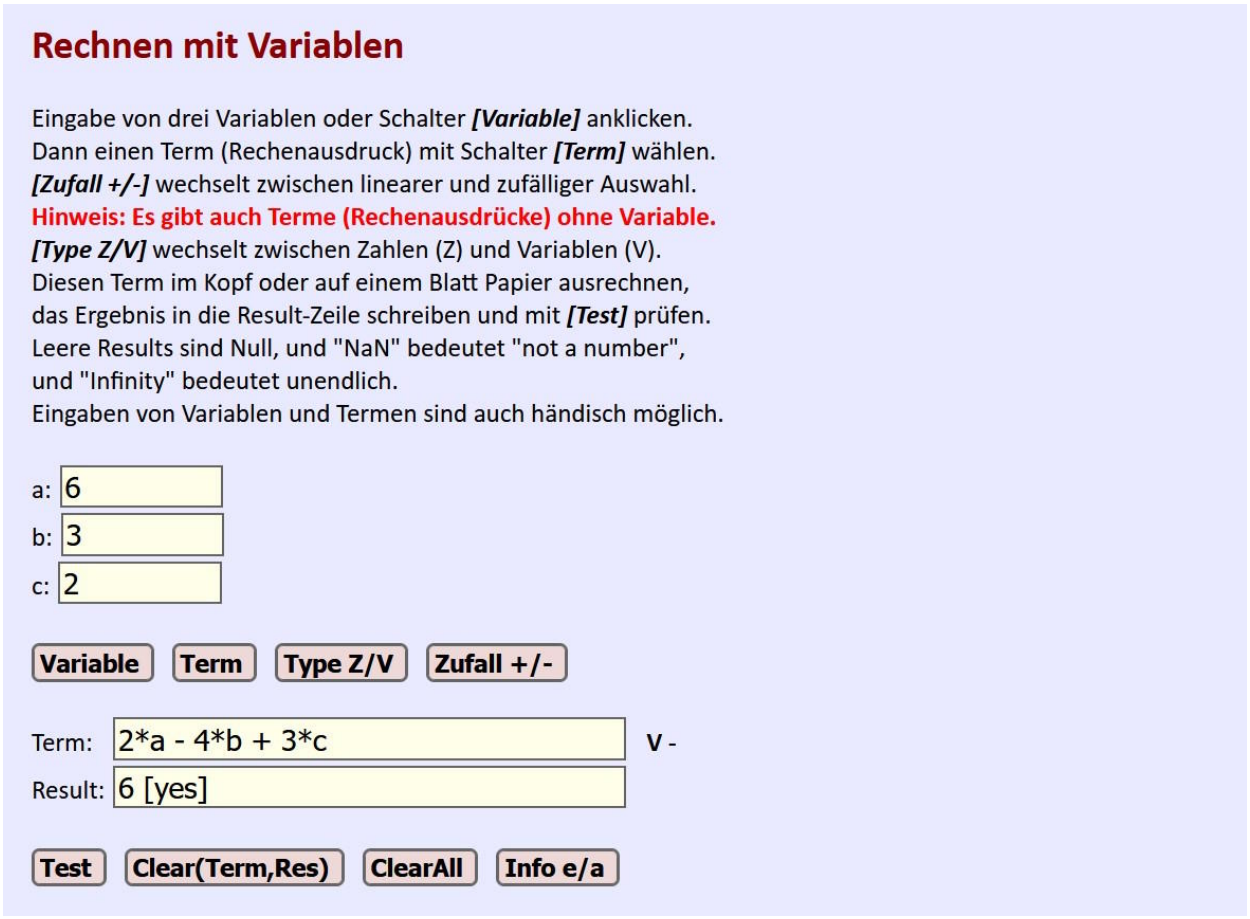
Die hier vorliegenden Seiten stammen aus dem Lehrbuch (***jspau.pdf***). Sie beinhalten erstens sieben einfache Lernprogramme zur Schulmathematik und zweitens eine nützliche Bibliothek (***tools.js***) von Javascript-Funktionen, die in den Programmen verwendet werden.

(1) ***variable.html***, Rechnen mit Zahlen und Variablen Seite 2
(2) ***trigon1.html***, Dreiecke im Koordinatensystem Seite 6
(3) ***trigon2.html***, Trigonometrie des Dreiecks Seite 10
(4) ***funplot.html***, Plotter für Mathematik-Funktionen Seite 14
(5) ***koerper.html***, Würfel und Pyramide im Raum Seite 17
(6) ***kudi.html***, Differenzialrechnung – Kurvendiskussion Seite 21
(7) ***integral.html***, Integralrechnung – Anwendungen Seite 28

Die Bibliothek (***tools.js***) besteht aus drei Teilen (Grafik, Mathematik, Parser) und liegt hier in komprimierter Form vor. In den Lernprogrammen ist die Bibliothek mittels ***<script src = "tools.js"></script>*** eingebunden.

Nach den Listings der sieben Programme folgt eine ausführliche Beschreibung von allen Funktionen der Bibliothek (ab Seite 38).

- **Das Programm „variable.html“**




```
<!DOCTYPE html>
<html lang = "de">
<head>
<title>variable.html</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="author" content="Paukert Herbert">
<meta name="description" content="Variable, Mathematikparser">

<style type = "text/css">
body {background-color:#EAEAFF; color:black; font-family:Calibri,Arial;
      font-size:20px; font-weight:normal; text-align:left; margin:2%; padding-left:2%;}
.cd1 {border: 2px solid #666666; background-color: #FFFFE8; font-size: 22px; margin:2px;}
.cd2 {border: 3px solid #666666; background-color: #EED8D8; border-radius: 6px;
      font-size: 18px; font-weight: bold; margin-top: 3px;}
#inf {visibility: hidden; font-size: 18px;}
.colred {color:red;}
h2 {color: darkred;}
</style>

<script src = 'tools.js'></script>

<script>

document.onkeydown = function(ev) {
// Sperrt Funktionstasten
   if (ev.ctrlKey) {return false; }          // Ctrl ...
   let taste = ev.keyCode || ev.which;
   if (taste === 123) { return false; }      // F12 ...
}

function smart() {
// Anpassung an Smartphones
   let smallWidth = 800;
   let scrWidth = window.screen.width;
   let smallScr = false;
   if (scrWidth <= smallWidth) { smallScr = true; }
```

```
   if (smallScr) {
       document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=0.65');
       liste1 = document.getElementsByClassName("cd1");
       for(var i = 0; i < liste1.length; i++) {
          liste1[i].style.margin = "2px";
          liste1[i].style.fontSize = "22px";
       }
       liste2 = document.getElementsByClassName("cd2");
       for(var i = 0; i < liste2.length; i++) {
          liste2[i].style.margin = "3px";
          liste2[i].style.fontSize = "18px";
       }
       document.getElementById("inf").style.fontSize = 18 + "px";
       document.getElementById("body").style.paddingTop = 10 + "px";
       document.getElementById("body").style.paddingLeft = 10 + "px";
       document.getElementById("body").style.fontSize = 20 + "px";
       document.getElementById("body").style.margin = 10 + "px";
       // alert('small');
   }
   else { document.addEventListener("contextmenu", (e) => {e.preventDefault()}); }
   // Damit wird das Kontextmenü gesperrt.
}

var tzufa  = false;  // Steuervariable für lineare oder zufällige Term-Auswahl
var tzahl = true;    // Steuervariable für Terme mit Zahlen oder mit Variablen
var n = 9            // Anzahl der wählbaren Terme - 1
var num = 0;         // Nummer eines Terms
var term = '';       // Wert eines Zufallterms
var hide = true;     // Steuervariable für die Information

var dat0 = new Array(n);
// 10 Terme mit Zahlen
dat0[0] = '3 * 7 - 1';
dat0[1] = '3 * (7 - 1)';
dat0[2] = '(4 + 5) * 3';
dat0[3] = '28 - 5 * 4';
dat0[4] = '(5 - 3) * (7 - 4)';
dat0[5] = '(-6) * 7';
dat0[6] = '6 * (-7)';
dat0[7] = '(-6) * (-7)';
dat0[8] = '3^2 + 4^2';
dat0[9] = 'sqrt(3^2 + 4^2)';

var dat1 = new Array(n)
// 10 Terme mit Variablen
dat1[0] = '2*a - 4*b + 3*c';
dat1[1] = 'a*(b - c)';
dat1[2] = '(a + b) * (a - b)';
dat1[3] = 'a^2 + 3';
dat1[4] = '3 * sqrt(a)';
dat1[5] = 'sqrt(a^2)';
dat1[6] = '(a + b)^2';
dat1[7] = '(a - b)^2';
dat1[8] = '(a/2)^2 - a';
dat1[9] = '(a/2)^2 - (b/2)^2';

function setValues(vari,value) {
// Variable setzen
  if (vari == 'a') { a = value; }
  if (vari == 'b') { b = value; }
  if (vari == 'c') { c = value; }
}

function clearVar() {
// Variable löschen
  a = 0; b = 0; c = 0;
}

function Calcul() {
// Rechnung ausführen
  let s,t,x,y,z;
  a = 1 * document.MyForm.ein1.value.trim();
  b = 1 * document.MyForm.ein2.value.trim();
  c = 1 * document.MyForm.ein3.value.trim();
  setValues('a',a);
  setValues('b',b);
  setValues('c',c);
  z = document.MyForm.aus.value;
  x = z.includes('yes');
  if (x) { return; }
  x = z.includes('no');
  if (x) { return; }
```

```
  s = document.MyForm.ein.value.trim();
  if (s == '') {document.MyForm.ein.value = '0'; }
  y = formel(s);
  t = 1 * document.MyForm.aus.value.trim();
  if (y != t) { document.MyForm.aus.value = t + ' [no] => ' + y; }
  else { document.MyForm.aus.value = y + ' [yes]'; }
}

function Clear() {
// Eingabezeile und Ausgabezeile löschen
  document.MyForm.ein.value = '';
  document.MyForm.aus.value = '';
}

function ClearAll(){
// Alles löschen
  document.MyForm.ein.value = '';
  document.MyForm.aus.value = '';
  document.MyForm.ein1.value = '';
  document.MyForm.ein2.value = '';
  document.MyForm.ein3.value = '';
}

function Info(){
// Information ein/aus
  hide = !hide;
  if (hide == true) { document.getElementById("inf").style.visibility = 'hidden'; }
  else { document.getElementById("inf").style.visibility = 'visible'; }
}

function Varia(){
// Variable definieren
  if (tzahl) { alert('Abbruch! Die Terme enthalten nur Zahlen.'); return; }
  let x = Math.floor(10 * Math.random());
  let y = Math.floor(10 * Math.random());
  let z = Math.floor(10 * Math.random());
  a = x.toString();
  b = y.toString();
  c = z.toString();
  document.MyForm.ein1.value = a;
  document.MyForm.ein2.value = b;
  document.MyForm.ein3.value = c;
}

function Zahl() {
// Wechsel zwischen Termen mit Zahlen  oder mit Variablen
   el = document.getElementById("typ");
   ClearAll();
   num = -1;
   tzahl = !tzahl;
   if (tzahl)  { el.innerHTML = 'Z'; alert('Terme mit Zahlen (Z)'); }
   if (!tzahl) { el.innerHTML = 'V'; Varia(); alert('Terme mit Variablen (V)'); }
}

function Zufall() {
// Wechsel zwischen linearer und zufälliger Term-Auswahl
   el = document.getElementById("zuf");
   ClearAll();
   num = -1;
   tzufa = !tzufa;
   if (tzufa)  { el.innerHTML = '+'; alert('Zufällige Term-Auswahl (+)'); }
   if (!tzufa) { el.innerHTML = '-'; alert('Lineare Term-Auswahl (-)'); }
   if (!tzahl) { Varia(); }
}

function Auswahl() {
// Auswahl eines Terms
   Clear();
   if (tzufa)   { num = Math.floor(n * Math.random() + 1); }
   if (!tzufa)  { num = num + 1; if (num > n) {num = 0; } }
   if (tzahl)   { term = dat0[num]; }
   if (!tzahl)  { term = dat1[num]; }
   document.MyForm.ein.value = term;
   document.MyForm.aus.focus();
}

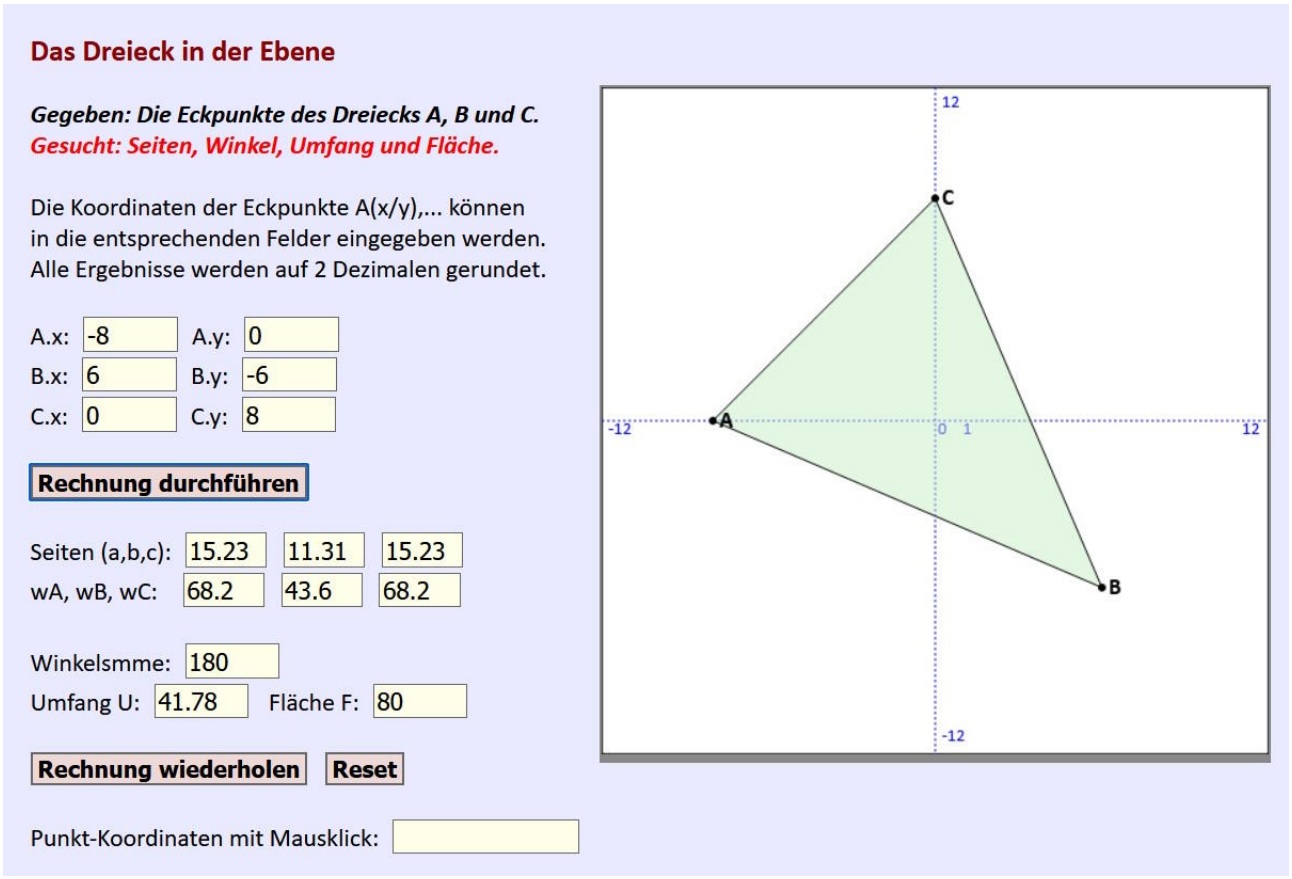
</script>
</head>
```

```
<body id="body">
<form name="MyForm">
<h2>Rechnen mit Variablen</h2>
Eingabe von drei Variablen oder Schalter <b><i>[Variable]</i></b> anklicken.<br>
Dann einen Term (Rechenausdruck) mit Schalter <b><i>[Term]</i></b> wählen.<br>
<b><i>[Zufall +/-]</i></b> wechselt zwischen linearer und zufälliger Auswahl.<br>
<b><span class="colred">Hinweis: Es gibt auch Terme (Rechenausdrücke) ohne Variable.</span></b><br>
<b><i>[Type Z/V]</i></b> wechselt zwischen Zahlen (Z) und Variablen (V).<br>
Diesen Term im Kopf oder auf einem Blatt Papier ausrechnen,<br>
das Ergebnis in die Result-Zeile schreiben und mit <b><i>[Test]</i></b> prüfen.<br>
Leere Results sind Null, und "NaN" bedeutet "not a number",<br>
und "Infinity" bedeutet unendlich.<br>
Eingaben von Variablen und Termen sind auch händisch möglich.<br>
<br>
a: <input type="text" name="ein1" value="" size="8" class="cd1"><br>
b: <input type="text" name="ein2" value="" size="8" class="cd1"><br>
c: <input type="text" name="ein3" value="" size="8" class="cd1"><br>
<br>
<input type="button" name="btn1" value="Variable " class="cd2" onclick="Varia()">  
<input type="button" name="btn2" value="Term " class="cd2" onclick="Auswahl()">  
<input type="button" name="btn3" value="Type Z/V " class="cd2" onclick="Zahl()">  
<input type="button" name="btn4" value="Zufall +/- " class="cd2" onclick="Zufall()">
<br><br>
Term:   <input type="text" name="ein" value="" size="30" class="cd1">  
<b><span id="typ">Z</span></b> <b><span id="zuf">-</span></b>
<br>
Result: <input type="text" name="aus" value=" " size="30" class="cd1"><br>
<br>
<input type="button" name="btn" value="Test " class="cd2" onclick="Calcul()">  
<input type="button" name="btn" value="Clear(Term,Res) " class="cd2" onclick="Clear()">  
<input type="button" name="btn" value="ClearAll " class="cd2" onclick="ClearAll()">  
<input type="button" name="btn" value="Info e/a " class="cd2" onclick="Info()"><br>
<br>
<div id='inf'>
------------------------------------------------------------<br>
Der universelle Mathematik-Parser erlaubt die<br>
Eingabe von 26 numerischen Variablen (a,b,...z)<br>
und einem String s als Term. Dieser wird mt<br>
"[Test]" ausgewertet und das entsprechende<br>
Resultat wird angezeigt. Im Programm hier<br>
werden nur die 3 Variablen a, b, c verwendet.<br>
Alle Variable müssen klein geschrieben werden,<br>
ausgenommen die zwei Konstanten PI und E. <br>
<br>
<span class="colred">Folgende Operatoren stehen zur Verfügung:
</span>
<br>
(, ), +, -, *, /, ^, sqrt, % = Rest, fac = Faktorielle,<br>
round, rd2, rd4, floor, abs, exp, log, lg, deg, rad, sin,<br>
cos, tan, asin, acos, atan, atan2(y,x) = Steigungs-<br>
winkel, random(x) = Zufallszahlen in [0,x), PI, E.<br>
("Infinity" = Unendlich, "NaN" = Not a Number).<br>
<br>
<span class="colred">Beispiele:</span>
<br>
a % b, sin(rad(x)), deg(asin(x)), deg(atan2(a,b)),<br>
sqrt(x), log(x), exp(x). Grundsätzlich werden<br>
Winkelfunktionen in Radianten berechnet.<br>
"deg(x)" wandelt Radianten in Grade um.<br>
"rad(x)" wandelt Grade in Radianten um.<br>
"rd2" bedeutet "auf zwei Dezimalen runden".<br>
<span class="colred">Beispiel 1:</span>
<br>
Eingabe der Variablen a = 1 und b = 3. <br>
Formeleingabe "rd2(sqrt(a^2 + b^2))" und "[Test]".<br>
<span class="colred">Beispiel 2:</span>
<br>
Eingabe der Variablen a = 30. <br>
Formeleingabe "rd2(sin(rad(a)))" und "[Test]".<br>
------------------------------------------------------------<br>
&copy; Herbert Paukert
<br>
</div>
</form>

<script>
   smart();
</script>

</body>
</html>
```

• Das Programm „trigon1.html“




```
<!DOCTYPE html>
<html lang = "de">
<head>
<title>trigon1.html</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="author" content="Paukert Herbert">
<meta name="description" content="Dreiecke im Koordinatensystem">

<style>
body {background-color:#EAEAFF; color:black; font-family:Calibri,Arial; font-size:19px; font-
weight:normal;
      text-align:left; padding-left:3%; padding-top:2%; }
.cd1 {border: 1px solid #666666; background-color: #FFFFE8; font-size: 18px; margin:2px;}
.cd2 {border: 2px solid #666666; background-color: #EED8D8; font-size: 17px; font-weight: bold;}
#dc {position:absolute; top:100px; left:550px; background-color:#888888; border: 2px solid #888888;}
.colred   {color:red;  font-weight: bold; font-style: italic; }
.colblack {color:black;}
.fsize16 { font-size:16px;}
h3 {color: darkred; }
hr {width:80%; margin-left:0; border: 2px solid #666666; }

</style>

<script src="tools.js"></script>

<script>

neu = false;

document.onkeydown = function(ev) {
// Sperrt Funktionstasten
   if (ev.ctrlKey) {return false; }         // Ctrl ...
   let taste = ev.keyCode || ev.which;
   if (taste === 123) { return false; }     // F12 ...
}
```

```
function smart() {
// Anpassung an Smartphones
   let smallWidth = 800;
   let smallScr = false;
   let scrWidth = screen.width;
   if (scrWidth <= smallWidth) { smallScr = true; }
   if (smallScr) {
      document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=0.70');
      let base = document.getElementById("body");
      let click = document.getElementById("click");
      base.style.paddingLeft = 10 + 'px';
      base.style.paddingTop = 10 + 'px';
      let end = click.offsetTop;
      document.getElementById("dc").style.top = end + 40 + "px";
      document.getElementById("dc").style.left = 20 + "px";
   }
   else { document.addEventListener("contextmenu", (e) => {e.preventDefault()}); }
}

function showPos(ev) {
// Anzeige von Punktkoordinaten mit Mausklick
   let cRect = document.getElementById("dc").getBoundingClientRect();
   let cTop  = cRect.top;
   let cLeft = cRect.left;
   let x1 = (ev.clientX - cLeft);
   let y1 = (ev.clientY - cTop);
   let x0 = x1*2*kw/kb - kw;
   let y0 = (kb - y1)*2*kw/kb - kw;
   let info = round2(x0) + '  /  ' + round2(y0);
   document.MyForm.aus10.value = info;
}

function Calcul()  {
// Hauptroutine zur Berechnung
let A = new TPoint;
let B = new TPoint;
let C = new TPoint;

document.MyForm.aus9.value = ' ';
A.x = 1 * document.MyForm.ein1.value;
A.y = 1 * document.MyForm.ein2.value;
B.x = 1 * document.MyForm.ein3.value;
B.y = 1 * document.MyForm.ein4.value;
C.x = 1 * document.MyForm.ein5.value;
C.y = 1 * document.MyForm.ein6.value;

let kw1, a, b, c,  wa, wb, wc, ws, s, u, f;

kw1 = Amax10(A.x,A.y,B.x,B.y,C.x,C.y);
kw1 = Math.round(Math.abs(kw1) + 4);
if (kw1 > 12) { kw = kw1; }
else { kw = 12; }

a = Slen(B,C);
b = Slen(A,C);
c = Slen(A,B);
u = Umfang(A,B,C);
f = Flaeche(A,B,C);
wa = Winkel(B,A,C);
wb = Winkel(A,B,C);
wc = Winkel(A,C,B);
ws = round2(wa + wb + wc);
wa = round2(wa);
wb = round2(wb);
wc = round2(wc);

u = round2(u);
f = round2(f);
a = round2(a);
b = round2(b);
c = round2(c);

initCanvas(kb,2,1);
drawColor("#000000");
drawWidth(1);
fillColor("rgba(200,240,200,0.5)");
drawLine3(A,B,C);
drawPoint("A",A);
drawPoint("B",B);
drawPoint("C",C);
```

```
if (f == 0) {
  document.MyForm.aus1.value = ' ';
  document.MyForm.aus2.value = ' ';
  document.MyForm.aus3.value = ' ';
  document.MyForm.aus4.value = ' ';
  document.MyForm.aus5.value = ' ';
  document.MyForm.aus6.value = ' ';
  document.MyForm.aus7.value = ' ';
  document.MyForm.aus8.value = ' ';
  document.MyForm.rech.focus();
  initCanvas(kb,2,1);
  alert('Die Punkte bilden kein Dreieck !');
  return;
}

document.MyForm.rech.focus();
if (neu == true) { neu = false; return; }
document.MyForm.aus1.value = a;
document.MyForm.aus2.value = b;
document.MyForm.aus3.value = c;
document.MyForm.aus4.value = wa;
document.MyForm.aus5.value = wb;
document.MyForm.aus6.value = wc;
document.MyForm.aus7.value = ws;
document.MyForm.aus8.value = u;
document.MyForm.aus9.value = f;
document.MyForm.rech.focus();
}

function Neu() {
// Programm-Wiederholung mit Zufallsdaten
document.MyForm.reset();
z = Math.random(); if (z < 0.5) { v = -1;} else {v = 1; }
z = 1 + 9 * Math.random();
p1 = v * Math.round(z);
z = Math.random(); if (z < 0.5) { v = -1;} else {v = 1; }
z = 1 + 9 * Math.random();
p2 = v * Math.round(z);
z = Math.random(); if (z < 0.5) { v = -1;} else {v = 1; }
z = 1 + 9 * Math.random();
p3 = v * Math.round(z);
z = Math.random(); if (z < 0.5) { v = -1;} else {v = 1; }
z = 1 + 9 * Math.random();
p4 = v * Math.round(z);
z = Math.random(); if (z < 0.5) { v = -1;} else {v = 1; }
z = 1 + 9 * Math.random();
p5 = v * Math.round(z);
z = Math.random(); if (z < 0.5) { v = -1;} else {v = 1; }
z = 1 + 9 * Math.random();
p6 = v * Math.round(z);
document.MyForm.ein1.value = p1;
document.MyForm.ein2.value = p2;
document.MyForm.ein3.value = p3;
document.MyForm.ein4.value = p4;
document.MyForm.ein5.value = p5;
document.MyForm.ein6.value = p6;
document.MyForm.aus1.value = '';
document.MyForm.aus1.value = ' ';
document.MyForm.aus2.value = ' ';
document.MyForm.aus3.value = ' ';
document.MyForm.aus4.value = ' ';
document.MyForm.aus5.value = ' ';
document.MyForm.aus6.value = ' ';
document.MyForm.aus7.value = ' ';
document.MyForm.aus8.value = ' ';
document.MyForm.aus9.value = ' ';
document.MyForm.aus10.value = ' ';
document.MyForm.rech.focus();
neu = true;
kw = 12;
Calcul();
}

function Reset() {
// Formular-Reset
  document.MyForm.reset();
  initCanvas(kb,2,1);
  neu = true;
  Calcul();
}

</script>
</head>
```

```
<body id="body" onload = "initCanvas(kb,2,1);")>

<div id='dc'>
  <canvas id='MyCanvas' width='500' height='500' onmousedown = "showPos(event)">
     Your browser does not support HTML5 Canvas.
  </canvas>
</div>

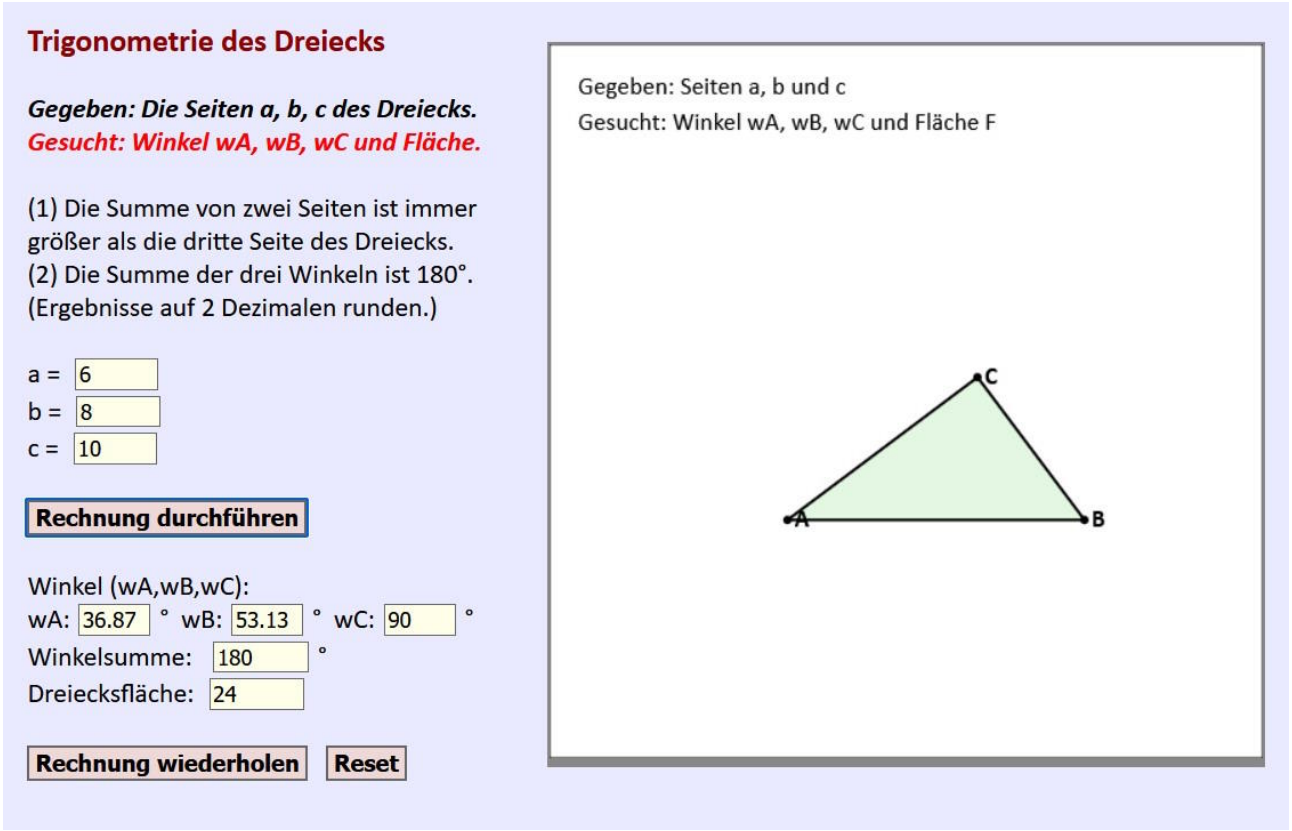
<form name="MyForm">
<h3>Das Dreieck in der Ebene</h3>
<b><i>
Gegeben: Die Eckpunkte des Dreiecks A, B und C.<br>
</i></b>
<span class = "colred">
Gesucht: Seiten, Winkel, Umfang und Fläche.<br>
</span>
<br>
Die Koordinaten der Eckpunkte A(x/y),... können<br>
in die entsprechenden Felder eingegeben werden.<br>
Alle Ergebnisse werden auf 2 Dezimalen gerundet.<br>
<br>
A.x:  <input type="text" name="ein1" value="-8" size="5" class="cd1"> 
A.y:  <input type="text" name="ein2" value="0" size="5" class="cd1">
<br>
B.x:  <input type="text" name="ein3" value="6" size="5" class="cd1"> 
B.y:  <input type="text" name="ein4" value="-6" size="5" class="cd1">
<br>
C.x:  <input type="text" name="ein5" value="0" size="5" class="cd1"> 
C.y:  <input type="text" name="ein6" value="8" size="5" class="cd1">
<br><br>
<input type="button" name="rech" value="Rechnung durchführen" class="cd2" onclick="Calcul()">
<br><br>
Seiten (a,b,c): 
<input type=text" name="aus1" value=" " size="4" class="cd1"> 
<input type=text" name="aus2" value=" " size="4" class="cd1"> 
<input type=text" name="aus3" value=" " size="4" class="cd1"><br>
wA, wB, wC:    <input type=text" name="aus4" value=" " size="4" class="cd1"> 
<input type=text" name="aus5" value=" " size="4" class="cd1"> 
<input type=text" name="aus6" value=" " size="4" class="cd1"><br>
<br>
Winkelsmme:  <input type=text" name="aus7" value=" " size="5" class="cd1"><br>
Umfang U:  <input type=text" name="aus8" value=" " size="5" class="cd1">  
Fläche F:  <input type=text" name="aus9" value=" " size="5" class="cd1">
<br><br>
<input type="button" name="knopf" value="Rechnung wiederholen" class="cd2" onclick=Neu()>  
<input type="button" name="rst" value="Reset" class="cd2" onclick="Reset()">
<br><br>
Punkt-Koordinaten mit Mausklick: 
<input type="text" id="click" name="aus10" value=" " size="12" class="cd1">
</form>

<script>
  document.MyForm.rech.focus();
  smart();
</script>

</body>
</html>

-------------------------------------------------------------------------------------------------------------------------
```

- **Das Programm „trigon2.html“**




```
<!DOCTYPE html>
<html lang="de">
<head>
<title>trigon2.html</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="author" content="Paukert Herbert">
<meta name="description" content="Trigonometrie des Dreiecks">

<style type = "text/css">
body {background-color:#EAEAFF; color:black; font-family:Calibri,Arial; font-size:19px;
      font-weight:normal; text-align:left; padding-left:2%; padding-top:1%; }
.cd1 {border: 1px solid #666666; background-color: #FFFFE8; font-size: 15px; margin:2px;}
.cd2 {border: 2px solid #666666; background-color: #EED8D8; font-size: 16px; font-weight: bold;}
#dc {position:absolute; top:60px; left:500px; background-color:#888888; border: 2px solid #888888;}
.colred   {color:red;}
.colblack {color:black;}
.fsize16 { font-size:16px;}
h3 {color: darkred; }
hr {width:80%; margin-left:0; border: 2px solid #666666; }
</style>

<script src="tools.js"></script>

<script>

var neu = false;

document.onkeydown = function(ev) {
// Sperrt Funktionstasten
   if (ev.ctrlKey) {return false; }         // Ctrl ...
   let taste = ev.keyCode || ev.which;
   if (taste === 123) { return false; }     // F12 ...
}

function smart() {
// Anpassung an Smartphones
   let smallWidth = 800;
   let smallScr = false;
   let scrWidth = screen.width;
   if (scrWidth <= smallWidth) { smallScr = true; }
```

```
   if (smallScr) {
      document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=0.80');
      let end = document.body.offsetHeight;
      document.getElementById("dc").style.top = end + 40 + "px";
      document.getElementById("dc").style.left = 10 + "px";
      document.getElementById("dc").style.width = scrWidth + "px";
   }
   else { document.addEventListener("contextmenu", (e) => {e.preventDefault()}); }
}

function Calcul() {
// Haupt-Funktion zur Berechnung
   let A = new TPoint;
   let B = new TPoint;
   let C = new TPoint;

   document.MyForm.aus1.value = '';
   document.MyForm.aus2.value = '';
   document.MyForm.aus3.value = '';
   document.MyForm.aus4.value = '';
   document.MyForm.aus5.value = '';

   let a = 1 * document.MyForm.ein1.value;
   let b = 1 * document.MyForm.ein2.value;
   let c = 1 * document.MyForm.ein3.value;

   let kw1, wa, wb, wc, u, s, d, f, h, x;

   u = a + b + c;
   s = u/2;
   d = s*(s-a)*(s-b)*(s-c);
   if (d <= 0) {
     alert('Die Seiten bilden KEIN Dreieck !');
     initCanvas(kb,2,0);
     return;
   }

   f = Math.sqrt(d);
   h = 2*f/c;
   f = round2(f);
   x = Math.sqrt(b*b - h*h);

   A.x = 0;
   A.y = 0;
   B.x = c;
   B.y = 0;
   C.x = x;
   C.y = h;

   wa = Winkel(B,A,C);
   wb = Winkel(A,B,C);
   wc = Winkel(A,C,B);
   wsum = wa + wb + wc;
   wsum = round2(wsum);
   wa = round2(wa);
   wb = round2(wb);
   wc = round2(wc);

   kw1 = Amax10(A.x,A.y,B.x,B.y,C.x,C.y);
   kw1 = Math.round(Math.abs(kw1) + 2);
   if (kw1 > 12) { kw = kw1; }
   else { kw = 12; }

   let v = kw/3;
   A.x = 0-v;
   A.y = 0-v;
   B.x = c-v;
   B.y = 0-v;
   C.x = x-v;
   C.y = h-v;

   initCanvas(kb,2,0);
   drawColor("#000000");
   drawWidth(2);
   fillColor("rgba(200,240,200,0.5)");
   drawLine3(A,B,C);
   drawPoint("A",A);
   drawPoint("B",B);
   drawPoint("C",C);
```

```
   drawText(35,"Gegeben: Seiten a, b und c");
   drawText(60,"Gesucht: Winkel wA, wB, wC und Fläche F");
   document.MyForm.rech.focus();

   if (neu == true) { neu = false; return; }

   document.MyForm.aus1.value = wa;
   document.MyForm.aus2.value = wb;
   document.MyForm.aus3.value = wc;
   document.MyForm.aus4.value = wsum;
   document.MyForm.aus5.value = f;
   document.MyForm.rech.focus();
}

function Neu() {
// Routine zur Programm-Wiederholung
   document.MyForm.reset();
   z = 4 + 16 * Math.random();
   p1 = Math.round(z);
   z = 4 + 16 * Math.random();
   p2 = Math.round(z);
   z = 4 + 16 * Math.random();
   p3 = Math.round(z);
   document.MyForm.ein1.value = p1;
   document.MyForm.ein2.value = p2;
   document.MyForm.ein3.value = p3;
   document.MyForm.aus1.value = '';
   document.MyForm.aus2.value = ' ';
   document.MyForm.aus3.value = ' ';
   document.MyForm.aus4.value = ' ';
   document.MyForm.rech.focus();
   neu = true;
   kw = 12;
   Calcul();
}

function Reset() {
// Formular-Reset
  document.MyForm.reset();
  initCanvas(kb,2,1);
  neu = true;
  Calcul();
}

</script>
</head>

<body onload = "initCanvas(kb,2,1)">

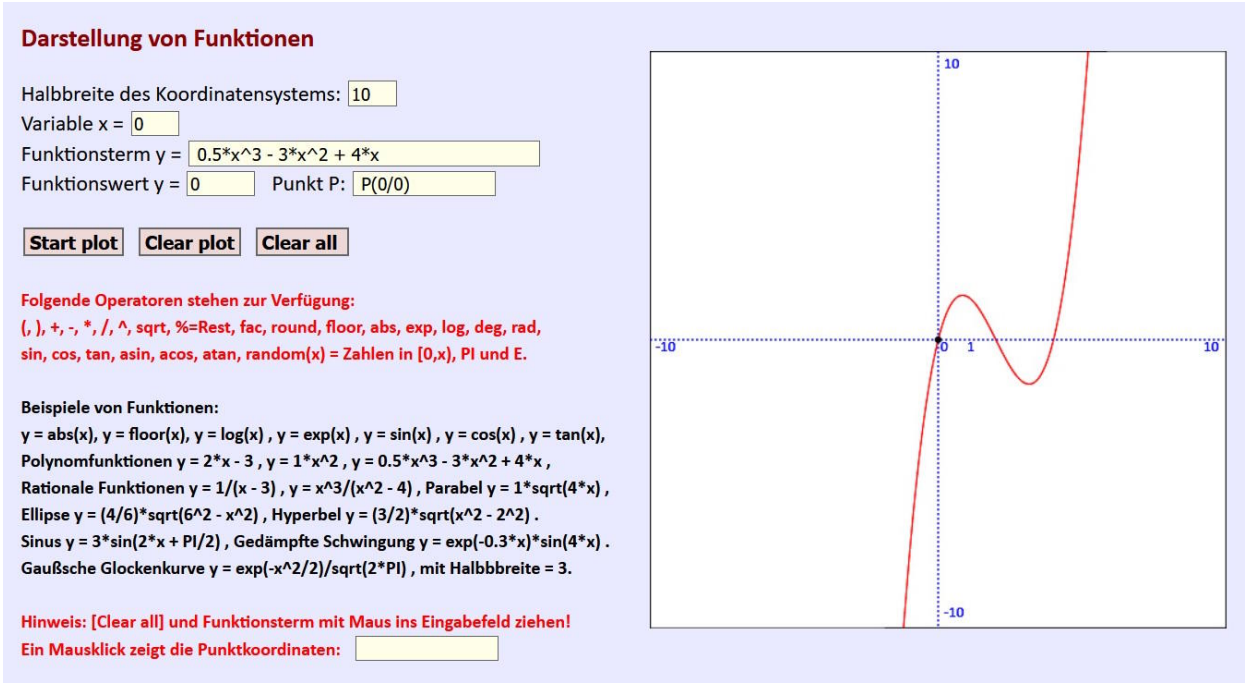
<div id='dc' style="position:absolute; top:60px; left:500px ">
  <canvas id='MyCanvas' width='500' height='500' onmousedown = "showPos(event)">
     Your browser does not support HTML5 Canvas.
  </canvas>
</div>

<form name="MyForm">
<h3>Trigonometrie des Dreiecks</h3>
<b><i>
<span class="colblack">
Gegeben: Die Seiten a, b, c des Dreiecks.<br>
</span>
<span class="colred">
Gesucht: Winkel wA, wB, wC und Fläche.<br>
</span>
</i></b>
<br>
(1) Die Summe von zwei Seiten ist immer<br>
größer als die dritte Seite des Dreiecks.<br>
(2) Die Summe der drei Winkeln ist 180°.<br>
(Ergebnisse auf 2 Dezimalen runden.)<br>
<br>
a =  <input type="text" name="ein1" value="6" size="5" class="cd1"><br>
b =  <input type="text" name="ein2" value="8" size="5" class="cd1"><br>
c =  <input type="text" name="ein3" value="10" size="5" class="cd1"><br>
<br>
<input type="button" name="rech" value="Rechnung durchführen" class="cd2" onclick="Calcul()">
<br>
```

```
<br>
Winkel (wA,wB,wC):<br>
wA: <input type=text" name="aus1" value=" " size="4" class="cd1"> ° 
wB: <input type=text" name="aus2" value=" " size="4" class="cd1"> ° 
wC: <input type=text" name="aus3" value=" " size="4" class="cd1"> °<br>
Winkelsumme:   <input type=text" name="aus4" value=" " size="6" class="cd1"> °
<br>
Dreiecksfläche:  <input type=text" name="aus5" value=" " size="6" class="cd1"><br>
<br>
<input type="button" name="knopf" value="Rechnung wiederholen" class="cd2" onclick=Neu()>  
<input type="button" name="rst" value="Reset" class="cd2" onclick="Reset()">
<br>
</form>
<script>
   document.MyForm.rech.focus();
   smart();
</script>
</body>
</html>

-----------------------------------------------------------------------------------------------------------------------------
```

• Das Programm „fplot.html“




```
<!DOCTYPE html>
<html lang="de">
<head>
<title>funplot.html</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="author" content="Paukert Herbert">
<meta name="description" content="Funktionsplotter">

<style type = "text/css">
body {background-color:#EAEAFF; color:black; font-family:Calibri,Arial; font-size:19px;
      font-weight:normal; text-align:left; padding-left:2%; padding-top:1%; }
.cd1 {border: 1px solid #666666; background-color: #FFFFE8; font-size: 15px; margin: 2px;}
.cd2 {border: 2px solid #666666; background-color: #EED8D8; font-size: 16px; font-weight: bold;
      margin: 2px; }
#dc {position:absolute; top:70px; left:650px; background-color:#888888; border: 2px solid #888888;}
.colred   {color:red;}
.colblack {color:black;}
.fsize16 { font-size:16px;}
h3 {color: darkred; }
hr {width:80%; margin-left:0; border: 2px solid #666666; }
</style>

<script src="tools.js"></script>

<script>

document.onkeydown = function(ev) {
// Sperrt Funktionstasten
   if (ev.ctrlKey) {return false; }         // Ctrl ...
   let taste = ev.keyCode || ev.which;
   if (taste === 123) { return false; }     // F12 ...
}

function smart() {
// Anpassung an Smartphones
   let smallWidth = 800;
   let smallScr = false;
   let scrWidth = screen.width;
   if (scrWidth <= smallWidth) { smallScr = true; }
   if (smallScr) {
      document.querySelector('meta[name="viewport"]').setAttribute('content', 'initial-scale=0.80');
      let end = document.body.offsetHeight;
      document.getElementById("dc").style.top = end + 40 + "px";
      document.getElementById("dc").style.left = 10 + "px";
      document.getElementById("dc").style.width = scrWidth + "px";
   }
   else { document.addEventListener("contextmenu", (e) => {e.preventDefault()}); }
}
```

```
function showPos(ev) {
// Anzeige von Punktkoordinaten mit Mausklick
  let cRect = document.getElementById("dc").getBoundingClientRect();
  let cTop  = cRect.top;
  let cLeft = cRect.left;
  let x1 = (ev.clientX - cLeft);
  let y1 = (ev.clientY - cTop);
  let x0 = x1*2*kw/kb - kw;
  let y0 = (kb - y1)*2*kw/kb - kw;
  let info = round2(x0) + '  /  ' + round2(y0);
  document.MyForm.aus4.value = info;
}

O = new TPoint(0,0);
A = new TPoint(0,0);

function Clear1() {
  // Grafik löschen
  document.MyForm.aus0.value = ' ';
  document.MyForm.aus1.value = ' ';
  document.MyForm.aus4.value = ' ';
  kw = 1 * document.MyForm.ein0.value;
  if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) { kw = 10; document.MyForm.ein0.value = kw; };
  initCanvas(kb,2,1);
}

function Clear2() {
  // Grafik und Eingabezeile löschen
  document.MyForm.ein1.value = 0;
  document.MyForm.ein2.value = ' ';
  document.MyForm.aus0.value = ' ';
  document.MyForm.aus1.value = ' ';
  document.MyForm.aus4.value = ' ';
  kw = 1 * document.MyForm.ein0.value;
  if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) { kw = 10; document.MyForm.ein0.value = kw; };
  initCanvas(kb,2,1);
}

function Calcul() {
  // Hauptroutine
  kw = 1 * document.MyForm.ein0.value;
  if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) { kw = 10; document.MyForm.ein0.value = kw; };
  document.MyForm.aus0.value = ' ';
  document.MyForm.aus1.value = ' ';
  document.MyForm.aus4.value = ' ';
  let x = 1 * document.MyForm.ein1.value;
  let s = document.MyForm.ein2.value;
  s = myTrim(s);
  let y = fun(s,x);
  y = round2(y);
  document.MyForm.aus0.value = y;
  A.x = x;
  A.y = y;
  drawFun(kw,kb,s);
  drawPoint("",A);
  texA = ' P(' + A.x + '/' + A.y + ')';
  document.MyForm.aus1.value = texA;
}

</script>
</head>

<body onload = "initCanvas(kb,2,1)" oncontextmenu="return false;">
<div id='dc' style="position:absolute; top:70px; left:650px ">
  <canvas id='MyCanvas' width='500' height='500' onmousedown = "showPos(event)">
     Your browser does not support HTML5 Canvas.
  </canvas>
</div>
<form name="MyForm">
<h3>Darstellung von Funktionen</h3>
Halbbreite des Koordinatensystems: <input type="text" name="ein0" value="10" size="3"
class="cd1"><br>
Variable x = <input type="text" name="ein1" value="1" size="3" class="cd1"><br>
Funktionsterm y = <input type="text" name="ein2" value="x^2" size="35" class="cd1"><br>
Funktionswert y = <input type="text" name="aus0" value=" " size="5" class="cd1"
readonly="true">  
Punkt P: <input type="text" name="aus1" value=" " size="13" class="cd1" readonly="true"><br>
<br>
<input type="button" name="calc1" value="Start plot" class="cd2" onclick="Calcul()"> 
<input type="button" name="calc2" value="Clear plot" class="cd2" onclick="Clear1()"> 
<input type="button" name="calc2" value="Clear all " class="cd2" onclick="Clear2()"><br>
```

```
<br>
<b>
<span class="fsize16">
<span class="colred">
Folgende Operatoren stehen zur Verfügung:<br>
(, ), +, -, *, /, ^, sqrt, %=Rest, fac, round, floor, abs, exp, log, deg, rad,<br>
sin, cos, tan, asin, acos, atan, random(x) = Zahlen in [0,x), PI und E.<br>
</span>
<br>
<span class="colblack">
Beispiele von Funktionen:<br>
y = abs(x), y = floor(x), y = log(x) , y = exp(x) , y = sin(x) , y = cos(x) , y = tan(x),<br>
Polynomfunktionen y = 2*x - 3 , y = 1*x^2 , y = 0.5*x^3 - 3*x^2 + 4*x ,<br>
Rationale Funktionen y = 1/(x - 3) , y = x^3/(x^2 - 4) , Parabel y = 1*sqrt(4*x) ,<br>
Ellipse y = (4/6)*sqrt(6^2 - x^2) , Hyperbel y = (3/2)*sqrt(x^2 - 2^2) .<br>
Sinus y = 3*sin(2*x + PI/2) , Gedämpfte Schwingung y = exp(-0.3*x)*sin(4*x) .<br>
Gaußsche Glockenkurve y = exp(-x^2/2)/sqrt(2*PI) , mit Halbbbreite = 3.<br>
</span>
<br>
<span class="colred">
Hinweis: [Clear all] und Funktionsterm mit Maus ins Eingabefeld ziehen!<br>
Ein Mausklick zeigt die Punktkoordinaten:  
<input type="text" name="aus4" value=" " size="13" class="cd1" readonly><br>
</span>
</span>
</b>
</form>
<script>
   document.MyForm.ein2.focus();
   smart();
</script>
</body>
</html>
```

- **Das Programm „koerper.html“**

Würfel mit Pyramide

Gegeben: Würfelkante s, die zugleich auch die doppelte Pyramidenhöhe h ist (h = s/2).
Gesucht: Das Volumen V und die Oberfläche O

V = Würfelvolumen VW + Pyramidenvolumen VP.
Oberfläche O = OW + OP - 2*Grundfläche GF (s^2).
VW = s^3, VP = s^2*h/3, V = VW + VP.
OW = 6*s^2, OP = s^2 + 4*(h1*s/2) mit
der Seitenhöhe h1 = sqrt((s/2)² + h²).

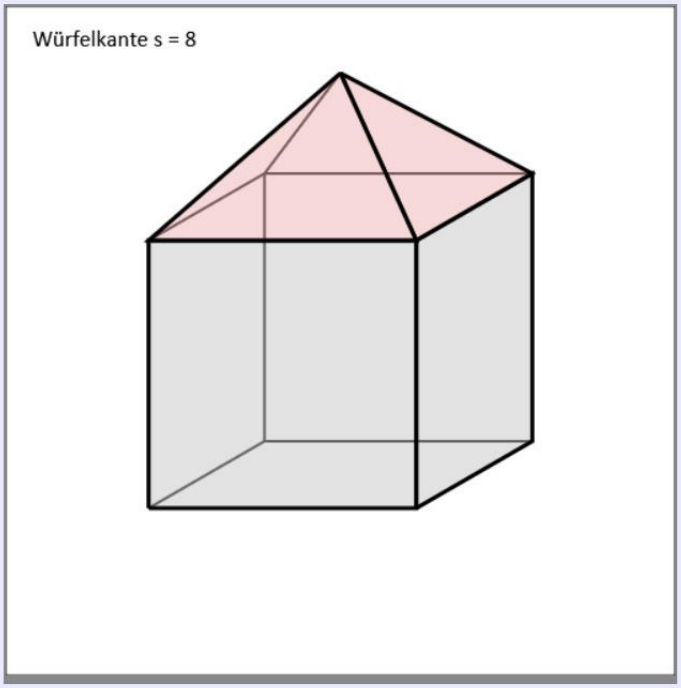
Halbbreite des Koordinatensystems: 10
Würfelkante s = 8

Berechne auf zwei Dezimalstellen gerundet:
V = 597.33
O = 370.6

Rechnung überprüfen

V = richtig
O = richtig

Rechnung wiederholen




```
<!DOCTYPE html>
<html>
<head>
<title>koerper.html</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=0.5">
<meta name="author" content="Paukert Herbert">
<meta name="description" content="Würfel und Pyramide">

<style>
body {background-color:#EAEAFF; color:black; font-family:Calibri,Arial; font-size:18px;
      font-weight:normal; text-align:left; padding-left:2%; padding-top:1%;}
.cd1 {border: 1px solid #666666; background-color: #FFFFE8; font-size: 15px; margin: 2px;}
.cd2 {border: 2px solid #666666; background-color: #EED8D8; font-size: 16px; font-weight: bold;}
#dc  {position:absolute; top:100px; left:550px; background-color:#888888; border:2px solid #888888;}
.colred   {color:red;  font-weight: bold; font-style: italic; }
h2 {color: darkred; }
hr {width:80%; margin-left:0; border: 2px solid #666666; }
</style>

<script src="tools.js"></script>

<script>

var neu = true;

document.onkeydown = function(ev) {
// Sperrt Funktionstasten
   if (ev.ctrlKey) {return false; }          // Ctrl ...
   let taste = ev.keyCode || ev.which;
   if (taste === 123) { return false; }      // F12 ...
}

function smart() {
// Anpassung an Smartphones
   var smallWidth = 800;
   var smallScr = false;
   var scrWidth = window.screen.width;
   var canWidth = scrWidth;
   if (scrWidth <= smallWidth) { smallScr = true; }
```

```
   if (smallScr) {
      document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=0.75');
      let end = document.body.offsetHeight;
      document.getElementById("dc").style.top = end + 40 + "px";
      document.getElementById("dc").style.left = 20 + "px";
   }
   else { document.addEventListener("contextmenu", (e) => {e.preventDefault()}); }
}

function Calcul() {
// Hauptroutine für Berechnung und dreidimensionale Darstellung
  let s,r,GF,VW,VP,V,OW,OP,O,V1,V2,z1,z2;

  initCanvas(kb,3);
  kw = 1 * document.MyForm.ein0.value;
  s = 1 * document.MyForm.ein1.value;
  r = s / 2;

  A3 = new TPoint3;
  B3 = new TPoint3;
  C3 = new TPoint3;
  D3 = new TPoint3;
  E3 = new TPoint3;
  F3 = new TPoint3;
  G3 = new TPoint3;
  H3 = new TPoint3;
  M3 = new TPoint3;
  S3 = new TPoint3;
  R3 = new TPoint3;

  A = new TPoint;
  B = new TPoint;
  C = new TPoint;
  D = new TPoint;
  E = new TPoint;
  F = new TPoint;
  G = new TPoint;
  H = new TPoint;
  M = new TPoint;
  S = new TPoint;
  R = new TPoint;

  A3.x = -r;
  A3.y = -r;
  A3.z = -r;

  B3.x = s-r;
  B3.y = -r;
  B3.z = -r;

  C3.x = s-r;
  C3.y = s-r;
  C3.z = -r;

  D3.x = -r;
  D3.y = s-r;
  D3.z = -r;

  E3.x = -r;
  E3.y = -r;
  E3.z = s-r;

  F3.x = s-r;
  F3.y = -r;
  F3.z = s-r;

  G3.x = s-r;
  G3.y = s-r;
  G3.z = s-r;

  H3.x = -r;
  H3.y = s-r;
  H3.z = s-r;

  M3.x = 0;
  M3.y = 0;
  M3.z = 0;

  S3.x = 0;
  S3.y = 0;
  S3.z = s;
```

```
  M = Schraeg(M3);
  S = Schraeg(S3);

  A = Schraeg(A3);
  B = Schraeg(B3);
  C = Schraeg(C3);
  D = Schraeg(D3);

  E = Schraeg(E3);
  F = Schraeg(F3);
  G = Schraeg(G3);
  H = Schraeg(H3);

  drawWidth(2);
  drawColor("#000000");

  drawLine(A,B);
  drawLine(B,C);
  drawLine(C,D);
  drawLine(D,A);

  drawLine(E,F);
  drawLine(F,G);
  drawLine(G,H);
  drawLine(H,E);

  drawLine(A,E);
  drawLine(B,F);
  drawLine(C,G);
  drawLine(D,H);
/*
  drawPoint("A",A);
  drawPoint("B",B);
  drawPoint("C",C);
  drawPoint("D",D);
  drawPoint("E",E);
  drawPoint("F",F);
  drawPoint("G",G);
  drawPoint("H",H);
  drawPoint("S",S);
*/
  drawLine(S,E);
  drawLine(S,F);
  drawLine(S,G);
  drawLine(S,H);

  drawWidth(3);
  fillColor("rgba(200,200,200,0.5)");
  drawColor("rgb(0,0,0)");
  drawLine4(D,C,G,H);
  drawLine4(C,B,F,G);
  fillColor("rgba(240,180,180,0.5)");
  drawLine3(S,G,F);
  drawLine3(S,G,H);

  r = s / 2;
  info = 'Würfelkante s = ' + s;
  drawText(30,info);

  document.MyForm.rech.focus();
  if (neu == true) { neu = false; return; }

  VW = s*s*s;
  VP = s*s*r/3;
  V = VW + VP;
  V = round2(V);

  OW = 6*s*s;
  GF = s*s;
  h1 = Math.sqrt(2*r^2);
  OP = GF + 2*s*h1;
  O = OW + OP - 2*GF;
  O = round2(O);

  V1 = 1 * myTrim(document.MyForm.aus1.value);
  O1 = 1 * myTrim(document.MyForm.aus2.value);

  if ( V1 == V) { z1 = 'richtig'; } else { z1 = 'falsch (' + V + ')'; };
  document.MyForm.erg1.value = z1;
  if ( O1 == O) { z2 = 'richtig'; } else { z2 = 'falsch (' + O + ')'; };
  document.MyForm.erg2.value = z2;
}
```

```
function Neu()  {
// Programmwiederholung
   let z = 2 + 7 * Math.random();
   document.MyForm.ein1.value = Math.round(z);
   document.MyForm.aus1.value = '';
   document.MyForm.aus2.value = '';
   document.MyForm.erg1.value = '';
   document.MyForm.erg2.value = '';
   neu = true;
   Calcul();
   neu = false;
   document.MyForm.rech.focus();
}

</script>
</head>

<body onload = "Calcul()">
<div id='dc' style="position:absolute; top:100px; left:500px ">
  <canvas id='MyCanvas' width='500' height='500'>
     Your browser does not support HTML5 Canvas.
  </canvas>
</div>

<form name="MyForm">
<h2>Würfel mit Pyramide</h2>
Gegeben: Würfelkante s, die zugleich auch die<br>
doppelte Pyramidenhöhe h ist (h = s/2).<br>
<span class = "colred">
Gesucht: Das Volumen V und die Oberfläche O<br>
</span>
<br>
V = Würfelvolumen VW  + Pyramidenvolumen VP.<br>
Oberfläche O = OW + OP - 2*Grundfläche GF (s²).<br>
VW = s³, VP = s²*h/3, V = VW + VP.<br>
OW = 6*s², OP = s² + 4*(h1*s/2) mit<br>
der Seitenhöhe h1 = sqrt((s/2)² + h²).<br>
<br>
Halbbreite des Koordinatensystems:  <input type="text" name="ein0" value="10" size="3"
class="cd1"><br>
Würfelkante s =  <input type="text" name="ein1" value="8" size="3" class="cd1"><br>
<br>
Berechne auf zwei Dezimalstellen gerundet:<br>
V =  <input type="text" name="aus1" value=" " size="7" class="cd1"><br>
O =  <input type="text" name="aus2" value=" " size="7" class="cd1"><br>
<br>
<input type="button" name="rech" value="Rechnung überprüfen" class="cd2" onclick="Calcul()"> <br>
<br>
V =  <input type="text" name="erg1" value=" " size="15" class="cd1"><br>
O =  <input type="text" name="erg2" value=" " size="15" class="cd1"><br>
<br>
<input type="button" name="rech2" value="Rechnung wiederholen" class="cd2" onclick="Neu()"> <br>
<br>
</form>
<script>
   document.MyForm.rech.focus();
   smart();
</script>
</body>
</html>
```

- **Das Programm „kudi.html“**

DIFFERENZIALRECHNUNG
Funktionen, Ableitungen, Kurvendiskussion

Halbbreite des Koordinatensystems: 10
Variable x = 2
f(x) = 0.5*x^3 - 3*x^2 + 4*x
y = 0 y' = -2 y'' = 0

Start | Clear | Clear all | Reset
Nullstelle | Extremum | Wendepunkt | Tangente

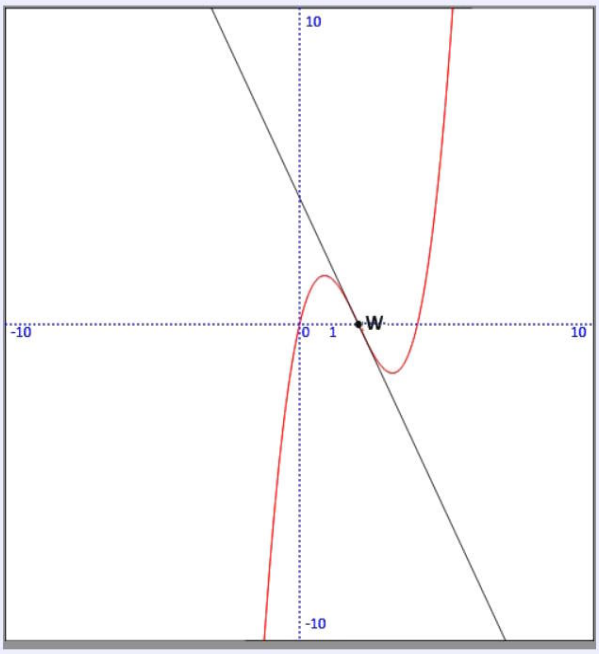
Tangente: y = -2 * x + (4)
Intervall: a = b =
N: E: W: 2/0

Operatoren und Funktionen: (,), +, -, *, /, ^, abs, floor, round, sqr, sqrt, exp, log, deg, rad, sin, cos, tan, asin, acos, atan, PI und E. (Die Winkelfunktionen werden im Bogenmaß berechnet.)

Beispiele: y = log(x) , y = exp(x) , y = sin(x) , y = cos(x) , y = tan(x)
Polynom-Funktionen y = 0.5*x^2 - 3*x - 2 , y = 0.5*x^3 - 3*x^2 + 4*x + 3
Rationale Funktionen y = 1/(x - 4) , y = 0.4*x^2/(x + 2) , y = x^3/(x^2 - 4)
Sinus y = 3*sin(x + PI/2) ,Gedämpfte Schwingung y = exp(-0.3*x)*sin(4*x)
Gaußsche Glockenkurve y = 3*exp(-x^2/2)/sqrt(2*PI) , mit Halbbreite = 3

Hinweis: [Clear all] und Funktion mit Maus ins Eingabefeld ziehen.
Ein Mausklick zeigt die Punktkoordinaten:

Info on/off



DIFFERENZIALRECHNUNG
Funktionen, Ableitungen, Kurvendiskussion

Halbbreite des Koordinatensystems: 10
Variable x = 2
f(x) = 0.5*x^3 - 3*x^2 + 4*x
y = 0 y' = -2 y'' = 0

Start | Clear | Clear all | Reset
Nullstelle | Extremum | Wendepunkt | Tangente

Tangente: y = -2 * x + (4)
Intervall: a = b =
N: E: W:

Operatoren und Funktionen: (,), +, -, *, /, ^, abs, floor, round, sqr, sqrt, exp, log, deg, rad, sin, cos, tan, asin, acos, atan, PI und E. (Die Winkelfunktionen werden im Bogenmaß berechnet.)

Beispiele: y = log(x) , y = exp(x) , y = sin(x) , y = cos(x) , y = tan(x)
Polynom-Funktionen y = 0.5*x^2 - 3*x - 2 , y = 0.5*x^3 - 3*x^2 + 4*x + 3
Rationale Funktionen y = 1/(x - 4) , y = 0.4*x^2/(x + 2) , y = x^3/(x^2 - 4)
Sinus y = 3*sin(x + PI/2) ,Gedämpfte Schwingung y = exp(-0.3*x)*sin(4*x)
Gaußsche Glockenkurve y = 3*exp(-x^2/2)/sqrt(2*PI) , mit Halbbreite = 3

Hinweis: [Clear all] und Funktion mit Maus ins Eingabefeld ziehen.
Ein Mausklick zeigt die Punktkoordinaten:

Info on/off

Auf einer Funktion y = f(x) liegen die zwei benachbarten Punkte P(a/f(a)) und Q(x/f(x)). Die Gerade g durch P und Q nennt man Sekante an die Kurve. Das Dreieck PQR mit dem Punkt R(x/f(a)) heißt Steigungsdreieck und der Winkel bei P heißt Steigungswinkel w. Für die Steigung k gilt: k = tan(w) = (f(x)-f(a))/(x-a).

Nähert sich der Punkt Q kontinuierlich dem Punkt P, dann wird die Sekante g zur einer Tangente t der Kurve, wenn der Punkt Q den Punkt P erreicht hat, und k = tan(w) heißt dann Tangentensteigung oder **Differenzialquotient** im Punkt P(a/f(a). Man schreibt f'(a) = lim(f(x)-f(a))/(x-a) für x → a. Eine etwas andere Schreibweise ist f'(a) = lim(f(a+h)-f(a))/h für h → 0, wobei h = (x-a) ist. Zu f'(a) sagt man »f Strich an der Stelle a«. f'(a) wird auch als erste Ableitung von f(x) bezeichnet. Auf die gleiche Art und Weise kann auch eine zweite Ableitung f''(a) (f Zwei-Strich) als Ableitung der Ableitung gebildet werden, usw... Ableitungen werden hier mit numerischen Näherungen ermittelt.

Die Geradengleichung der Tangente t ist y = k*x + d. Weil nun P(a/f(a) auf t liegt, gilt f(a) = f'(a)*a + d. Es folgt d = f(a) - f'(a)*a und für die Tangentengleichung gilt dann: y = f'(a)*(x - a) + f(a).

Für den Graphen einer Funktion gelten folgende Beschreibungen, die in einer **Kurvendiskussion** erfasst werden (hier ohne Beweise):
(1a) f(x) ist in x = a monoton steigend, wenn dort f'(a) > 0 ist.
(1b) f(x) ist in x = a monoton fallend, wenn dort f'(a) < 0 ist.
(2a) f(x) ist in x = a nach oben hohl gekrümmt, wenn dort f''(a) > 0 ist.
(2b) f(x) ist in x = a nach unten hohl gekrümmt, wenn dort f''(a) < 0 ist.
(3) Eine Funktion f(x) hat bei a einen Scheitelpunkt (Extremwert), wenn sich dort nur die Monotonie ändert, d.h. wenn f'(a) = 0 und f''(a) <> 0. Dann verläuft dort die Tangente parallel zur x-Achse und es gilt: Tiefpunkt mit f'(a) = 0 und f''(a) > 0. Hochpunkt mit f'(a) = 0 und f''(a) < 0.
(4) Eine Funktion f(x) hat bei a einen Wendepunkt, wenn sich dort nur die Krümmung ändert, d.h. wenn f'(a) <> 0 und f''(a) = 0 und f'''(a) <> 0.
(5) Eine Funktion f(x) hat bei a einen Flachpunkt, wenn dort gilt: f'(a) = 0 und f''(a) = 0 und f'''(a) = 0 (z.B. bei a = 0 bei f(x) = x^4/10).

```
<!DOCTYPE html>
<html lang="de">
<head>
<title>kudi.html</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="author" content="Paukert Herbert">
<meta name="description" content="Differenzialrechnung">

<style>
body  {background-color:#E4E4F8; color:black; font-family:Arial;
       font-size:18px; font-weight:normal; text-align:left; margin:2%;}
.cd1  {border: 1px solid #666666; background-color: #FFFFFF; font-size: 16px; margin: 2px;}
.cd2  {border: 2px solid #666666; border-radius: 15%; background-color: #FAEADA;
       font-size: 16px; font-weight: bold; margin: 3px;}
#txt2 {color: black; font-size: 16px; }
#tex  {position:absolute; top:20px; left:650px; color: black; font-size:15px; visibility:hidden;}
#tex1 {color: blue;}
#dc   {position:absolute; top:100px; left:670px; background-color:#888888;
       border:2px solid #888888;}
</style>

<script src="tools.js"> </script>

<head>
<script>
  var s; // Funktionsterm
  var tng = false;
  var run = 0;
  var anz = 0;
  var max = 4;

  A = new TPoint(0,0);
  B = new TPoint(0,0);
  C = new TPoint(0,0);
  D = new TPoint(0,0);
  E = new TPoint(0,0);
  W = new TPoint(0,0);
  O = new TPoint(0,0);
  T = new TPoint(0,0);

  function Clear1() {
  // Grafik löschen
    anz = 0;
    document.MyForm.ein1.value = 0;
    document.MyForm.aus0.value = ' ';
    document.MyForm.aus4.value = ' ';
    document.MyForm.aus5.value = ' ';
    document.MyForm.aus5a.value = ' ';
    document.MyForm.aus6.value = ' ';
    document.MyForm.aus6a.value = ' ';
    document.MyForm.aus6b.value = ' ';
    document.MyForm.aus8.value = ' ';
    document.MyForm.aus9.value = ' ';
    document.MyForm.aus9a.value = ' ';
    kw = 1 * document.MyForm.ein0.value;
    if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) {kw = 10; document.MyForm.ein0.value = kw;};
    initCanvas(kb,2,1);
  }

  function Clear2() {
  // Grafik und Eingabezeile löschen
    anz = 0;
    document.MyForm.ein1.value = 0;
    document.MyForm.ein2.value = ' ';
    document.MyForm.aus0.value = ' ';
    document.MyForm.aus4.value = ' ';
    document.MyForm.aus5.value = ' ';
    document.MyForm.aus5a.value = ' ';
    document.MyForm.aus6.value = ' ';
    document.MyForm.aus6a.value = ' ';
    document.MyForm.aus6b.value = ' ';
    document.MyForm.aus8.value = ' ';
    document.MyForm.aus9.value = ' ';
    document.MyForm.aus9a.value = ' ';
    kw = 1 * document.MyForm.ein0.value;
    if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) {kw = 10; document.MyForm.ein0.value = kw;};
    initCanvas(kb,2,1);
  }
```

```
function Multi() {
// Darstellung von maximal sechs Funktionen
  anz = anz + 1;
  if (anz > max) { anz = 1; }
  kw = 1 * document.MyForm.ein0.value;
  if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) {kw = 10; document.MyForm.ein0.value = kw;};
  if (anz == 1) { initCanvas(kb,2,1); }
  document.MyForm.aus0.value = ' ';
  document.MyForm.aus4.value = ' ';
  document.MyForm.aus5.value = ' ';
  document.MyForm.aus5a.value = ' ';
  document.MyForm.aus6.value = ' ';
  document.MyForm.aus6a.value = ' ';
  document.MyForm.aus6b.value = ' ';
  document.MyForm.aus8.value = ' ';
  document.MyForm.aus9.value = ' ';
  document.MyForm.aus9a.value = ' ';
  x = 1 * document.MyForm.ein1.value;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  t = 'f'+anz+'(x)=' + s;
  y = fun(s,x);
  y = round2(y);
  document.MyForm.aus0.value = y;
  A.x = x;
  A.y = y;
  drawFun(kw,kb,s);
  drawPosText((-kw+1),(kw-1)-(anz*kw/10),t)
}

function Calcul() {
// Hauptroutine
  if (anz > 0) {alert('Bitte [Clear] und [Start] ausführen.'); return;}
  anz = 0;
  kw = 1 * document.MyForm.ein0.value;
  if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) {kw = 10; document.MyForm.ein0.value = kw;};
  initCanvas(kb,2,1);
  document.MyForm.aus0.value = ' ';
  document.MyForm.aus4.value = ' ';
  document.MyForm.aus5.value = ' ';
  document.MyForm.aus5a.value = ' ';
  document.MyForm.aus6.value = ' ';
  document.MyForm.aus6a.value = ' ';
  document.MyForm.aus6b.value = ' ';
  document.MyForm.aus8.value = ' ';
  document.MyForm.aus9.value = ' ';
  document.MyForm.aus9a.value = ' ';

  x = 1 * document.MyForm.ein1.value;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y = fun(s,x);                                  // ==> "tools.js" (parser)
  y = round2(y);
  document.MyForm.aus0.value = y;
  A.x = x;
  A.y = y;
  drawFun(kw,kb,s);                              // ==> "tools.js" (graph)
  if (run < 1) { drawPoint('',A); }

  O.x = 0;
  O.y = 0;
  C.x = 0;
  C.y = 0;
  D.x = 0;
  D.y = 0;

  l = A.len();
  l = round2(l);
  w = A.win();
  w = round2(deg(w));
  // alert('Betrag OA = ' + l + ', Winkel OA = ' + w);

  x = 1 * document.MyForm.ein1.value;
  s = document.MyForm.ein2.value;
  ky = differ(s,x);                              // ==> "tools.js" (mathe)
  kky = differ2(s,x);                            // ==> "tools.js" (mathe)
  document.MyForm.aus5.value = round2(ky);
  document.MyForm.aus5a.value = round2(kky);
```

```
if (tng) {
drawWidth(1);
drawColor("#000000");
T = tangente(s,A);
texT = ' y = ' + T.x + ' * x + (' + T.y + ')';
if (Number.isNaN(T.y)) { texT = ' x = ' + T.x; }
if (T.y == Infinity) { texT = ' x = ' + T.x; }

document.MyForm.aus6.value = texT;

if ( Number.isNaN(A.y) ) {
  A.y = 0;
  drawPoint("",A);
  document.MyForm.aus6.value = '';
}
else {
if ( (Number.isNaN(T.y)) || (T.y == Infinity) ) {
  B.x = A.x;
  B.y = -0.999;
  //  drawPoint("B",B);
  drawLine1(A,B);
}
else {
  B.x = A.x - 0.999;
  B.y = T.x * B.x + T.y;
  //  drawPoint("B",B);
  drawLine1(A,B);
}
}
}

if (run == 1) {
// Nullstelle
tng = false;
x3 = prompt('Linke Intervallgrenze a','1');
x3 = myTrim(x3);
x3 = 1*x3;
x4 = prompt('Rechte Intervallgrenze b','3');
x4 = myTrim(x4);
x4 = 1*x4;
if (x3 == x4) { return; }
if (x4 < x3) { return; }

document.MyForm.aus6a.value = x3;
document.MyForm.aus6b.value = x4;
document.MyForm.aus0.value = '';
document.MyForm.aus5.value = '';
document.MyForm.aus5a.value = ' ';
document.MyForm.aus6.value = '';
s = document.MyForm.ein2.value;
xN = nullst(s,x3,x4);                          // ==> "tools.js" (mathe)
if (xN == -999) {
   alert('Keine Nullstelle gefunden!');
   document.MyForm.ein1.value = '';
   return;
}
document.MyForm.ein1.value = xN;
xN = round2(xN);
document.MyForm.aus8.value = xN + '/0';
E.x = 1*xN;
E.y = 0;
drawPoint("N",E);
}

if (run == 2) {
// Extremstelle
tng = false;
x5 = prompt('Linke Intervallgrenze a','-1');
x5 = myTrim(x5);
x5 = 1*x5;
x6 = prompt('Rechte Intervallgrenze b','3');
x6 = myTrim(x6);
x6 = 1*x6;
if (x5 == x6) { return; }
if (x6 < x5) { return; }

document.MyForm.aus6a.value = x5;
document.MyForm.aus6b.value = x6;
s = document.MyForm.ein2.value;
E = extrema(s,x5,x6);                          // ==> "tools.js" (mathe)
```

```
  document.MyForm.aus0.value = '';
  document.MyForm.aus5.value = '';
  document.MyForm.aus5a.value = ' ';
  document.MyForm.aus6.value = '';
  document.MyForm.ein1.value = E.x;
  E.x = round2(E.x);
  if (E.x == -999) {
    alert('Keine Extremstelle gefunden!');
    document.MyForm.ein1.value = '';
    return;
  }
  E.y = round2(E.y);
  texE = ' ' + E.x + '/' + E.y;
  document.MyForm.aus9.value = texE;
  drawPoint("E",E);
  }

  if (run == 3) {
  // Wendestelle
  tng = false;
  x5 = prompt('Linke Intervallgrenze a','0');
  x5 = myTrim(x5);
  x5 = 1*x5;
  x6 = prompt('Rechte Intervallgrenze b','4');
  x6 = myTrim(x6);
  x6 = 1*x6;
  if (x5 == x6) { return; }
  if (x6 < x5) { return; }

  document.MyForm.aus6a.value = x5;
  document.MyForm.aus6b.value = x6;
  s = document.MyForm.ein2.value;
  W = wendpun(s,x5,x6);                          // ==> "tools.js" (mathe)
  document.MyForm.aus0.value = '';
  document.MyForm.aus5.value = '';
  document.MyForm.aus5a.value = ' ';
  document.MyForm.aus6.value = '';
  document.MyForm.ein1.value = W.x;
  W.x = round2(W.x);
  if (W.x == -999) {
    alert('Keine Wendestelle gefunden!');
    document.MyForm.ein1.value = '';
    return;
  }
  W.y = round2(W.y);
  texW = ' ' + W.x + '/' + W.y;
  document.MyForm.aus9a.value = texW;
  drawPoint("W",W);
  }
 }

function Tang() { tng = true; Calcul(); tng = false; }   // Tengente
function Null() { run = 1; Calcul(); run = 0; }          // Nullstelle
function Extr() { run = 2; Calcul(); run = 0; }          // Extremstelle
function Wend() { run = 3; Calcul(); run = 0; }          // Wendestelle

var hide = true;

function Info(){
// Hilfstext ein/aus
  hide = !hide;
  if (hide) {
    document.getElementById("dc").style.visibility = 'visible';
    document.getElementById("tex").style.visibility = 'hidden';
  }
  if (!hide) {
    document.getElementById("dc").style.visibility = 'hidden';
    document.getElementById("tex").style.visibility = 'visible';
  }
}

function Reset(){ document.MyForm.reset(); Clear1(); }

</script>
</head>
```

```
<body id="body" onload = "initCanvas(kb,2,1)" oncontextmenu="return false">
<div id='dc'>
  <canvas id='MyCanvas' width='500' height='500' onmousedown = "showPos(event)">
     Your browser does not support HTML5 Canvas.
  </canvas>
</div>
<form name="MyForm">
<h3>DIFFERENZIALRECHNUNG<br>Funktionen, Ableitungen, Kurvendiskussion</h3>
Halbbreite des Koordinatensystems: <input type="text" name="ein0" value="10" size="4"
class="cd1"><br>
Variable x = <input type="text" name="ein1" value="0" size="22" class="cd1"><br>
f(x) = <input type="text" name="ein2" value="0.5*x^3 - 3*x^2 + 4*x" size="28" class="cd1"><br>
y    = <input type="text" name="aus0" value=" " size="4" class="cd1" readonly>  
y' = <input type="text" name="aus5" value=" " size="4" class="cd1" readonly>  
y'' = <input type="text" name="aus5a" value=" " size="4" class="cd1" readonly> <br>
<br>
<input type="button" name="calc1" value="Start" class="cd2" onclick="Calcul()"> 
<input type="button" name="calc2" value="Clear" class="cd2" onclick="Clear1()"> 
<input type="button" name="calc3" value="Clear all " class="cd2" onclick="Clear2()"> 
<input type="button" name="calc0" value="Reset " class="cd2" onclick="Reset()"> 
<br>
<input type="button" name="calc5" value="Nullstelle" class="cd2" onclick="Null()"> 
<input type="button" name="calc6" value="Extremum" class="cd2" onclick="Extr()"> 
<input type="button" name="calc7" value="Wendepunkt" class="cd2" onclick="Wend()"> 
<input type="button" name="tang" value="Tangente" class="cd2" onclick="Tang()"> 
<br><br>
Tangente: <input type="text" name="aus6" value=" " size="20" class="cd1" readonly><br>
Intervall:    a = <input type="text" name="aus6a" value=" " size="4" class="cd1"
readonly>  
b = <input type="text" name="aus6b" value=" " size="4" class="cd1" readonly> <br>
N: <input type="text" name="aus8" value=" " size="10" class="cd1" readonly>  
E: <input type="text" name="aus9" value=" " size="10" class="cd1" readonly>  
W: <input type="text" name="aus9a" value=" " size="10" class="cd1" readonly>
<br><br>
<div id="txt2">
<b>Operatoren und Funktionen:</b> (, ), +, -, *, /, ^, abs, floor, round, <br>
sqr, sqrt, exp, log, deg, rad, sin, cos, tan, asin, acos, atan, PI und E.<br>
(Die Winkelfunktionen werden im Bogenmaß berechnet.)<br>
<br>
<b>Beispiele:</b> y = log(x) , y = exp(x) , y = sin(x) , y = cos(x) , y = tan(x) <br>
Polynom-Funktionen y = 0.5*x^2 - 3*x - 2 , y = 0.5*x^3 - 3*x^2 + 4*x + 3 <br>
Rationale Funktionen y = 1/(x - 4) , y = 0.4*x^2/(x + 2) , y = x^3/(x^2 - 4) <br>
Sinus y = 3*sin(x + PI/2) ,Gedämpfte Schwingung y = exp(-0.3*x)*sin(4*x) <br>
Gaußsche Glockenkurve y = 3*exp(-x^2/2)/sqrt(2*PI) , mit Halbbreite = 3 <br>
<br>
<b>Hinweis:</b>
[Clear all] und Funktion mit Maus ins Eingabefeld ziehen.<br>
Ein Mausklick zeigt die Punktkoordinaten:  
<input type="text" name="aus4" value=" " size="12" class="cd1" readonly><br>
<br>
<input type="button" id="inf" name="btn" value="Info on/off" class="cd2" onclick="Info()"><br>
</div>
</form>

<div id='tex'>
-------------------------------------------------------------------------------------<br>
Auf einer Funktion y = f(x) liegen die zwei benachbarten Punkte<br>
P(a/f(a)) und Q(x/f(x)). Die Gerade g durch P und Q nennt man<br>
Sekante an die Kurve. Das Dreieck PQR mit dem Punkt R(x/f(a))<br>
heißt Steigungsdreieck und der Winkel bei P heißt Steigungs-<br>
winkel w. Für die Steigung k gilt: k = tan(w) = (f(x)-f(a))/(x-a).<br>
-------------------------------------------------------------------------------------<br>
Nähert sich der Punkt Q kontinuierlich dem Punkt P, dann<br>
wird die Sekante g zur einer Tangente t der Kurve, wenn der<br>
Punkt Q den Punkt P erreicht hat, und k = tan(w) heißt dann<br>
Tangentensteigung oder <b>Differenzialquotient</b> im Punkt P(a/f(a).<br>
Man schreibt f'(a) = lim(f(x)-f(a))/(x-a) für x &rightarrow; a. Eine etwas<br>
andere Schreibweise ist f'(a) = lim(f(a+h)-f(a))/h für h &rightarrow; 0,<br>
wobei h = (x-a) ist. Zu f'(a) sagt man »f Strich an der Stelle a«.<br>
f'(a) wird auch als erste Ableitung von f(x) bezeichnet. Auf die<br>
gleiche Art und Weise kann auch eine zweite Ableitung f''(a) <br>
(f Zwei-Strich) als Ableitung der Ableitung gebildet werden, usw...<br>
Ableitungen werden hier mit numerischen Näherungen ermittelt.<br>
-------------------------------------------------------------------------------------<br>
<div id="tex1">
Die Geradengleichung der Tangente t ist y = k*x + d. Weil nun <br>
P(a/f(a) auf t liegt, gilt f(a) = f'(a)*a + d. Es folgt d = f(a) - f'(a)*a<br>
und für die Tangentengleichung gilt dann: y = f'(a)*(x - a) + f(a).<br>
</div>
-------------------------------------------------------------------------------------<br>
```

```
Für den Graphen einer Funktion gelten folgende Beschreibungen,<br>
die in einer <b>Kurvendiskussion</b> erfasst werden (hier ohne Beweise):<br>
(1a) f(x) ist in x = a monoton steigend, wenn dort f'(a) &gt; 0 ist.<br>
(1b) f(x) ist in x = a monoton fallend, wenn dort f'(a) &lt; 0 ist.<br>
(2a) f(x) ist in x = a nach oben hohl gekrümmt, wenn dort f''(a) &gt; 0 ist.<br>
(2b) f(x) ist in x = a nach unten hohl gekrümmt, wenn dort f''(a) &lt; 0 ist.<br>
(3) Eine Funktion f(x) hat bei a einen Scheitelpunkt (Extremwert), wenn<br>
sich dort nur die Monotonie ändert, d.h. wenn f'(a) = 0 und f''(a) &lt;&gt; 0.<br>
Dann verläuft dort die Tangente parallel zur x-Achse und es gilt:<br>
Tiefpunkt mit f'(a) = 0 und f''(a) > 0. Hochpunkt mit f'(a) = 0 und f''(a) &lt; 0.<br>
(4) Eine Funktion f(x) hat bei a einen Wendepunkt, wenn sich dort nur<br>
die Krümmung ändert, d.h. wenn f'(a) &lt;&gt; 0 und f''(a) = 0 und f'''(a) &lt;&gt; 0.<br>
(5) Eine Funktion f(x) hat bei a einen Flachpunkt, wenn dort gilt:<br>
f'(a) = 0 und f''(a) = 0 und f'''(a) = 0 (z.B. bei a = 0 bei f(x) = x^4/10).<br>
--------------------------------------------------------------------------------<br>
</div>

<script>
   var smallWidth = 800;
   var smallScr = false;
   var scrWidth = window.screen.width;
   var canWidth = scrWidth;
   if (scrWidth <= smallWidth) { smallScr = true; }

   document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=0.65');

   function showPos(ev) {
   // Grafik-Koordinaten auslesen
      var cRect = document.getElementById("dc").getBoundingClientRect();
      var cTop  = cRect.top;
      var cLeft = cRect.left;
      var x1 = (ev.clientX - cLeft);
      var y1 = (ev.clientY - cTop);
      var x0 = x1*2*kw/kb - kw;
      var y0 = (kb - y1)*2*kw/kb - kw;
      var info = round2(x0) + '  /  ' + round2(y0);
      document.MyForm.aus4.value = info;
   }

   document.onkeydown = function(e) {
   // Sperrt alle Funktionstasten (ausgenommem auf Smartphones)
     if (smallScr) { return; }
     if (e.ctrlKey) { return false; }      // Ctrl ...
     let taste = e.keyCode || e.which;
     if (taste === 123)  { return false; } // F12
     if (taste === 122)  { return false; } // F11
     if (taste === 121)  { return false; } // F10
     if (taste === 120)  { return false; } // F9
     if (taste === 119)  { return false; }
     if (taste === 118)  { return false; }
     if (taste === 117)  { return false; }
     if (taste === 116)  { return false; }
     if (taste === 115)  { return false; }
     if (taste === 114)  { return false; }
     if (taste === 113)  { return false; }
     if (taste === 112)  { return false; } // F1
   }
   if (!smallScr) { document.addEventListener("contextmenu", (e) => {e.preventDefault()}); }

   function smart() {
   // Anpassung an Smartphones
      document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=0.65');
      let anf = document.body.offsetTop;
      let el0 = document.getElementById("body");
      let el1 = document.getElementById("dc");
      let el2 = document.getElementById("tex");
      el0.style.margin = 10 + "px";
      el1.style.top = anf + 700 + "px";
      el1.style.left = 10 + "px";
      el2.style.top = anf + 700 + "px";
      el2.style.left = 10 + "px";
      window.scroll(0,0);
      // alert('Weiter');
   }

   document.MyForm.calc1.focus();
   if (smallScr) { smart(); }
</script>

</body>
</html>
```

- **Das Programm „integral.html“**

INTEGRALRECHNUNG
Bestimmtes Integral ∫f(x)dx auf [a,b]

Halbbreite des Koordinatensystems: 10
f(x) = sqrt(4*x)
x-Wert = 0 y-Wert =

Start | Clear | Clear all | Reset
Fläche | Flächen-Schwerpunkt
Volumen | Mantel | Körper-Schwerpunkt
Bogenlänge | Bogenschwerpunkt

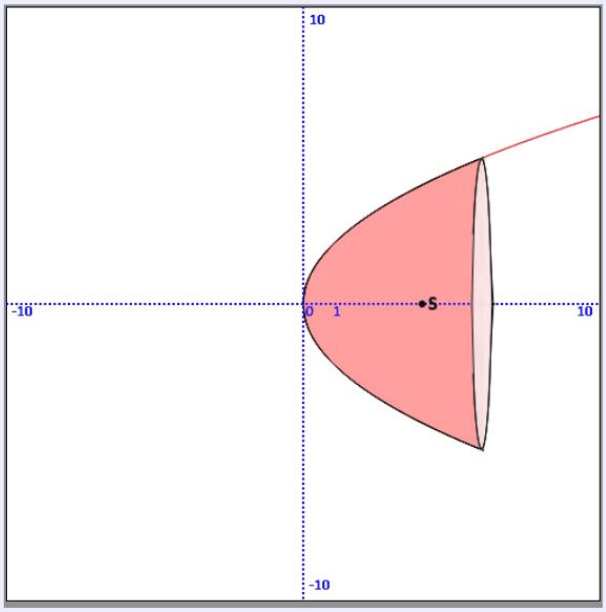
Intervall: a = 0.0005 b = 6
Im Intervall kein Vorzeichenwechsel von f(x).
Integral (Fläche): Schwerpunkt:
Volumen: 226.19 Mantel: 146.77
Körperschwerpunkt: S(4/0)
Bogenlänge: Bogenschwerpunkt:

Operatoren und Funktionen: (,), +, -, *, /, ^, abs, floor, round, sqr, sqrt, exp, log, deg, rad, sin, cos, tan, asin, acos, atan, PI und E.
(Die Winkelfunktionen werden im Bogenmaß berechnet.)

Beispiele: y = log(x) , y = exp(x) , y = sin(x) , y = cos(x) , y = tan(x)
y = 1*x , y = 0.5*x^3 - 3*x^2 + 4*x , y = 2/x , Parabel y = sqrt(4*x)
Ellipse y = (3/5)*sqrt(5^2 - x^2) , Hyperbel y = (3/2)*sqrt(x^2 - 2^2)

Hinweis: [Clear all] und Funktion mit Maus ins Eingabefeld ziehen.
Ein Mausklick zeigt die Punktkoordinaten:

Info on/off



INTEGRALRECHNUNG
Bestimmtes Integral ∫f(x)dx auf [a,b]

Halbbreite des Koordinatensystems: 10
f(x) = sqrt(4*x)
x-Wert = 0 y-Wert =

Start | Clear | Clear all | Reset
Fläche | Flächen-Schwerpunkt
Volumen | Mantel | Körper-Schwerpunkt
Bogenlänge | Bogenschwerpunkt

Intervall: a = 0.0005 b = 6
Im Intervall kein Vorzeichenwechsel von f(x).
Integral (Fläche): Schwerpunkt:
Volumen: 226.19 Mantel: 146.77
Körperschwerpunkt: S(4/0)
Bogenlänge: Bogenschwerpunkt:

Operatoren und Funktionen: (,), +, -, *, /, ^, abs, floor, round, sqr, sqrt, exp, log, deg, rad, sin, cos, tan, asin, acos, atan, PI und E.
(Die Winkelfunktionen werden im Bogenmaß berechnet.)

Beispiele: y = log(x) , y = exp(x) , y = sin(x) , y = cos(x) , y = tan(x)
y = 1*x , y = 0.5*x^3 - 3*x^2 + 4*x , y = 2/x , Parabel y = sqrt(4*x)
Ellipse y = (3/5)*sqrt(5^2 - x^2) , Hyperbel y = (3/2)*sqrt(x^2 - 2^2)

Hinweis: [Clear all] und Funktion mit Maus ins Eingabefeld ziehen.
Ein Mausklick zeigt die Punktkoordinaten:

Info on/off

Mit dem bestimmten Integral ∫ydx der stetigen Funktion y = f(x) auf dem Intervall [a,b] wird jene Fläche A in der Ebene ermittelt, welche von der Funktionskurve y = f(x), der x-Achse, und links und rechts von den beiden Geraden x = a und x = b begrenzt wird. Die Integrale werden hier mit numerischen Näherungen ermittelt. Die nachfolgenden Anwendungen werden ohne Beweise erklärt.

Für den Schwerpunkt S(c/d) der Fläche A gilt:
Koordinate c = 1/A*∫x*ydx, und Koordinate d = 1/(2A)*∫y²dx.
Hinweis:
Wenn V = ¶*∫y²dx das Volumen des Drehkörpers ist, dann gilt:
V = ¶*∫y²dx = A*2¶*d. Diese Formel heißt **»Erste Guldin-Regel«**:
Das Volumen eines Drehkörpers ist gleich dem Produkt aus der rotierenden Fläche A und der Kreisbahn des Schwerpunktes.

Die Kurve rotiert um die x-Achse und erzeugt einen Drehkörper.
Für das Volumen V des Drehkörpers gilt: V = ¶*∫y²dx.
Für die Mantelfläche M gilt: M = 2¶*∫y*sqrt(1+[y']²)dx.
y' = f'(x) ist die erste Ableitung, sqrt ist die Quadratwurzel.
Für den Körper-Schwerpunkt S(c/0) gilt: c = (1/V)*¶*∫x*y²dx.

Für die Länge des Kurvenbogens e gilt: e = ∫sqrt(1+[y']²)dx.
y' = f'(x) ist die erste Ableitung, sqrt ist die Quadratwurzel.
Für den Schwerpunkt S(c/d) des Bogens e gilt:
c = (1/e)*∫x*sqrt(1+[y']²)dx, und d = (1/e)*∫y*sqrt(1+[y']²)dx.
Hinweis:
Wenn M = 2¶*∫y*sqrt(1+[y']²)dx der Mantel des Drehkörpers ist, gilt: M = 2¶*∫y*sqrt(1+[y']²)dx = e*2¶*d. **»Zweite Guldin-Regel«**:
Der Mantel eines Drehkörpers ist gleich dem Produkt aus dem rotierenden Bogen e und der Kreisbahn des Schwerpunktes.

Achtung: Wenn f(x), f'(x) oder f''(x) in einem Randpunkt eines Intervalls unendlich wird, dann können Fehler auftreten. Dann muss für den x-Wert des Randpunktes ein davon nur geringfügig abweichender Wert genommen werden. Beispielsweise ist bei der Kreisfunktion f(x) = sqrt(25-x^2) die erste Ableitung f'(x) an der Stelle x = 5 unendlich (senkrechte Kurventangente). Zum Weiterrechnen muss dann x = 4.9995 genommen werden.

```
<!DOCTYPE html>
<html lang="de">
<head>
<title>integral.html</title>
<meta charset="ISO-8859-1">
<meta name="viewport" content="width=device-width, initial-scale=1.00">
<meta name="author" content="Paukert Herbert">
<meta name="description" content="Integralrechnung">

<style>
body  {background-color:#E4E4F8; color:black; font-family:Arial;
       font-size:18px; font-weight:normal; text-align:left; margin:2%;}
.cd1  {border: 1px solid #666666; background-color: #FFFFFF; font-size: 16px; margin: 2px;}
.cd2  {border: 2px solid #666666; border-radius: 4px; background-color: #FAEADA;
       font-size: 16px; font-weight: bold; margin: 3px;}
#txt2 {color: black; font-size: 16px;}
#tex  {position:absolute; top:20px; left:650px; font-size:15px; visibility:hidden;}
#tex1 {color: blue;}
#tex2 {color: blue;}
#tex3 {color: darkred;}
#dc   {position:absolute; top:125px; left:650px; background-color:#888888;
       border: 2px solid #888888;}
.lh-  {line-height: 5px; margin-top: 5px; margin-bottom: 5px;}
.colred {color:red;  font-weight: bold; font-style: italic;}
h3 {color: darkred;}
hr {width:80%; margin-left:0; border: 2px solid #666666;}
</style>

<script src = "tools.js"></script>

<script>
  var okay = false;
  var run = 0;
  var anz = 0;
  A = new TPoint(0,0);
  B = new TPoint(0,0);
  C = new TPoint(0,0);
  D = new TPoint(0,0);
  E = new TPoint(0,0);
  F = new TPoint(0,0);
  G = new TPoint(0,0);
  H = new TPoint(0,0);
  S = new TPoint(0,0);
  O = new TPoint(0,0);
  var s;  // +Funktionsterm
  var t;  // -Funktionsterm
  var x1 = 0;
  var x2 = 5;
  var fla;
  var vol;
  var sX, sY;
  var swp;

  function Clear1() {
  // Grafik löschen
    document.MyForm.aus4.value = ' ';
    document.MyForm.aus6a.value = ' ';
    document.MyForm.aus6b.value = ' ';
    document.MyForm.aus7.value = ' ';
    document.MyForm.aus8.value = ' ';
    document.MyForm.aus9.value = ' ';
    document.MyForm.aus10.value = ' ';
    document.MyForm.aus11.value = ' ';
    document.MyForm.aus12.value = ' ';
    document.MyForm.aus13.value = ' ';
    kw = 1 * document.MyForm.ein0.value;
    if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) {kw = 10; document.MyForm.ein0.value = kw;};
    initCanvas(kb,2,1);
    document.MyForm.calc1.focus();
  }

  function Clear2() {
  // Grafik und Funktionszeile löschen
    x1 = 0;
    x2 = 5;
    document.MyForm.ein2.value = ' ';
    document.MyForm.aus4.value = ' ';
    document.MyForm.aus6a.value = ' ';
    document.MyForm.aus6b.value = ' ';
    document.MyForm.aus7.value = ' ';
    document.MyForm.aus8.value = ' ';
    document.MyForm.aus9.value = ' ';
```

```
    document.MyForm.aus10.value = ' ';
    document.MyForm.aus11.value = ' ';
    document.MyForm.aus12.value = ' ';
    document.MyForm.aus13.value = ' ';
    kw = 1 * document.MyForm.ein0.value;
    if ((isNaN(kw)) || (kw < 0) || (kw > 1000)) {kw = 10; document.MyForm.ein0.value = kw;};
    initCanvas(kb,2,1);
    document.MyForm.calc1.focus();
  }

function Calcul1() {
// Hauptroutine zur grafischen Funktions-Darstellung
  Clear1();
  x = 1 * document.MyForm.ein1.value;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y = fun(s,x);                                  // ==> "tools.js" (parser)
  y = round4(y);
  document.MyForm.aus0.value = y;
  A.x = x;
  A.y = y;
  drawFun(kw,kb,s);
  drawPoint('',A);
  O.x = 0; O.y = 0;
  C.x = 0; C.y = 0;
  D.x = 0; D.y = 0;
  l = A.len();
  l = round4(l);
  w = A.win();
  w = round4(deg(w));
  okay = true;
//  alert('Betrag OA = ' + l + ', Winkel OA = ' + w);
  document.MyForm.calc1.focus();
 }

function Calcul2() {
// Hauptroutine für Inhalt und Schwerpunkt der Fläche unter der Kurve
  Clear1();
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  drawFun(kw,kb,s);
  O.x = 0; O.y = 0;
  C.x = 0; C.y = 0;
  D.x = 0; D.y = 0;
  E.x = 0; E.y = 0;

if (run == 1) {
// Fläche (Integral) ------------------------------------------------
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  x1 = prompt('Linke Intervallgrenze a',x1);
  x1 = myTrim(x1);
  x1 = 1*x1;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y2 = fun(s,x1);
  C.x = x1; C.y = 0;
  E.x = x1; E.y = y2;
  x2 = prompt('Rechte Intervallgrenze b',x2);;
  x2 = myTrim(x2);
  x2 = 1*x2;
  if (x1 == x2) { return; }
  if (x2 < x1) { return; }
  s = document.MyForm.ein2.value;
  y2 = fun(s,x2);
  D.x = x2; D.y = 0;
  F.x = x2; F.y = y2;
  document.MyForm.aus6a.value = x1;
  document.MyForm.aus6b.value = x2;
  drawWidth(2);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,s);
  drawPoint("a",C);
  drawPoint("b",D);
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  fla = integ(s,x1,x2);                          // ==> "tools.js" (mathe)
  if (fla == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  fla = round2(fla);
  document.MyForm.aus7.value = fla;
  return;
}
```

```
if (run == 2) {
// Schwerpunkt ----------------------------------------------------
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  x1 = prompt('Linke Intervallgrenze a',x1);
  x1 = myTrim(x1);
  x1 = 1*x1;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y2 = fun(s,x1);
  C.x = x1; C.y = 0;
  drawPoint("a",C);
  E.x = x1; E.y = y2;
  x2 = prompt('Rechte Intervallgrenze b',x2);;
  x2 = myTrim(x2);
  x2 = 1*x2;
  if (x1 == x2) { return; }
  if (x2 < x1) { return; }
  s = document.MyForm.ein2.value;
  y2 = fun(s,x2);
  D.x = x2; D.y = 0;
  drawPoint("b",D);
  F.x = x2; F.y = y2;
  document.MyForm.aus6a.value = x1;
  document.MyForm.aus6b.value = x2;
  drawWidth(2);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,s);
  drawPoint("a",C);
  drawPoint("b",D);
  s = document.MyForm.ein2.value;
  fla = integ(s,x1,x2);                          // ==> "tools.js" (mathe)
  if (fla == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  document.MyForm.aus7.value = round2(fla);
  s = document.MyForm.ein2.value;
  sX = integflaX(s,x1,x2);                       // ==> "tools.js" (mathe)
  if (sX == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  sX = round2(sX/fla);
  s = document.MyForm.ein2.value;
  sY = integflaY(s,x1,x2);                       // ==> "tools.js" (mathe)
  if (sY == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  sY = round2(sY/fla);
  swp = 'S(' + sX + '/' + sY + ')';
  document.MyForm.aus8.value = swp;
  S.x = sX;
  S.y = sY;
  drawPoint("S",S);
}
}

function Inte2()   { run = 1; Calcul2(); run = 0; }   // Flächeninhalt
function Schwer2() { run = 2; Calcul2(); run = 0; }   // Flächenschwerpunkt

function Calcul3() {
// Hauptroutine für Volumen, Oberfläche und Schwerpunkt
  Clear1();
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  t = '-(' + s + ')';
  drawFun(kw,kb,s);
  O.x = 0;
  O.y = 0;
  C.x = 0;
  C.y = 0;
  D.x = 0;
  D.y = 0;
  E.x = 0;
  E.y = 0;

if (run == 1) {
// Volumen --------------------------------------------------------
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  x1 = prompt('Linke Intervallgrenze a',x1);
  x1 = myTrim(x1);
  x1 = 1*x1;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y1 = fun(s,x1);
  C.x = x1; C.y = 0;
  E.x = x1; E.y = y1;
```

```
  x2 = prompt('Rechte Intervallgrenze b',x2);;
  x2 = myTrim(x2);
  x2 = 1*x2;
  if (x1 == x2) { return; }
  if (x2 < x1) { return; }
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y2 = fun(s,x2);
  D.x = x2; D.y = 0;
  F.x = x2; F.y = y2;
  document.MyForm.aus6a.value = x1;
  document.MyForm.aus6b.value = x2;
  drawWidth(2);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,s);
  drawFun1(kw,kb,x1,x2,s);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,t);
  drawFun1(kw,kb,x1,x2,t);
  vol = integvol(s,x1,x2);                        // ==> "tools.js" (mathe)
  if (vol == -999) { alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  vol = round2(vol);
  document.MyForm.aus9.value = vol;
  G.x = E.x; G.y = -E.y;
  H.x = F.x; H.y = -F.y;
  drawLine(E,G);
  drawLine(F,H);
  f = 1.5 * document.MyForm.ein0.value;
  if ( Math.abs(y1) > Math.abs(y2) ) {
     drawColor("rgb(240,220,220)");
     fillEll(C,y1/f,y1,1);
     fillEll(C,y1/f,y1,-1);
     drawColor("rgb(0,0,0)");
     drawEll(C,y1/f,y1,1);
     drawEll(C,y1/f,y1,-1);
  }
  else {
     drawColor("rgb(240,220,220)");
     fillEll(D,y2/f,y2,1);
     fillEll(D,y2/f,y2,-1);
     drawColor("rgb(0,0,0)");
     drawEll(D,y2/f,y2,1);
     drawEll(D,y2/f,y2,-1);
  }
  // drawPoint("a",C); drawPoint("b",D);
    return;
}

if (run == 2) {
// Oberfläche ----------------------------------------------------
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  x1 = prompt('Linke Intervallgrenze a',x1);
  x1 = myTrim(x1);
  x1 = 1*x1;
  y1 = fun(s,x1);
  C.x = x1; C.y = 0;
  E.x = x1; E.y = y1;
  x2 = prompt('Rechte Intervallgrenze b',x2);;
  x2 = myTrim(x2);
  x2 = 1*x2;
  if (x1 == x2) { return; }
  if (x2 < x1) { return; }
  y2 = fun(s,x2);
  D.x = x2; D.y = 0;
  F.x = x2; F.y = y2;
  document.MyForm.aus6a.value = x1;
  document.MyForm.aus6b.value = x2;
  drawWidth(2);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,s);
  drawFun1(kw,kb,x1,x2,s);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,t);
  drawFun1(kw,kb,x1,x2,t);

  vol = integvol(s,x1,x2);                        // ==> "tools.js" (mathe)
  if (vol == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  vol = round2(vol);
  document.MyForm.aus9.value = vol;
```

```
  fla = integfla(s,x1,x2);                          // ==> "tools.js" (mathe)
  if (fla == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  fla = round2(fla);
  document.MyForm.aus10.value = fla;

  G.x = E.x; G.y = -E.y;
  H.x = F.x; H.y = -F.y;
  drawLine(E,G);
  drawLine(F,H);
  f = 1.5 * document.MyForm.ein0.value;
  if ( Math.abs(y1) > Math.abs(y2) ) {
     drawColor("rgb(240,220,220)");
     fillEll(C,y1/f,y1,1);
     fillEll(C,y1/f,y1,-1);
     drawColor("rgb(0,0,0)");
     drawEll(C,y1/f,y1,1);
     drawEll(C,y1/f,y1,-1);
  }
  else {
     drawColor("rgb(240,220,220)");
     fillEll(D,y2/f,y2,1);
     fillEll(D,y2/f,y2,-1);
     drawColor("rgb(0,0,0)");
     drawEll(D,y2/f,y2,1);
     drawEll(D,y2/f,y2,-1);
  }
  //  drawPoint("a",C); drawPoint("b",D);
}

if (run == 3) {
// Schwerpunkt ----------------------------------------------------
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  x1 = prompt('Linke Intervallgrenze a',x1);
  x1 = myTrim(x1);
  x1 = 1*x1;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y1 = fun(s,x1);
  C.x = x1; C.y = 0;
  E.x = x1; E.y = y1;
  x2 = prompt('Rechte Intervallgrenze b',x2);;
  x2 = myTrim(x2);
  x2 = 1*x2;
  if (x1 == x2) { return; }
  if (x2 < x1) { return; }
  s = document.MyForm.ein2.value;
  y2 = fun(s,x2);
  D.x = x2; D.y = 0;
  F.x = x2; F.y = y2;
  document.MyForm.aus6a.value = x1;
  document.MyForm.aus6b.value = x2;
  drawWidth(2);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,s);
  drawFun1(kw,kb,x1,x2,s);
  drawColor("rgb(250,150,150)");
  fillFun1(kw,kb,x1,x2,t);
  drawFun1(kw,kb,x1,x2,t);
  s = document.MyForm.ein2.value;
  vol = integvol(s,x1,x2);                          // ==> "tools.js" (mathe)
  if (vol == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  vol = round2(vol);
  document.MyForm.aus9.value = round2(vol);
  fla = integfla(s,x1,x2);                          // ==> "tools.js" (mathe)
  if (fla == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  fla = round2(fla);
  document.MyForm.aus10.value = fla;

  G.x = E.x; G.y = -E.y;
  H.x = F.x; H.y = -F.y;
  drawLine(E,G);
  drawLine(F,H);
  f = 1.5 * document.MyForm.ein0.value;
  if ( Math.abs(y1) > Math.abs(y2) ) {
     drawColor("rgb(240,220,220)");
     fillEll(C,y1/f,y1,1);
     fillEll(C,y1/f,y1,-1);
     drawColor("rgb(0,0,0)");
     drawEll(C,y1/f,y1,1);
     drawEll(C,y1/f,y1,-1);
  }
```

```
  else {
     drawColor("rgb(240,220,220)");
     fillEll(D,y2/f,y2,1);
     fillEll(D,y2/f,y2,-1);
     drawColor("rgb(0,0,0)");
     drawEll(D,y2/f,y2,1);
     drawEll(D,y2/f,y2,-1);
  }
//  drawPoint("a",C); drawPoint("b",D);
  s = document.MyForm.ein2.value;
  sX = integvolX(s,x1,x2);                        // ==> "tools.js" (mathe)
  sX = sX / vol;
  if (sX == -999) {alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!'); return;}
  sX = round2(sX);
  sY = 0;
  swp = 'S(' + sX + '/' + sY + ')';
  document.MyForm.aus11.value = swp;
  S.x = sX;
  S.y = sY;
  drawPoint("S",S);
}
document.MyForm.calc1.focus();
}

function Volu()    { run = 1; Calcul3(); run = 0; }  // Volumen
function Flae()    { run = 2; Calcul3(); run = 0; }  // Oberfläche
function Schwer3() { run = 3; Calcul3(); run = 0; }  // Schwerpunkt


function Calcul4() {
// Hauptroutine für Bogenlänge und Bogenschwerpunkt
  Clear1();
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  drawFun(kw,kb,s);
  O.x = 0;
  O.y = 0;
  C.x = 0;
  C.y = 0;
  D.x = 0;
  D.y = 0;
  E.x = 0;
  E.y = 0;

if (run == 1) {
// Bogen ------------------------------------------------------------
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  x1 = prompt('Linke Intervallgrenze a',x1);
  x1 = myTrim(x1);
  x1 = 1*x1;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y2 = fun(s,x1);
  C.x = x1; C.y = 0;
  drawPoint("a",C);
  E.x = x1; E.y = y2;
  drawPoint("",E);
  drawLine(C,E);
  x2 = prompt('Rechte Intervallgrenze b',x2);;
  x2 = myTrim(x2);
  x2 = 1*x2;
  if (x1 == x2) { return; }
  if (x2 < x1) { return; }
  s = document.MyForm.ein2.value;
  y2 = fun(s,x2);
  D.x = x2; D.y = 0;
  drawPoint("b",D);
  F.x = x2; F.y = y2;
  drawPoint("",F);
  drawLine(D,F);
  drawColor("#000000");
  drawWidth(3);
  drawFun1(kw,kb,x1,x2,s);
  document.MyForm.aus6a.value = x1;
  document.MyForm.aus6b.value = x2;
  s = document.MyForm.ein2.value;
  bog = integbog(s,x1,x2);                        // ==> "tools.js" (mathe)
  if (bog == -999) { alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!');  return; }
  bog = round2(bog);
  document.MyForm.aus12.value = bog;
}
```

```
if (run == 2) {
// Schwerpunkt ----------------------------------------------------
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  x1 = prompt('Linke Intervallgrenze a',x1);
  x1 = myTrim(x1);
  x1 = 1*x1;
  s = document.MyForm.ein2.value;
  s = myTrim(s);
  y2 = fun(s,x1);
  C.x = x1; C.y = 0;
  drawPoint("a",C);
  E.x = x1; E.y = y2;
  drawPoint("",E);
  drawLine(C,E);
  x2 = prompt('Rechte Intervallgrenze b',x2);;
  x2 = myTrim(x2);
  x2 = 1*x2;
  if (x1 == x2) { return; }
  if (x2 < x1) { return; }

  s = document.MyForm.ein2.value;
  y2 = fun(s,x2);
  D.x = x2; D.y = 0;
  drawPoint("b",D);
  F.x = x2; F.y = y2;
  drawPoint("",F);
  drawLine(D,F);
  document.MyForm.aus6a.value = x1;
  document.MyForm.aus6b.value = x2;
  s = document.MyForm.ein2.value;
  bog = integbog(s,x1,x2);                      // ==> "tools.js" (mathe)
  if (bog == -999) { alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!');  return; }
  document.MyForm.aus12.value = round2(bog);
  s = document.MyForm.ein2.value;
  sX = integbogX(s,x1,x2);                      // ==> "tools.js" (mathe)
  if (sX == -999) { alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!');  return; }
  sX = round2(sX/bog);
  s = document.MyForm.ein2.value;
  sY = integbogY(s,x1,x2);                      // ==> "tools.js" (mathe)
  if (sY == -999) { alert('Abbruch: f(x) = Undefined oder f´(x) = Infinity!');  return; }
  sY = round2(sY/bog);
  swp = 'S(' + sX + '/' + sY + ')';
  document.MyForm.aus13.value = swp;
  S.x = sX;
  S.y = sY;
  drawPoint("S",S);
}
}

function Bogen()   { run = 1; Calcul4(); run = 0; }  // Bogenlänge
function Schwer4() { run = 2; Calcul4(); run = 0; }  // Bogenschwerpunkt

var hide = true;

function Info(){
// Hilfstext ein/aus
  hide = !hide;
  if (hide) {
    document.getElementById("dc").style.visibility = 'visible';
    document.getElementById("tex").style.visibility = 'hidden';
  }
  if (!hide) {
    document.getElementById("dc").style.visibility = 'hidden';
    document.getElementById("tex").style.visibility = 'visible';
  }
}

function Reset() { document.MyForm.reset(); Clear1(); }
</script>
</head>

<body id="body" onload = "initCanvas(kb,2,1)" oncontextmenu="return false">
<div id='dc'>
  <canvas id='MyCanvas' width='500' height='500' onmousedown = "showPos(event)">
     Your browser does not support HTML5 Canvas.
  </canvas>
</div>
```

```
<form name="MyForm">
<h3>INTEGRALRECHNUNG<br>Bestimmtes Integral &int;f(x)dx auf [a,b]</h3>
Halbbreite des Koordinatensystems: <input type="text" name="ein0" value="10" size="4" class="cd1">
<br>
f(x) = <input type="text" name="ein2" value="x^2/4" size="40" class="cd1"><br>
x-Wert = <input type="text" name="ein1" value="0" size="8" class="cd1">  
y-Wert = <input type="text" name="aus0" value=" " size="8" class="cd1" readonly><br>
<div class="lh-">  </div>
<input type="button" name="calc1" value="Start" class="cd2" onclick="Calcul1()"> 
<input type="button" name="calc2" value="Clear" class="cd2" onclick="Clear1()"> 
<input type="button" name="calc3" value="Clear all " class="cd2" onclick="Clear2()">
<input type="button" name="calc3" value="Reset " class="cd2" onclick="Reset()"><br>
<input type="button" name="calc4" value="Fläche" class="cd2" onclick="Inte2()"> 
<input type="button" name="calc5" value="Flächen-Schwerpunkt" class="cd2" onclick="Schwer2()"><br>
<input type="button" name="calc6" value="Volumen" class="cd2" onclick="Volu()"> 
<input type="button" name="calc7" value="Mantel" class="cd2" onclick="Flae()"> 
<input type="button" name="calc8" value="Körper-Schwerpunkt" class="cd2" onclick="Schwer3()"><br>
<input type="button" name="calc9" value="Bogenlänge" class="cd2" onclick="Bogen()"> 
<input type="button" name="calc10" value="Bogenschwerpunkt" class="cd2" onclick="Schwer4()"><br>
<div class="lh-">  </div>
Intervall:  a = <input type="text" name="aus6a" value=" " size="4" class="cd1" readonly> 
b = <input type="text" name="aus6b" value=" " size="4" class="cd1" readonly> <br>
Im Intervall kein Vorzeichenwechsel von f(x).<br>
Integral (Fläche):   <input type="text" name="aus7" value=" " size="6" class="cd1"
readonly> 
Schwerpunkt: <input type="text" name="aus8" value=" " size="12" class="cd1" readonly><br>
Volumen: <input type="text" name="aus9" value=" " size="6" class="cd1" readonly>  
Mantel: <input type="text" name="aus10" value=" " size="7" class="cd1" readonly><br>
Körperschwerpunkt: <input type="text" name="aus11" value=" " size="16" class="cd1" readonly><br>
Bogenlänge: <input type="text" name="aus12" value=" " size="6" class="cd1" readonly>
Bogenschwerpunkt: <input type="text" name="aus13" value=" " size="12" class="cd1" readonly><br>
<div class="lh-">  </div>
<div id="txt2">
<b>Operatoren und Funktionen:</b> (, ), +, -, *, /, ^, abs, floor, round, <br>
sqr, sqrt, exp, log, deg, rad, sin, cos, tan, asin, acos, atan, PI und E.<br>
(Die Winkelfunktionen werden im Bogenmaß berechnet.)<br>
<div class="lh-">  </div>
<b>Beispiele:</b> y = log(x) , y = exp(x) , y = sin(x) , y = cos(x) , y = tan(x) <br>
y = 1*x , y = 0.5*x^3 - 3*x^2 + 4*x , y = 2/x , Parabel y = sqrt(4*x) <br>
Ellipse y = (3/5)*sqrt(5^2 - x^2) , Hyperbel y = (3/2)*sqrt(x^2 - 2^2) <br>
<div class="lh-">  </div>
<b>Hinweis:</b> [Clear all] und Funktion mit Maus ins Eingabefeld ziehen.<br>
Ein Mausklick zeigt die Punktkoordinaten: 
<input type="text" name="aus4" value=" " size="12" class="cd1" readonly><br>
<input type="button" id="inf" name="btn" value="Info on/off" class="cd2" onclick="Info()">
</div>
</form>

<div id='tex'>
<br>
Mit dem bestimmten Integral &int;ydx der stetigen Funktion y = f(x)<br>
auf dem Intervall [a,b] wird jene Fläche A in der Ebene ermittelt,<br>
welche von der Funktionskurve y = f(x), der x-Achse, und links<br>
und rechts von den beiden Geraden x = a und x = b begrenzt wird.<br>
Die Integrale werden hier mit numerischen Näherungen ermittelt.<br>
Die nachfolgenden Anwendungen werden ohne Beweise erklärt.<br>
-------------------------------------------------------------------------------<br>
Für den Schwerpunkt S(c/d) der Fläche A gilt:<br>
Koordinate c = 1/A*&int;x*ydx, und Koordinate d = 1/(2A)*&int;y²dx.<br>
<b>Hinweis:</b><br>
Wenn V = ¶*&int;y²dx das Volumen des Drehkörpers ist, dann gilt:<br>
V = ¶*&int;y²dx = A*2¶*d. Diese Formel heißt <b>»Erste Guldin-Regel«</b>:<br>
<div id="tex2">
Das Volumen eines Drehkörpers ist gleich dem Produkt aus der<br>
rotierenden Fläche A und der Kreisbahn des Schwerpunktes.<br>
</div>
-------------------------------------------------------------------------------<br>
Die Kurve rotiert um die x-Achse und erzeugt einen Drehkörper.<br>
Für das Volumen V des Drehkörpers gilt: V = ¶*&int;y²dx.<br>
Für die Mantelfläche M gilt: M = 2¶*&int;y*sqrt(1+[y']²)dx.<br>
y' = f'(x) ist die erste Ableitung, sqrt ist die Quadratwurzel.<br>
Für den Körper-Schwerpunkt S(c/0) gilt: c = (1/V)*¶*&int;x*y²dx.<br>
-------------------------------------------------------------------------------<br>
Für die Länge des Kurvenbogens e gilt: e = &int;sqrt(1+[y']²)dx.<br>
y' = f'(x) ist die erste Ableitung, sqrt ist die Quadratwurzel.<br>
Für den Schwerpunkt S(c/d) des Bogens e gilt:<br>
c = (1/e)*&int;x*sqrt(1+[y']²)dx, und d = (1/e)*&int;y*sqrt(1+[y']²)dx.<br>
<b>Hinweis:</b><br>
Wenn M = 2¶*&int;y*sqrt(1+[y']²)dx der Mantel des Drehkörpers ist,<br>
gilt: M = 2¶*&int;y*sqrt(1+[y']²)dx = e*2¶*d. <b>»Zweite Guldin-Regel«</b>:<br>
<div id="tex2">
Der Mantel eines Drehkörpers ist gleich dem Produkt aus dem<br>
rotierenden Bogen e und der Kreisbahn des Schwerpunktes.<br>
</div>
```

```
-------------------------------------------------------------------------------------<br>
<div id="tex3">
<b>Achtung:</b> Wenn f(x), f'(x) oder f''(x) in einem Randpunkt eines<br>
Intervalls unendlich wird, dann können Fehler auftreten. Dann<br>
muss für den x-Wert des Randpunktes ein davon nur geringfügig<br>
abweichender Wert genommen werden. Beispielsweise ist bei<br>
der Kreisfunktion f(x) = sqrt(25-x^2) die erste Ableitung f'(x)<br>
an der Stelle x = 5 unendlich (senkrechte Kurventangente).<br>
Zum Weiterrechnen muss dann x = 4.9995 genommen werden.<br>
</div>
<br>
</div>

<script>
   var smallWidth = 800;
   var smallScr = false;
   var scrWidth = window.screen.width;
   var canWidth = scrWidth;
   if (scrWidth <= smallWidth) { smallScr = true; }

   document.querySelector('meta[name="viewport"]').setAttribute('content','initial-scale=0.65');

   function showPos(ev) {
   // Grafik-Koordinaten auslesen
      var cRect = document.getElementById("dc").getBoundingClientRect();
      var cTop  = cRect.top;
      var cLeft = cRect.left;
      var x1 = (ev.clientX - cLeft);
      var y1 = (ev.clientY - cTop);
      var x0 = x1*2*kw/kb - kw;
      var y0 = (kb - y1)*2*kw/kb - kw;
      var info = round2(x0) + '  /  ' + round2(y0);
      document.MyForm.aus4.value = info;
   }

   document.onkeydown = function(e) {
   // Sperrt alle Funktionstasten (ausgenommem auf Smartphones)
     if (smallScr) { return; }
     if (e.ctrlKey) { return false; }      // Ctrl ...
     let taste = e.keyCode || e.which;
     if (taste === 123)  { return false; } // F12
     if (taste === 122)  { return false; } // F11
     if (taste === 121)  { return false; } // F10
     if (taste === 120)  { return false; } // F9
     if (taste === 119)  { return false; }
     if (taste === 118)  { return false; }
     if (taste === 117)  { return false; }
     if (taste === 116)  { return false; }
     if (taste === 115)  { return false; }
     if (taste === 114)  { return false; }
     if (taste === 113)  { return false; }
     if (taste === 112)  { return false; } // F1
   };
   if (!smallScr) { document.addEventListener("contextmenu", (e) => {e.preventDefault()}); }

   function smart() {
   // Anpassung an Smartphones
      document.getElementById("txt2").style.fontSize = 16 + "px";
      let end = document.body.offsetHeight;
      let el0 = document.getElementById("body");
      let el1 = document.getElementById("dc");
      let el2 = document.getElementById("tex");
      let anf = document.body.offsetTop;
      el0.style.margin = 10 + "px";
      el1.style.top = anf + 790 + "px";
      el1.style.left = 10 + "px";
      el2.style.top = anf + 790 + "px";
      el2.style.left = 10 + "px";
      window.scroll(0,0);
   }

   document.MyForm.calc1.focus();
   if (smallScr) { smart(); window.scroll(0,0); }
</script>

</body>
</html>

-------------------------------------------------------------------------------------------------------------------
```

• Verzeichnis aller „Grafik“-Bibliotheks-Funktionen

```
var kw = 10;                            // halbe Weltbreite
var kb = 500;                           // ganze Bildbreite
var visa = 1;                           // Sichtbarkeit des Koordinatensystems
var srk = 0.5;                          // Schrägrissverkürzung
var srw = 30;                           // Schrägrisswinkel

function myTrim(x)                      // Führende und folgende Blanks von x entfernen
function round2(x)                      // x auf 2 Dezimalen runden
function round4(x)                      // x auf 4 Dezimalen runden
function deg(x)                         // Umwandlung von Bogenmaß in Gradmaß
function rad(x)                         // Umwandlung von Gradmaß in Bogenmaß
function Max6(a,b,c,d,e,f)              // Maximum von 6 Zahlen
function Min6(a,b,c,d,e,f)              // Minimum von 6 Zahlen
function Amax10(a,b,c,d,e,f,g,h,i,j)    // Absolutes Maximum von 10 Zahlen

function TPoint(tx,ty)                  // Konstruktor für ein Punkt-Objekt (2-dim.)
function TPoint3(tx,ty,tz)              // Konstruktor für ein Punkt-Objekt (3-dim.)
function w2p(P9,kw,kb)                  // World-To-Picture-Transformation von Punkt P9
function p2w(P9,kw,kb)                  // Picture-To-World-Transformation von Punkt P9
function Schraeg(P9)                    // Schrägrisstransformation von Punkt P9
function initCanvas(kb,n,visa)          // Canvas initialisieren mit optionalem n-dim.KS
function clearImg(n,visa)               // Grafik löschen
function drawImg(name)                  // Eine JPG-Grafikdatei laden
function drawCoord2(kw,kb)              // Koordinatensystem zeichnen (2-dim.)
function drawCoord3(kw,kb)              // Koordinatensystem zeichnen (3-dim.)
function drawPoint(tex,P9)              // Punkt P9 zeichnen und mit tex beschriften in schwarz
function killPoint(tex,P9)              // Punkt P9 zeichnen und mit tex beschriften in weiß
function drawPosText(x,y,tex)           // Text positioniert ausgeben (in Weltkoordinaten)
function drawText(y,tex)                // Text positioniert ausgeben (in Bildkoordinaten)
function drawWidth(f)                   // Liniendicke f
function drawColor(f)                   // Linienfarbe f
function fillColor(f)                   // Füllfarbe f
function drawCircle(M9,r9)              // Kreis zeichnen
function fillCircle(M9,r9)              // Kreis füllen
function fillArc(M9,r9,w1,w2)           // Kreissektor füllen
function drawLine(R9,S9)                // Strecke durch die Punkte R9 und S9 zeichnen
function drawLine3(W7,W8,W9)            // gefülltes Dreieck [W7,W8,W9,W7] zeichnen
function drawLine4(W6,W7,W8,W9)         // gefülltes Vierieck [W6,W7,W8,W9,W6] zeichnen
function drawLine1(R9,S9)               // Gerade durch die Punkte R9 und S9 zeichnen
function funEll(x,a,b)                  // Funktionswert von Ellipse [a,b] mit Argument x
function drawEll(M9,a,b,r9)             // Ellipse [a,b] um M9(x/y) mit r9 = +/-1 zeichnen
function fillEll(M9,a,b,r9)             // Ellipse [a,b] um M9(x/y) mit r9 = +/-1 füllen

// Funktions-Plotter OHNE "parser.js" ------------------------------------------------

   var fpa, fpb, fpc, fpd;              // vier Funktionsparameter für 13 Funktionstypen
   var fselect = 1;                     // Funktions-Steuervariable (neu)
   var fselect_old = 0;                 // Funktions-Steuervariable (alt)

   function fun01(x) { y = fpa*x + fpb; return y; }
   function fun02(x) { y = fpa*x*x + fpb*x + fpc; return y; }
   function fun03(x) { y = fpa*x*x*x + fpb*x*x + fpc*x + fpd; return y; }
   function fun04(x) { y = 1/(x - fpa); return y; }
   function fun05(x) { y = 1/((x - fpa)*(x - fpa)); return y; }
   function fun06(x) { y = fpb * Math.sqrt(fpa*x); return y; }
   function fun07(x) { y = (fpb/fpa) * Math.sqrt(fpa*fpa - x*x); return y; }
   function fun08(x) { y = (fpb/fpa) * Math.sqrt(x*x - fpa*fpa); return y; }
   function fun09(x) { y = fpa * Math.exp(fpb*x); return y; }
   function fun10(x) { y = fpa * Math.log(fpb*x); return y; }
   function fun11(x) { y = fpa * Math.sin(fpb*x + fpc); return y; }
   function fun12(x) { y = fpa * Math.exp(fpb*x) * Math.sin(fpc*x); return y; }
   function fun13(x) { y = fpa * Math.exp(-x*x/2) / Math.sqrt(2*Math.PI); return y; }

   function selectParam()               // Erzeugung der Funktions-Parameter
   function drawFunk(fselect)           // ausgewählte Funktionen zeichnen

// Funktions-Plotter MIT "parser.js" -------------------------------------------------

   function drawFun(kw,kb,s)
    // kw = Halbbreite des Koordinatensystems, kb = Canvas-Bildbreite
    // Funktion fun(s,x) zeichnen, benötigt "parser.js"
    // s = Funktionsterm, x = Argument

   function drawFun1(kw,kb,xa,xb,s)
    // kw = Halbbreite des Koordinatensystems, kb = Canvas-Bildbreite
    // Funktion fun(s,x) zeichnen, zwischen xa und xb, benötigt "parser.js"
    // s = Funktionsterm, x = Argument, benötigt "parser.js"

   function fillFun1(kw,kb,xa,xb,s)
    // kw = Halbbreite des Koordinatensystems, kb = Canvas-Bildbreite
    // Funktion fun(s,x) mit Farbe füllen, zwischen xa und xb, benötigt "parser.js"
    // s = Funktionsterm, x = Argument
```

• Verzeichnis aller „Mathematik“-Bibliotheks-Funktionen

```
var info, info1, info2, info3, info4;   // Fünf Stringvariable
var error = -999;                       // Numerische Error-Variable
var n = 3;                              // Numerische Dimensions-Variable
var L1xy;                               // Erster Lösungsvektor (n-Array)
var L2xy;                               // Zweiter Lösungsvektor (n-Array)
var mat;                                // mehrdimensionale (n)x(n+1)- Matrix

function myTrim(x)                      // Führende und folgende Blanks von x entfernen
function round2(x)                      // x auf 2 Dezimalen runden
function round4(x)                      // x auf 4 Dezimalen runden
function deg(x)                         // Umwandlung von Bogenmaß in Gradmaß
function rad(x)                         // Umwandlung von Gradmaß in Bogenmaß
function Max6(a,b,c,d,e,f)              // Maximum von 6 Zahlen
function Min6(a,b,c,d,e,f)              // Minimum von 6 Zahlen

function TPoint(tx, ty)                 // Konstruktor für das Punkt-Objekt (2-dim.)
function TPoint3(tx, ty, tz)            // Konstruktor für das Punkt-Objekt (3-dim.)

function Flaeche1(a,b,c)                // Fläche eines Dreiecks
function Winkel1(a,b,c)                 // Winkel w bei Eckpunkt B
function ggt(a,b)                       // größter gemeinsamer Teiler von a und b
function kgv(a,b)                       // kleinstes gemeinsames Vielfache von a und b
function Primfakt(x)                    // Primfaktoren einer Zahl x

function differ(u,x)
  // Erste Ableitung, u = Funktionsterm f(x), x = Argument

function differ2(u,x)
  // Zweite Ableitung, u = Funktionsterm f(x), x = Argument

function tangente(u,Q)
  // Tangente y = k*x + d in Punkt Q mit u als f(x): k = returnWert.x, d = returnWert.y

function nullst(u,a,b)
   // Nullstelle zwischen a und b (mit u als f(x))

function extrema(u,a,b)
   // Extremstelle zwischen a und b (mit u als f(x))

function wendpun(u,a,b)
   // Wendestelle zwischen a und b (mit u als f(x))

function integ(u,a,b)
   // bestimmtes Integral von a bis b (mit u als f(x))

function integbog(u,a,b)
   // Bogenlänge einer Kurve von a bis b (mit u als f(x)), Bogenlänge = returnWert

function integbogX(u,a,b)
   // x-Wert des Bogenschwerpunktes (S) einer Kurve von a bis b (mit u als f(x))
   // xS = returnWert / Bogenlänge

function integbogY(u,a,b)
   // y-Wert des Bogenschwerpunktes (S) einer Kurve von a bis b (mit u als f(x))
   // yS = returnWert / Bogenlänge

function integflaX(u,a,b)
   // x-Wert des Flächenschwerpunktes (S) einer Kurve von a bis b (mit u als f(x))
   // xS = returnWert / Fläche

function integflaY(u,a,b)
   // y-Wert des Flächenschwerpunktes (S) einer Kurve von a bis b (mit u als f(x))
   // yS = returnWert / Fläche

function integfla(u,a,b)
   // Oberfläche eines Drehkörpers von a bis b (mit u als f(x))
   // Oberfläche = returnWert

function integvol(u,a,b)
   // Volumen eines Drehkörpers von a bis b (mit u als f(x))
   // Volumen = returnWert

function integvolX(u,a,b)
   // x-Wert des Drehkörperschwerpunkt(S) einer Kurve von a bis b (mit u als f(x))
   // xS = returnWert / Volumen
```

```
function GER(P9,Q9)
   // Gerade durch die Punkte P9 und Q9, Ergebnisse in L1xy
   // Senkrechte Gerade bei "-999"
   // wenn k = ~, dann L1xy[0] = 1, L1xy[1] = x, L1xy[2] = -999
   // wenn k <> ~, dann L1xy[0] = 0, L1xy[1] = k, L1xy[2] = d

function SGG(A9,B9,C9,D9)
   // Schnittpunkt der Geraden g(A9,B9) und h(C9,D9)
   // wenn kg = ~, dann L1xy[0] = 1, L1xy[1] = x, L1xy[2] = -999
   // wenn kg <> ~, dann L1xy[0] = 0, L1xy[1] = k, L1xy[2] = d
   // wenn kh = ~, dann L2xy[0] = 1, L2xy[1] = x, L2xy[2] = -999
   // wenn kh <> ~, dann L2xy[0] = 0, L2xy[1] = k, L2xy[2] = d
   // globale Zusatzinformationen in info1, info2 und info3

function SKG(M9,r9,A9,B9)
   // Schnittpunkte der Geraden g(A9,B9) mit dem Kreis k(M9,r9)
   // Erster Schnittpunkt: (L1xy[1] / L2xy[1])
   // Zweiter Schnittpunkt: (L1xy[2] / L2xy[2])
   // KEINE Schnittpunkte bei L1xy[0] = 3
   // Geradengleichung in "info1"

function SKK(M11,r11,M22,r22)
  // Schnitt von Kreis und Kreis
  // Erster Schnittpunkt  S1(L1xy[1]/L2xy[1])
  // Zweiter Schnittpunkt S2(L1xy[2]/L2xy[2])
  // Fehlererkennung mit "-999"

function SSM(A,B)
   // Streckensymmetrale y = k*x+d von AB
   // mit k = L1xy[1] und d = L1xy[2]
   // und Streckenmittelpunkt L2xy[1] = M9.x und L2xy[2] = M9.y
   // Senkrechte Gerade bei "-999"

function WSM(P,Q,R)
   // Winkelsymmetrale y = k*x + d von w(PQR)
   // L1xy[1] = k und L1xy[2] = d
   // Senkrechte Gerade bei "-999"

function LOT(C,A,B)
   // Das Lot y = k*x + d von C auf AB
   // mit k = L1xy[1] und d = L1xy[2]
   // Lotfußpunkt C mit C.x = L2xy[1] und C.y = L2xy[2]
   // Senkrechte Gerade bei "-999"

function deta(b,n)
  // Berechnung der Determinante
  // b bzw. a = Koeffizienten-Matrix "mat[i][j]", n = Anzahl

function linglei(b,n,z)
  // Lösung eines linearen Gleichungsystems in n Variablen
  // b bzw. a = Koeffizienten-Matrix "mat[i][j]"
  // z bzw. x = lösungs-Vektor "L1xy[i]"

function quagl(a,b,c)
  // Lösungen einer quadratischen gleichung
  // Lösungsart L1xy[0], ( 1 = eine, 2 = zwei, 3 = keine )
  // Lösungen L1xy[1] und L1xy[2]

function vekmal(i,A)                  // Multiplikation eines Vektors mit einer Zahl (2D)
function veksum(A,B)                  // Addition zweier Vektoren (2D)
function veklin(s,t,A,B)              // Linearkombination zweier Vektoren (2D)
function Shap(P,Q)                    // Streckenhalbierungspunkt (2D)
function Spunkt(P,Q,R)                // Schwerpunkt (2D)
function Slen(P,Q)                    // Distanz zweier Punkte (2D)
function Umfang(P,Q,R)                // Umfang eines Dreiecks (2D)
function Flaeche(P,Q,R)               // Fläche eines Dreiecks (2D)
function Winkel(P,Q,R)                // Winkel w bei Punkt Q (2D)
function vekmul(A,B)                  // Skalarprodukt zweier Vektoren (2D)

function vekmal3(n,P3)                // Multiplikation eines Vektors mit einer Zahl (3D)
function veksum3(P3,Q3)               // Addition zweier Vektoren (3D)
function veklin3(s,t,P3,Q3)           // Linearkombination zweier Vektoren (3D)
function Shap3(P,Q)                   // Streckenhalbierungspunkt (3D)
function Spunkt3(P,Q,R)               // Schwerpunkt (3D)
function Slen3(P3,Q3)                 // Distanz zweier Punkte (3D)
function Umfang3(P3,Q3,R3)            // Umfang eines Dreiecks (3D)
function Flaeche3(P3,Q3,R3)           // Fläche eines Dreiecks (3D)
function Winkel3(P3,Q3,R3)            // Winkel w bei Punkt Q3 (3D)
function vekmul3(P3,Q3)               // Skalarprodukt zweier Vektoren (3D)
function vekkreuz3(P3,Q3)             // Kreuzprodukt zweier Vektoren (3D)
function vekspat3(P3,Q3,R3)           // Spatprodukt zweier Vektoren (3D)
```

● Verzeichnis aller „Parser“-Bibliotheks-Funktionen

```
function fun(s,x)     // Funktionterm "s" in nur EINER Variablen x, returnWert = Funktionswert
function formel(s)    // Formel "s" in 26 Variablen a,b,c,...,y,z, returnWert = Funktionswert
```

Die Funktion „***fun(s,x)“*** dient der Programmierung von einfachen mathematischen Funktionen, wo der erste Parameter der Funktionsterm ist und der zweite Parameter das Funktionsargument.

Die Funktion ***„formel(s)“*** ermöglicht die Eingabe einer mathematischen Formel ***(s)***, welche die 26 Variablen ***a, b, c, . . ., x, y, z*** enthalten kann. Zurückgeliefert wird entweder eine ***relle Zahl*** oder bei fehlerhafter Eingabe „***NaN***" (Not a Number) bzw. "***Infinity***" (Unendlich).

Zulässige **Operatoren** sind: (,), +, -, *, /, ^, sqrt, % (=Rest), fac (=Faktorielle), round, rd2, rd4, floor, abs, exp, log, lg, deg, rad, sin, cos, tan, asin, acos, atan, atan2(y,x) (=Steigungswinkel), random(x) (=Zufallszahlen in [0,x)), PI (=Ludolfsche Zahl) und E (=Eulersche Zahl).

Beispiele: a % b, sin(rad(x)), deg(asin(x)), deg(atan2(a,b)), sqrt(x), log(x), exp(x), ……. Grundsätzlich werden alle Winkelfunktionen im Radianten-Maß berechnet. „deg(x)“ wandelt Radianten in Grade um. Hingegen wandelt „rad(x)“ Grade in Radianten um.

Hinweis: Viele Übungsprogramme des Lehrbuchs „***jspau.pdf***“ wurden schon vor einigen Jahren mit der HTML-Version 4.x erstellt. Daher wurden dort HTML-Tags verwendet, welche in neueren HTML-Versionen zwar als „obsolet“ bezeichnet sind, aber weiterhin noch unterstützt werden, beispielsweise die Befehle ***<font>*** und *** .***

ENDE