

Werner Geers

Berufliche Informatik

ISBN:978-3-427-60080-0

Bestellnr.:60080



Zusatzinformationen

1	Grafik mit PHP	2
1.1	Vorbemerkungen	2
1.1.1	Grundlegende Aussagen zur Darstellung von Grafiken auf dem Bildschirm.....	2
1.1.2	Erstellung einer Grafikarbeitsfläche mit einer Hintergrundfarbe	3
1.1.3	Speicherung einer Grafik	4
1.1.4	Text.....	5
1.1.5	Linien	6
1.1.6	Pixel.....	7
1.1.7	Kreise und Ellipsen	8
1.1.8	Recktecke	11
1.1.9	Polygone (Vielecke).....	12

Dieses Skript darf im Unterricht in Verbindung mit dem oben angegebenen Lehrbuch oder einem anderen Lehrbuch desselben Schulbuchautors eingesetzt werden.

1 Grafik mit PHP

1.1 Vorbemerkungen

Ein Bild sagt mehr als tausend Worte. Diese Aussage lässt sich gerade im Hinblick auf die Darstellung von mathematischen, wirtschaftlichen usw. Sachverhalten untermauern. Die Wiedergabe der Werte einer Funktionsgleichung in Form eines Funktionsgraphen ist weitaus aussagekräftiger als die Wiedergabe der Werte in Zahlenform. Beides kann sich jedoch auch sinnvoll ergänzen.

Auf den folgenden Seiten wird die Erstellung von Grafiken nacheinander sachlogisch dargestellt. Zunächst wird der Aufbau des Grafikbildschirms erklärt, danach wird eine Grafik mit einem Hintergrund erstellt. Das Einfügen von Texten, Linien usw. wird anschließend erklärt.

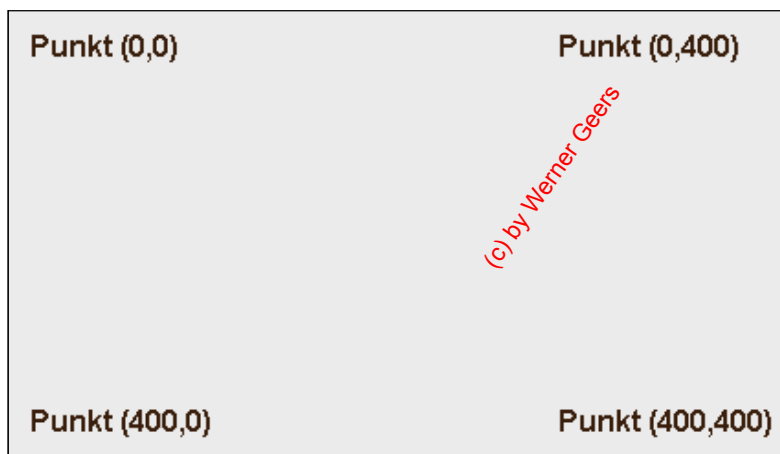
Größere Projekte runden die Arbeit mit Grafikbefehlen ab. Mit Hilfe eines erstellten Skripts ist beispielsweise die Darstellung von Funktionsgraphen möglich. Geschäftsgrafiken in Form von Kreis-, Säulen-, Kegel-, Punkt- und Liniendiagrammen können einen wirtschaftlichen und sonstigen Sachverhalt besonders gut verdeutlichen. Die einzelnen Listings werden durch Kommentare verdeutlicht.

Mit der Erstellung von Grafiken werden viele bisher bereits benötigte PHP-Befehle eingesetzt. Die Listings sind daher hervorragend geeignet, bereits erworbenes Wissen zu festigen.

1.1.1 Grundlegende Aussagen zur Darstellung von Grafiken auf dem Bildschirm

Die Größe der Fläche einer Grafik kann auf dem Bildschirm bestimmt werden. Dabei ist zu beachten, dass der Punkt (0,0) in der linken oberen Ecke ist. Entsprechend ist beispielsweise der rechte untere Punkt (im Beispiel (400,400)) der maximal anzusprechende Punkt.

Die folgende Darstellung veranschaulicht den Sachverhalt:



Die Lage der hat Auswirkungen auf die Programmierung einer Grafik. Soll beispielsweise ein Funktionsgraph dargestellt werden, muss der Funktionsgraph von unten links ausgehend dargestellt werden. Die Skalierung einer Y-Achse wird beispielsweise ebenfalls von unten nach oben vorgenommen.

Bei der Erstellung der Grafiken werden viele bereits bekannte Elemente genutzt. So ist beispielsweise der FOR-Schleife zum Setzen der einzelnen Punkte eines Funktionsgraphen notwendig. Außerdem sollen vor allem Beispiele genutzt werden, deren Ergebnisse bereits rechnerisch ermittelt wurden.

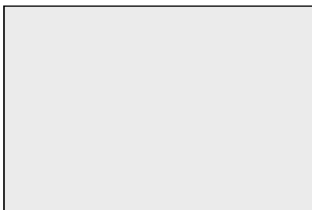
1.1.2 Erstellung einer Graficarbeitsfläche mit einer Hintergrundfarbe

Im ersten Schritt soll eine Hintergrundfläche mit einer Hintergrundfarbe erstellt werden und Grundlagen für die Abspeicherung einer Grafik gelegt werden.

PHP/Beispiel/Beschreibung
<code>header("Content-type: image/jpg");</code>
Die Grafikbibliothek wird eingebunden. Dies ist bei einigen Befehlen notwendig, jedoch nicht immer. Daher sollte sie grundsätzlich eingebunden.
<code>resource imagecreate (int x_size, int y_size)</code>
<code>\$Hintergrund = imagecreate(500,500)</code>
Eine Arbeitsfläche für ein Bild mit der Breite (<code>x_size</code>) und der Höhe (<code>y_size</code>) wird erstellt.
<code>int imagecolorallocate (int im, int red, int green, int blue)</code>
<code>\$Hintergrundfarbe = imagecolorallocate (\$Hintergrund,200,200,200);</code>
Die Hintergrundfarbe für den definierten Hintergrund wird festgelegt.
<code>int imagefill (int im, int x, int y, int col)</code>
<code>imagefill(\$Hintergrund,0,0,\$Hintergrundfarbe);</code>
Eine Fläche wird ab einem festgelegten Punkt mit einer Hintergrundfarbe ausgefüllt.
<code>int imagejpeg (resource \$im [, string \$filename [, int \$quality]])</code>
<code>imagejpeg(\$Hintergrund);</code>
Ein zuvor mit dem Befehl <code>imagecreate</code> erzeugtes Bild wird im Format <code>jpeg</code> an den Browser gesandt. Es können auch Bilder in den Formaten <code>gif</code> und <code>png</code> erzeugt werden.

Grafik_Hintergrund.php	htdocs/Grafik
<pre> 1 <?php 2 header("Content-type: image/jpeg"); 3 // Grafikfläche und Hintergrundfarbe 4 \$Hintergrund = imagecreate(300,300); 5 \$Hintergrundfarbe = imagecolorallocate(\$Hintergrund,235,235,235); 6 imagefill(\$Hintergrund,45,45,\$Hintergrundfarbe); 7 imagejpeg(\$Hintergrund); 8 ?></pre>	(c) by Werner Geers

Anzeige im Browser – Ergebnis



1.1.3 Speicherung einer Grafik

Ein erstelltes Bild kann sofort bei Ausführung des Skripts als Grafik gespeichert werden.

<pre>int imagegif (int im [, string filename [, int quality]])</pre>
<pre>imagegif(\$Hintergrund, "Hintergrund.gif");</pre>
<p>Ein zuvor mit dem Befehl <i>imagecreate</i> erzeugtes Bild wird im Format <i>gif</i> an den Browser gesandt und unter dem angegebenen Namen im aktuellen Verzeichnis gespeichert.</p>
<pre>int imagejpeg (int im [, string filename [, int quality]])</pre>
<pre>imagejpeg(\$Hintergrund, "Hintergrund.jpeg");</pre>
<p>Ein zuvor mit dem Befehl <i>imagecreate</i> erzeugtes Bild wird im Format <i>jpeg</i> an den Browser gesandt und unter dem angegebenen Namen im aktuellen Verzeichnis gespeichert.</p>
<pre>int imagepng (int im [, string filename])</pre>
<pre>imagepng(\$Hintergrund, "Hintergrund.png");</pre>
<p>Ein zuvor mit dem Befehl <i>imagecreate</i> erzeugtes Bild wird im Format <i>png</i> an den Browser gesandt und unter dem angegebenen Namen im aktuellen Verzeichnis gespeichert.</p>
<pre>int imagegif (int im [, string filename [, int quality]])</pre>
<pre>imagegif(\$Hintergrund, "Hintergrund.gif");</pre>
<p>Ein zuvor mit dem Befehl <i>imagecreate</i> erzeugtes Bild wird im Format <i>gif</i> an den Browser gesandt und unter dem angegebenen Namen im aktuellen Verzeichnis gespeichert.</p>
<p>Hinweis: Aus rechtlichen Gründen kann es sein, dass das Format nicht unterstützt wird.</p>




Grafik_Hintergrund.php

htdocs/Grafik

```

1  <?php
2  header("Content-type: image/jpeg");
3  // Hintergrundgröße
4  $Hintergrund = imagecreate(500,500);
5  // Hintergrundfarbe
6  $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
7  // Bild im Format gif an den Browser senden
8  imagejpeg($Hintergrund);
9  // Erstellung eine Bildes im Format jpeg
10 imagejpeg($Hintergrund, "Hintergrund.jpeg", 100);
11 // Erstellung eine Bildes im Format png
12 imagepng($Hintergrund, "Hintergrund.png");
13 // Erstellung eine Bildes im Format gif
14 imagegif($Hintergrund, "Hintergrund.gif");
15 ?>
```

Anzeige im Explorer – Ergebnis

 Hintergrund.gif	1 KB	18.02.2006 11:36
 Hintergrund.jpeg	5 KB	18.02.2006 11:36
 Hintergrund.png	1 KB	18.02.2006 11:36

1.1.4 Text

Eine Grafik wird grundsätzlich erst dann aussagekräftig, wenn bestimmte Inhalte beschriftet werden. Im nachfolgenden Beispiel werden verschiedene Möglichkeiten gezeigt, einen Text in einer Grafik zu platzieren.

PHP/Beispiel/Beschreibung

array **imagefttext** (int im, int size, int angle, int x, int y, int col, string fontfile, string text)

```
imagefttext($Hintergrund, 25, 360, 150, 40, $Schriftfarbe, "Arial.ttf", "Funktion");
```

In die Grafik wird ein Text geschrieben. Es werden der Hintergrund, die Schriftanordnung (Schriftgröße (Pixel), Schriftausrichtung (Grad), Textausgabe von links, Textausgabe von oben), die Schriftfarbe, die Schriftart und der auszugebende Text bestimmt.

Grafik_Text.php

htdocs/Grafik

```

1 <?php
2 header("Content-type: image/jpeg");
3 $Hintergrund = imagecreate(500,500);
4 $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5 $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6 imagefttext($Hintergrund,25,360,150,40,$Schriftfarbe,"Arial.ttf","Funktion");
7 $Schriftfarbe = imagecolorallocate($Hintergrund,200,40,106);
8 imagefttext($Hintergrund,15,90,30,150,$Schriftfarbe,"Arial.ttf","Preis");
9 $Schriftfarbe = imagecolorallocate($Hintergrund,200,40,200);
10 imagefttext($Hintergrund,15,360,50,150,$Schriftfarbe,"Arial.ttf","Menge");
11 imagefttext($Hintergrund,15,270,450,80,$Schriftfarbe,"Arial.ttf","Legende");
12 imagefttext($Hintergrund,15,50,200,150,$Schriftfarbe,"Arial.ttf","Inhalt");
13 imagefttext($Hintergrund,15,300,300,90,$Schriftfarbe,"Arial.ttf","Inhalt");
14 imagegif($Hintergrund);
15 ?>
```

Anzeige im Browser – Ergebnis



1.1.5 Linien

Linien können beliebig von einem Anfangs- zu einem Endpunkt gezogen werden.

PHP/Beispiel/Beschreibung

```
int imageline ( int im, int x1, int y1, int x2, int y2, int col )
```

```
imageline($Hintergrund, $x_Punkt1, $y_Punkt1, $x_Punkt2, $y_Punkt1, $Linienfarbe1);
```

Eine Linie wird in einer bestimmten Farbe von einem Punkt zu einem anderen Punkt auf einem Hintergrund (Arbeitsfläche) gezogen.

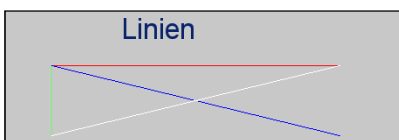
Grafik_Linien.php

htdocs/Grafik

```

1  <?
2  header("Content-type: image/jpeg");
3  $Hintergrund = imagecreate(500,500);
4  $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5  $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6  imagettftext($Hintergrund,25,360,150,40,$Schriftfarbe,"Arial.ttf","Linien");
7  $Linienfarbe1 = imagecolorallocate($Hintergrund,255,0,0);
8  $Linienfarbe2 = imagecolorallocate($Hintergrund,100,255,100);
9  $Linienfarbe3 = imagecolorallocate($Hintergrund,0,0,255);
10 $Linienfarbe4 = imagecolorallocate($Hintergrund,255,255,255);
11 // Ausgangswerte
12 $x_Punkt1 = 70;
13 $x_Punkt2 = 400;
14 $y_Punkt1 = 70;
15 $y_Punkt2 = 150;
16 // Horizontalen Linie
17 imageline($Hintergrund,$x_Punkt1,$y_Punkt1,$x_Punkt2,
           $y_Punkt1,$Linienfarbe1);
18 // Vertikale Linie
19 imageline($Hintergrund,$x_Punkt1,$y_Punkt1,$x_Punkt1,
           $y_Punkt2,$Linienfarbe2);
21 // Linien
22 imageline($Hintergrund,$x_Punkt1,$y_Punkt1,$x_Punkt2,
           $y_Punkt2,$Linienfarbe3);
23 imageline($Hintergrund,$x_Punkt2,$y_Punkt1,$x_Punkt1,
           $y_Punkt2,$Linienfarbe4);
24 imagegif($Hintergrund);
25 ?>
```

Anzeige im Browser – Ergebnis



1.1.6 Pixel

Die Darstellung von Linien kann ebenfalls mit Hilfe von Pixeln, die aneinander gereiht werden, vorgenommen. Von einem Anfangspunkt wird die Linie zu einem Endpunkt gezogen.

PHP/Beispiel/Beschreibung

int **imagesetpixel** (int im, int x, int y, int col)

imagesetpixel(\$Hintergrund, \$x_Achse1, \$x_Achse2, \$Linienfarbe);

Ein einzelner Pixel wird in einer bestimmten Farbe auf einem Hintergrund (Arbeitsfläche) geschrieben. Die Koordinaten des Pixels stehen in den Parametern x und y.

Grafik_Pixel.php

htdocs/Grafik

```

1  <?
2  header("Content-type: image/jpeg");
3  $Hintergrund = imagecreate(500,500);
4  $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5  $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6  imagettftext($Hintergrund, 25, 360, 200, 40, $Schriftfarbe, "Arial.ttf", "Pixel");
7  $Linienfarbe1 = imagecolorallocate($Hintergrund,212,105,50);
8  $Linienfarbe2 = imagecolorallocate($Hintergrund,0,0,255);
9  // Ausgangswerte einer horizontalen Linie
10 $x_Achse1 = 50;
11 $x_Achse2 = 100;
12 $Pixel = 400;
13 // Punkte einer horizontalen Linie
14 for($x=0;$x <=$Pixel;$x++){
15     imagesetpixel($Hintergrund, $x_Achse1, $x_Achse2, $Linienfarbe1);
16     $x_Achse1 = $x_Achse1 + 1;
17 }
18 // Ausgangswerte einer vertikalen Linie
19 $y_Achse1 = 50;
21 $y_Achse2 = 100;
22 //Punkte eine vertikalen Linie
23 for($y=0;$y <=$Pixel;$y++){
24     imagesetpixel($Hintergrund, $y_Achse2, $y_Achse1, $Linienfarbe2);
25     $y_Achse1 = $y_Achse1 + 5;
26 }
27 imagegif($Hintergrund);
28 ?>

```

Anzeige im Browser – Ergebnis



1.1.7 Kreise und Ellipsen

Ellipsen, Kreise und Teilbereiche davon lassen sich z. B. einsetzen, um Kreisdiagramme usw. zu erstellen. Nachfolgend werden die Grundlagen der Erstellung dieser Objekte erklärt.

PHP/Beispiel/Beschreibung
int imagearc (int im, int cx, int cy, int w, int h, int s, int e, int col)
imagearc (\$Hintergrund, X_Mittelpunkt, Y_Mittelpunkt, Hoehe, Breite, Grad1, Grad2, Linienfarbe);
Ein Kreis bzw. eine Ellipse werden von einem Mittelpunkt ausgehend in einer definierten Farbe erstellt. Die Höhe und Breite des Objektes kann bestimmt werden. Außerdem kann ein Teilobjekt durch Gradangaben definiert werden.
int imagefilltoborder (int im, int x, int y, int border, int col)
imagefilltoborder (\$Hintergrund, \$X_Mittelpunkt, \$Y_Mittelpunkt, \$Linienfarbe, \$Linienfarbe);
Ein vollständig geschlossenes Objekt, z. B. ein Kreis, wird mit einer Farbe ausgefüllt.

Grafik_Kreis_1.php

htdocs/Grafik

```

1  <?
2  header("Content-type: image/jpeg");
3  $Hintergrund = imagecreate(500,500);
4  $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5  $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6  imagettftext($Hintergrund, 25, 360, 100, 40,
7              $Schriftfarbe, "Arial.ttf", "Rechteck");
8  $Linienfarbe1 = imagecolorallocate($Hintergrund,212,105,50);
9  $Linienfarbe2 = imagecolorallocate($Hintergrund,0,0,255);
10 // Kreis
11 $X_Mittelpunkt1 = 120; $Y_Mittelpunkt1 = 120;
12 $Hoehe1 = 100; $Breite1 = 100;
13 $Grad1 = 0; $Grad2 = 360;
14 imagearc($Hintergrund,$X_Mittelpunkt1,$Y_Mittelpunkt1,
15          $Hoehe1,$Breite1,$Grad1,$Grad2,$Linienfarbe1);
16 // Ellipse
17 $X_Mittelpunkt2 = 320;
18 $Y_Mittelpunkt2 = 120;
19 $Hoehe2 = 160;
20 $Breite2 = 100;
21 imagearc($Hintergrund,$X_Mittelpunkt2,$Y_Mittelpunkt2,
22          $Hoehe2,$Breite2,$Grad1,$Grad2,$Linienfarbe2);
23 imagefilltoborder($Hintergrund,$X_Mittelpunkt2,$Y_Mittelpunkt2,
24                  $Linienfarbe2,$Linienfarbe2);
25 // Erstellung der Grafik
26 imagegif($Hintergrund);
27 ?>

```


Anzeige im Browser – Ergebnis



Selbstverständlich können auch Teilbereiche eines Kreises bzw. einer Ellipse in der gewünschten Form definiert werden.

Grafik_Kreis_2.php

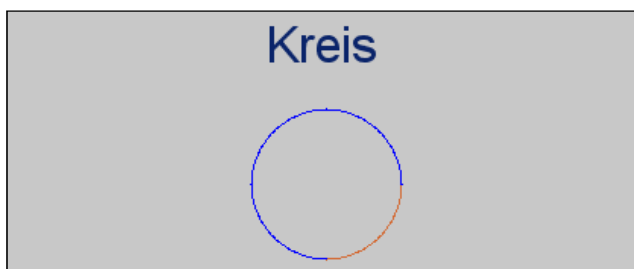
htdocs/Grafik

```

1 <?
2 header("Content-type: image/jpeg");
3 $Hintergrund = imagecreate(500,500);
4 $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5 $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6 imagettftext($Hintergrund, 25, 360, 210, 40,
7             $Schriftfarbe, "Arial.ttf", "Kreis");
8 $Linienfarbe1 = imagecolorallocate($Hintergrund,212,105,50);
9 $Linienfarbe2 = imagecolorallocate($Hintergrund,0,0,255);
10 $X_Mittelpunkt = 250; $Y_Mittelpunkt = 120;
11 // Definition des ersten Teil-Kreises
12 $Hoehe = 100; $Breite = 100; $Grad1 = 0; $Grad2 = 90;
13 imagearc($Hintergrund,$X_Mittelpunkt,$Y_Mittelpunkt,
14          $Hoehe,$Breite,$Grad1,$Grad2,$Linienfarbe1);
15 // Definition des zweiten Teil-Kreises
16 $Grad3 = 91; $Grad4 = 360;
17 imagearc($Hintergrund,$X_Mittelpunkt,$Y_Mittelpunkt,
18          $Hoehe,$Breite,$Grad3,$Grad4,$Linienfarbe2);
19 imagegif($Hintergrund);
20 ?>

```

Anzeige im Browser – Ergebnis



Während das Ausfüllen eines geschlossenen Kreises oder einer geschlossenen Ellipse, eines Rechtecks oder eines Vielecks keinerlei Probleme bereitet, ist das Ausfüllen von Teilbereichen eines Kreises oder einer Ellipse nicht mit den zur Verfügung stehenden Befehlen zum Ausfüllen einer Fläche nicht oder außerordentlich umständlich zu realisieren. Es bietet sich jedoch an, mit Hilfe einer *For-Schleife* verschiedene Kreise zu ziehen und damit zu einer ausgefüllten Fläche zu kommen.

Bedingt durch die Pixeldarstellung eines Bildschirms ist es durchaus möglich, dass ein Kreis nicht vollständig rund dargestellt und in einer Fläche u. U. einzelne Pixel nicht eingefärbt werden. Mit diesen relativ geringen Einschränkungen kann man sich sicherlich zufrieden geben, ansonsten muss man nach anderen Lösungen suchen.

Grafik_Kreis_3.php

htdocs/Grafik

```

1 <?
2 header("Content-type: image/jpeg");
3 $Hintergrund = imagecreate(500,500);
4 $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5 $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6 imagettftext($Hintergrund, 25, 360, 210, 40,
7             $Schriftfarbe, "Arial.ttf", "Kreis");
8 $Linienfarbe1 = imagecolorallocate($Hintergrund,212,105,50);
9 $Linienfarbe2 = imagecolorallocate($Hintergrund,0,0,255);
10 $X_Mittelpunkt = 250; $Y_Mittelpunkt = 120;
11 // Definition des ersten Teil-Kreises
12 $Hoehe = 100; $Breite = 100; $Grad1 = 0; $Grad2 = 90;
13 for($x=0; $x <= $Hoehe; $x = $x + 1){
14     imagearc($Hintergrund, $X_Mittelpunkt, $Y_Mittelpunkt,
15             $x, $x, $Grad1, $Grad2, $Linienfarbe1);
16 }
17 // Definition des zweiten Teil-Kreises
18 $Grad3 = 91; $Grad4 = 360;
19 for($x=0; $x <= $Hoehe; $x = $x + 10){
20     imagearc($Hintergrund, $X_Mittelpunkt, $Y_Mittelpunkt,
21             $x,$x, $Grad3, $Grad4,$Linienfarbe2);
22 }
23 imagegif($Hintergrund);
24 ?>
```

Anzeige im Browser – Ergebnis



1.1.8 Rechtecke

Die Darstellung von Säulen- oder Balkendiagrammen usw. ist es notwendig, dass Rechtecke definiert werden können. Diese Rechtecke sollten daneben ausgefüllt werden können.

PHP/Beispiel/Beschreibung
<pre>int imagerectangle (int im, int x1, int y1, int x2, int y2, int col)</pre>
<pre>imagerectangle(\$Hintergrund, \$X_Punkt1, \$Y_Punkt1, \$X_Punkt2, \$Y_Punkt2, \$Farbe);</pre>
Ein Rechteck wird von einem Ausgangspunkt bis zu einem Endpunkt gebildet.
<pre>int imagefilledrectangle (int im, int x1, int y1, int x2, int y2, int col)</pre>
<pre>imagefilledrectangle(\$Hintergrund, \$X_Punkt1, \$Y_Punkt1, \$X_Punkt2, \$Y_Punkt2, \$Farbe);</pre>
Ein mit einer Farbe gefülltes Rechteck wird auf der Arbeitsfläche ausgegeben.

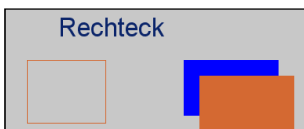
Grafik_Rechteck.php

htdocs/Grafik

```

1  <?
2  header("Content-type: image/jpeg");
3  $Hintergrund = imagecreate(500,500);
4  $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5  $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6  imagettftext($Hintergrund, 25, 360, 100, 40,
7              $Schriftfarbe, "Arial.ttf", "Rechteck");
8  $Linienfarbe1 = imagecolorallocate($Hintergrund,212,105,50);
9  $Linienfarbe2 = imagecolorallocate($Hintergrund,0,0,255);
10 // Rechteck ohne Füllung
11 $X_Punkt1 = 60; $Y_Punkt1 = 70;
12 imagerectangle($Hintergrund,$X_Punkt1, $Y_Punkt1,
13               $X_Punkt1 + 100 , $Y_Punkt1 + 80, $Linienfarbe1);
14 // Rechteck mit Füllung 1
15 $X_Punkt2 = 260; $Y_Punkt2 = 70;
16 imagefilledrectangle($Hintergrund, $X_Punkt2, $Y_Punkt2,
17                     $X_Punkt2 + 120 , $Y_Punkt2 + 70, $Linienfarbe2);
18 // Rechteck mit Füllung 2
19 $X_Punkt3 = 280; $Y_Punkt3 = 90;
20 imagefilledrectangle($Hintergrund,$X_Punkt3, $Y_Punkt3,
21                     $X_Punkt3 + 120 , $Y_Punkt3 + 70, $Linienfarbe1);
22 // Erstellung der Grafik
23 imagegif($Hintergrund);
24 ?>
```

Anzeige im Browser – Ergebnis



1.1.9 Polygone (Vielecke)

Benötigt man eine besondere grafische Darstellung, beispielsweise eine Pfeilspitze, kann dies durch ein Polygon dargestellt werden.

PHP/Beispiel/Beschreibung

int **imagepolygon** (int im, array points, int num_points, int col)

```
$Punkte = array(45,165,180,60,78,78);
imagepolygon($Hintergrund, $Punkte, 3 , $Farbe);
```

Die xy-Werte mehrerer Punkte werden in ein Array geschrieben. Danach werden der Hintergrund, die einzelnen xy-Werte, die Anzahl der Punkte und die Farbe definiert. Danach wird das nicht ausgefüllte Objekt ausgegeben.

int **imagefilledpolygon** (int im, array points, int num_points, int col)

```
$Punkte = array(245,65,320,160,378,76,420,60);
imagefilledpolygon($Hintergrund, $Punkte, 4 , $Farbe);
```

Die xy-Werte mehrerer Punkte werden in ein Array geschrieben. Danach werden der Hintergrund, die einzelnen xy-Werte, die Anzahl der Punkte und die Farbe definiert. Danach wird das gefüllte Objekt ausgegeben.

Grafik_Polygon.php

htdocs/Grafik

```
1 <?
2 header("Content-type: image/jpeg");
3 $Hintergrund = imagecreate(500,500);
4 $Hintergrundfarbe = imagecolorallocate($Hintergrund,200,200,200);
5 $Schriftfarbe = imagecolorallocate($Hintergrund,10,36,106);
6 imagettftext($Hintergrund, 25, 360, 150, 40,
7             $Schriftfarbe, "Arial.ttf", "Polygon");
8 $Linienfarbe1 = imagecolorallocate($Hintergrund,212,105,50);
9 $Linienfarbe2 = imagecolorallocate($Hintergrund,0,0,255);
10 // Polygon ohne Füllung
11 $Punkte = array(45,165,180,60,78,78);
12 imagepolygon($Hintergrund, $Punkte, 3 , $Linienfarbe1);
13 // Polygon mit Füllung
14 $Punkte = array(245,65,320,160,378,76,420,60);
15 imagefilledpolygon($Hintergrund, $Punkte, 4 , $Linienfarbe2);
16 // Erstellung der Grafik
17 imagegif($Hintergrund);
18 ?>
```

Anzeige im Browser – Ergebnis

