

Lektion 10: Cookies: Begrüßung und Web-Umfrage

Schön, dass Sie uns wieder besuchen! Auf wie vielen Webseiten hast du diesen oder einen ähnlichen Spruch wohl schon gelesen?

Und das Schärfste ... es stimmt!

Noch nach Wochen kennt ein Online-Shop die Produkte, die du beim vorigen Besuch in den Warenkorb gelegt (und nicht gekauft) hattest. Alles noch da als wäre nix gewesen.

Was steckt dahinter? Hexerei? Fauler Webzauber? Big Brother online, das neuartige Überwachungs- und Protokollprogramm des BND? Weder noch!

Es sind nichts weiter als Cookies!

Was sind Cookies?

Cookies sind wohl eine der am häufigsten missverstandenen Techniken! Daten von deiner Festplatte könnten ausspioniert werden, Cookies seien ein Einfallstor für Viren.

Nun, alles das ist falsch! Cookies sind nichts weiter als Textdateien, mit denen eine Webseite Infos auf deinem Rechner speichert.

Übersetze das Wort Cookie weniger als Keks sondern eher als „Krümel“ oder besser: „Informationskrümel“.

Eine Webseite speichert auf deinem Rechner, dass du sie besucht hast. Ein Web-Shop merkt sich durch dich, welche Produkte noch in deinem Warenkorb stecken.

Das Ganze wird nur bei Bedarf wieder ausgelesen. Von wem und von wo? Diese Daten können nur von der Seite wieder ausgelesen werden, die sie gesetzt hat. Oder, um genau zu sein, von HTML-Dokumenten aus dem gleichen Ordner bzw. HTML-Seiten aus Unterordnern. Aber im Normalfall nicht von einer anderen Domäne!

Cookies sind eine Privatsache zwischen der Webseite, deinem Browser und dir.

Wo liegen die Cookies?

Die Daten lagern auf deiner Festplatte. Der Internet Explorer legt Cookies z. B. in der Regel unter `C:\Windows\Cookies` ab. Beim Firefox wählst du **EXTRAS|EINSTELLUNGEN**, gehst in den Bereich **Datenschutz** und klickst auf die Schaltfläche **COOKIES ANZEIGEN** – schon siehst du sie.

Du kannst Cookies bedenkenlos einsehen und löschen.

Falls du nicht fündig wirst, fahnde einfach mit der Windows-Suchfunktion nach dem Stichwort `cookies`. Gerade bei Mehrbenutzer-Umgebungen musst du etwas nach den Cookies suchen, da sie hier in anderen Ordnern abgelegt werden. Orientiere dich am besten am Datum!

Die Anzahl der Cookies per Benutzer ist auf 300 begrenzt. Ein Cookie darf nicht größer als 4 KB sein. Eine Domain kann nicht mehr als 20 Kekse „backen“.

Sind Cookies ein Sicherheitsrisiko?

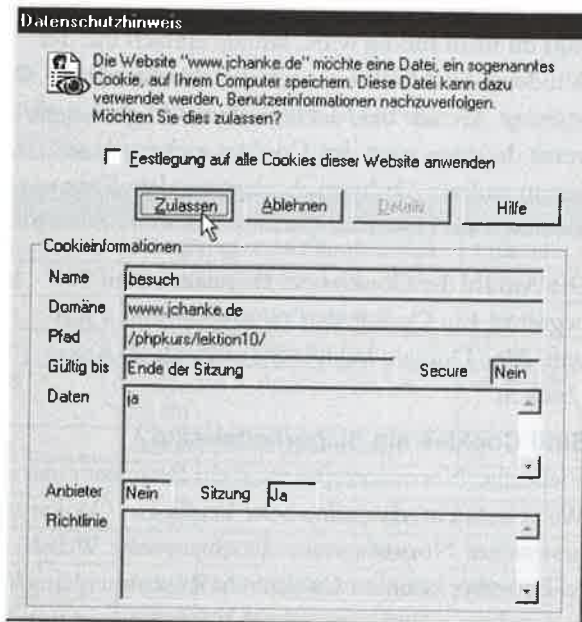
Vielleicht. Normalerweise weiß ein Betreiber einer Webseite nicht, dass seine Seite bei dir ein Cookie gesetzt hat. Normalerweise. Doch gewiefte Website-Betreiber könnten Cookies im Zusammenhang mit anderen von dir erhaschten Informationen nutzen, um Surf- und Nutzerprofile zu erstellen. Cookies werden also dann gefährlich, wenn du irgendwo zusätzlich noch deine Daten wie Telefonnummer, Anschrift oder E-Mail-Adresse hinterlässt.

Auch die in Verruf geratene Spyware *Windows XP Media Player* führt dazu, dass dich Microsoft beim Abspielen von DVDs oder CDs via www.windowsmedia.com mit ganz eindeutigen Keksen beglückt. Ob man dann irgendwie herausfinden kann, welche nicht originalen CDs/DVDs du gerade abspielst? Ich würde als Vorsichtsmaßnahme erstens CDs/DVDs nie im Online-Modus abspielen und zweitens (geht ab Internet Explorer 6) die Cookies von [windowsmedia.com](http://www.windowsmedia.com) generell blocken.

Einstellungen zu Cookies

Du möchtest Cookies von einer bestimmten Domain blocken? Wähle im IE 6 **EXTRAS/INTERNET-OPTIONEN**, gehe ins Register **DATENSCHUTZ**. Der Knopf **BEARBEITEN** erlaubt dir, Cookies von bestimmten Websites selektiv zu sperren.

Für deine Übungszwecke kannst du den Browser auch so einstellen, dass Cookies erst „auf Nachfrage“ (Eingabeaufforderung) gespeichert werden. Das ist dann ideal, wenn du testen möchtest, ob deine Info-Kekse überhaupt ankommen.



Die Cookie-Daten müssen erst bestätigt werden

Im Internet Explorer 6 wählst du dazu **EXTRAS/INTERNETOPTIONEN**, gehst ins Register **DATENSCHUTZ** und klickst auf **[ERWEITERT]**. Hebe die *automatische Cookiebehandlung* auf und wähle überall *Eingabeaufforderung*.

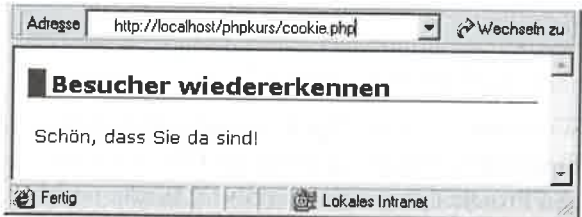
Im Internet Explorer 5.x wählst du dagegen **EXTRAS/INTERNETOPTIONEN** und gehst ins Register **SICHERHEIT**. Klicke auf **[STUFE ANPASSEN]** und nimm die entsprechenden Einstellungen im Bereich „Einstellungen“ unter „Cookies“ vor.

Hinweis: In meinen Tests funktionierten die Einstellungen beim IE nur dann, wenn ich die Seiten vorher auf den Webserver gespielt hatte.

Cookies setzen mit PHP

Nach diesen ganzen Vorbemerkungen bist du derjenige, der Kekse bäckt. Wir machen es so wie im JavaScript-Heft „Homepages für Fortgeschrittene“.

Wir wollen unsere Besucher persönlich begrüßen!



Beim ersten Aufruf wird das Cookie erst gesetzt

Du setzt auf der Seite des Surfers ein Cookie. Beim nächsten Besuch liest „du“ dieses wieder aus und der Surfer wird „persönlich“ begrüßt.



Nach einem Reload wird das Cookie ausgelesen

Unser Musterkeks soll "besuch"="ja" heißen. Das Setzen eines Cookies ist in PHP (wie fast alles) wieder mal revolutionär einfach.

Beachte: Cookies sind Strings.

Die Funktion setcookie()

Die Funktion `setcookie()` löst die Aufgabe für dich im Handumdrehen. Die Grundsyntax sieht folgendermaßen aus:

```
setcookie("Cookiename", "Cookiewert")
```

Wichtig ist, dass das Cookie im Header verschickt wird. Dafür muss die Funktion ganz zu Beginn aufgerufen werden, noch vor dem einleitenden HTML-Tag.

Merke: Das Cookie wird zuerst verschickt!

Alle Cookies werden automatisch in `$_COOKIE` gespeichert, der Cookiename wird zum Key.

Quelltext

Und nun fällt es dir sicher überhaupt nicht schwer, das korrespondierende Simpel-Skript zu durchschauen. Unser Musterkeks heißt "besuch" = "ja", demzufolge kannst du über \$_COOKIE['besuch'] auf den Cookie-Wert zurückgreifen. Hier die erste Version des Skriptes, siehe Datei cookie1.php im Ordner lektion10.

```
<?php
// Cookie wird gesetzt
setcookie("besuch","ja");
?>
<html>
<head>
<title>Ein Cookie setzen</title>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="../css/neu.css">
</head>
<body bgcolor="white">
<h1>Besucher wiedererkennen</h1>
<p>
<?php
// Ist das Cookie vorhanden und lautet der Wert ja?
if (isset($_COOKIE['besuch']) && $_COOKIE['besuch'] == "ja") {
    echo "Schön, dass Sie <b>wieder</b> da sind!\n";
} else {
    echo "Schön, dass Sie da sind!\n";
}
?>
</p>
</body>
</html>
```

Du merkst schon: PHP hat's so mit seinen Arrays. Und so verwundert es nicht, dass alle Cookies in dem eben erwähnten assoziativen Array namens \$_COOKIE gespeichert werden.

Interessant sind die Parallelen zur Formularauswertung, zu \$_POST und \$_GET. Auch bei Cookies gilt: Früher konntest du auf Cookies einfach über ihren Namen zugreifen. Der Wert des Cookies besuch stand sofort in \$besuch zur Verfügung. Neuerdings geht das nur noch über \$_COOKIE['Name_des_Cookies'], im Beispiel also über \$_COOKIE['besuch'] – zumindest wenn die Konfigurationseinstellung register_globals auf Off steht. Mehr zu dieser Problematik und zum Stichwort register_globals erfährst du in „PHP und MySQL auf der Homepage“.

Hinweis zum Nachschlagen: Das Kapitel zu assoziativen Arrays beginnt auf Seite 22. Wie Formulardaten als assoziatives Array behandelt werden können, habe ich ab Seite 38 erläutert.

Lebensdauergarantie für Cookies

Uuups, was ist denn da passiert? Nach einem Reload klappt alles super mit unserem Begrüßungs-Keks. Doch ein neuerlicher Aufruf des Browser zu einem späteren Zeitpunkt zeigt: Alles weg. Du wirst wieder so begrüßt, als wärest du zum ersten Mal auf dieser Seite.

Leidet dein Browser etwa unter Cookie-Alzheimer, dem „Online-Gedächtnisschwund“?

Mitnichten! Wir haben vergessen, dem Cookie eine „Lebensdauergarantie“ mit auf den Weg zu geben.

Bisher spricht man von einem temporären Cookie.

Ohne Lebensdauergarantie bleibt das Cookie für eine Sitzung aktiv, bestenfalls vielleicht noch ein paar Stunden länger. Wir wünschen uns jedoch, dass das Cookie 30 Tage gespeichert bleibt.

Ablaufdatum angeben

Die erweiterte Cookiesyntax mit Lebensdauergarantie sieht folgendermaßen aus:

```
setcookie("Cookiename", "Cookiewert", Ablaufdatum)
```

Wie ermitteln wir die Lebensdauer? Dazu brauchen wir zuerst die Anzahl der Sekunden, die seit Beginn der so genannten Unix-Epoche vergangen sind.

Die Unix-Epoche begann am 1.1.1970 um 0:00 Uhr.

Da sich dieser Wert ständig verändert, nutzen wir die Funktion `time()` für diesen Zweck. Mit `time()` ermittelst du die Anzahl der Sekunden seit Beginn der Unix-Epoche.

Doch wir wollen dem Cookie eine Lebensdauergarantie mit auf den Weg geben. Wir addieren die Zahl der Sekunden, die wir in die Zukunft denken.

Soll das Cookie 30 Tage aktiv sein? Dann müssen wir 2592000 Sekunden addieren. Wir können das auch übersichtlicher schreiben:

```
time()+86400*30
```

Das Skript anpassen

In unserem Beispiel mußt du also nichts weiter tun als die Zeile

```
setcookie("besuch", "ja");
```

wie folgt zu ändern:

```
setcookie("besuch", "ja", time() + 86400*30);
```

Das so angepasste Skript mit Lebensdauergarantie findest du unter dem Namen `cookie2.php`.

Wie löscht man ein Cookie?

Und wie entfernt man ein Cookie wieder? Beispielsweise, weil die Besucherin oder der Besucher seine Bestellung aus dem Warenkorb endlich aufgegeben hat?

Ganz einfach: Übermittle exakt das gleiche Cookie noch einmal. Sowohl Name als auch Wert müssen identisch sein. Gib diesmal jedoch ein Ablaufdatum an, welches in der Vergangenheit liegt.

Folgende Zeile löscht unser Beispielcookie:

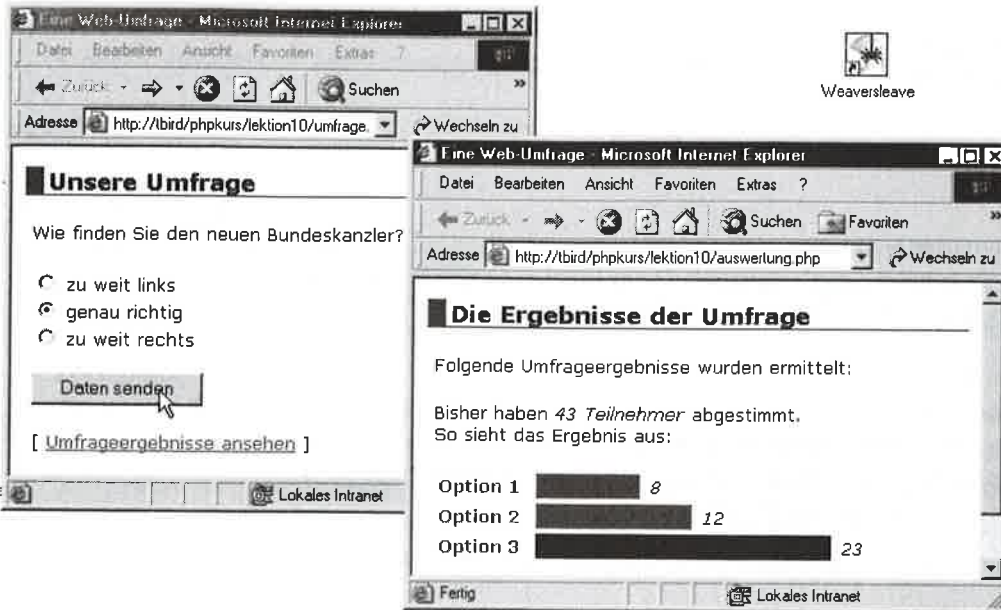
```
setcookie("besuch", "ja", time()-86400);
```

Du findest mein Musterskript in der Datei `cookieweg.php`. Beachte beim Ausprobieren, dass der Löschvorgang evtl. erst nach dem Schließen des Browsers abgeschlossen wird.

1, 2 oder 3? Web-Umfrage mit allen Schikanen!

Wer erinnert sich nicht gerne an die Kindersendungen von und mit ... Michael Schanze! Eins, zwei oder drei ... Du musst dich entscheiden. Ob du richtig bist oder nicht sagt dir gleich das Licht.

Wir begnügen uns an dieser Stelle nicht mit Reminiszenzen an die Kindheit. Hier geht es um Web-Umfragen, denn diese sind auch unter Erwachsenen populär. Außerdem gibt es kein richtig oder falsch, schließlich leben wir in einer Demokratie. Deshalb zeige ich dir an dieser Stelle, wie du selber solch eine Umfrage programmierst. Du kannst hinterher ganz demokratisch entscheiden, ob dir das Skript zusagt oder nicht. Betrachte es als Vorschlag, den du an deine Bedürfnisse anpassen kannst!



Eine Umfrage mit allen Schikanen: Verhindern von Mehrfachvotes und grafische Auswertung

Unsere Beispielumfrage erlaubt dem Besucher, sich zwischen einer von drei Wahlmöglichkeiten zu entscheiden. Im Beispiel erfrage ich die Meinung zum neuen Bundeskanzler. Oder zur neuen Bundeskanzlerin. Wer es wird, kann ich nicht wissen, denn Redaktionsschluss dieser Neuauflage war ausgerechnet wenige Tage vor der Bundestagswahl 2005!

Schluss mit der Politik und zurück zur Umfrage: Du kannst gerne andere Texte einsetzen, denn das Skript ist universal gehalten. Es lässt sich ganz einfach an (fast) jeden Wunsch anpassen. Außerdem habe ich mir ein paar weitere Extras ausgedacht. Erstens: Das Umfrageergebnis soll in einer simplen Textdatei mit kommagetrennten Werten gespeichert werden. Nix SQL-Datenbank, wie bei den meisten Musterskripten im Web. (Denke nach dem Hochladen ggf. an „chmod“, siehe S. 77!)

Erstelle dafür bitte eine Textdatei namens `ergebnis.txt`. Das ist unser Datenspeicher. Notiere hier die Startwerte 0,0,0 für „dreimal Null“. Als Trennzeichen dient das Komma!

Zweitens möchte ich Betrugsmöglichkeiten wie das mehrfache Abstimmen möglichst unterdrücken. Das lösen wir mit einem Cookie. Abstimmen darf nur, wer Cookies zulässt. Wer schon abgestimmt hat, bekommt einen weiteren Keks verpasst. Eine zweite Abstimmung ist dann nicht mehr möglich. (Dass ein Besucher natürlich mit einem anderen Browser noch einmal abstimmen kann oder Cookies einfach deaktiviert, müssen wir schlucken.) Drittens soll das Endergebnis als Säulendiagramm grafisch ausgewertet werden. Und zwar in prozentualer Verhältnismäßigkeit. Hier ist simple Mathematik im Spiel, du wirst schon sehen. Doch zuerst beginnen wir knallhart mit unserem Formular!

Das Skript im Überblick

Zuerst schauen wir uns die Datei(en) im Überblick an. Danach erkläre ich dir alles ganz in Ruhe. Wenn du keine Lust zum Abschreiben hast: Das Umfrage-Skript findest du unter dem Namen `umfrage.php`, die Auswertungs-Datei dagegen unter `auswertung.php`. Hinweis für Leser von Auflage 1 und 2: Bei diesem Skript handelt es sich um eine verbesserte Version mit etwas mehr Schummelsicherheit.

■ `umfrage.php`

Hier zeige ich dir zuerst das komplette Umfrage-Skript.

```
<?php
setcookie("check", "1");
if (isset($_POST['submit'])) {
    setcookie("voted", "1");
}
?>
<html>
<head>
<title>Eine Web-Umfrage</title>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="../css/neu.css">
</head>
<body bgcolor="white">
<h1>Unsere Umfrage</h1>
<p>Wie finden Sie den neuen Bundeskanzler?</p>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<input type="radio" name="antwort" value="0">
zu weit links<br>
<input type="radio" name="antwort" value="1">
genau richtig<br>
<input type="radio" name="antwort" value="2">
zu weit rechts
<br><br>
<?php
if (empty($_POST['submit']) && empty($_COOKIE['voted'])) {
// Submit-Button nur anzeigen, wenn Formular noch nicht
// abgeschickt und noch nicht gevoted!
?>
<input name="submit" type="submit" value="Daten senden">
<?php
} else {
    echo "<p>Danke für die Übermittlung der Daten!</p>\n";
// Formular abgeschickt? Cookies an? Aber bisher noch nicht gevoted?
if (isset($_POST['antwort']) && isset($_COOKIE['check']) && empty($_COOKIE['voted'])) {
// Dateiname in Variable speichern
$datei = "ergebnis.txt";
$zeiger = fopen($datei, "r+");
$umfrage = fread($zeiger, filesize($datei));
// String aus Datei mit Komma als Trenner in Array zerlegen
$arr_umfrage = explode(",", $umfrage); // explode wandelt String in Array um
// Welcher Wert wurde im Formular ausgewählt?
// Diese Position wird um 1 erhöht!
$antwort = $_POST['antwort'];
$arr_umfrage[$antwort]++;
```

```

// String neu zusammensetzen
$umfrage = implode(" ", $arr_umfrage); // implode verbindet Array-Elemente zu String
rewind($zeiger);
// neuen String in Datei schreiben
fputs($zeiger, $umfrage);
fclose($zeiger);
}
}
?>
</form>
<p>
[ <a href="auswertung.php" target="_blank">Umfrageergebnisse ansehen</a> ]
</p>
</body>
</html>

```

■ Cookie setzen

Ganz zu Anfang werden zwei Cookies gesetzt. Das eine namens *check* in jedem Fall. Das zweite namens *voted* jedoch – dank der *if*-Abfrage – erst, nachdem der Formularinhalt abgeschickt wurde! Beide Cookies sollen nur für die aktuelle Sitzung gelten, sie bekommen keine Lebensdauergarantie mit auf den Weg.

■ Formularteil

Der Formularteil des Skriptes leuchtet dir sicher längst ein. Wir haben das Thema bisher oft genug besprochen. Ich erzeuge eine Gruppe von Radioknöpfen mit dem Namen *antwort*. Daraus wird die Variable `$_POST['antwort']`, die den Wert 0, 1 oder 2 gespeichert hat. Warum ich mir diese Werte ausgedacht habe, klären wir gleich. Auch das dynamische Einblenden des Submit-Buttons ist dir sicher klar. Dieser soll nur dann erscheinen, wenn der Formularinhalt bisher noch nicht abgeschickt wurde, also wenn der Nutzer noch gar nicht *gevoted* hat!

Der Submit-Button bekommt absichtlich den Namen *submit* verpasst. Denn nur nach Klick auf diesen Knopf entsteht eine Variable namens `$_POST['submit']`. Und deren Vorhandensein fragen wir nachher ab. So wird das *voted*-Cookie z.B. erst dann gesetzt, wenn der Formularinhalt abgeschickt wurde.

Die Funktionen explode() und implode()

In unserem Formular wirkt zum ersten Mal eine für uns noch neue und vor allem supergeniale Funktion namens `explode()`. Diese Funktion zerteilt einen String automatisch anhand eines Trennzeichens und verwandelt ihn in ein Array, in unsere allseits beliebte und vielzitierte Werteliste. Die Werte zwischen den Trennzeichen werden so zu den Elementen des Arrays. Das Gegenteil macht `implode()`, diese Funktion zerlegt das Array, weist wieder Trennzeichen zu und gibt einen String zurück.

So geschieht es dann, dass unsere unschuldigen Zahlen (bisher 0,0,0) als erster, zweiter oder dritter Wert eines Arrays angesprochen werden. Oder vielmehr als nullter, erster und zweiter, um genau zu sein. Denn wir erinnern uns dunkel, dass das Indexieren bei Arrays stets bei Null beginnt!

Genau das ist beabsichtigt! Ich erzeuge ein Array namens `$arr_umfrage` und kann die einzelnen Positionen über `$umfrage[0]`, `$umfrage[1]` und `$umfrage[2]` ansprechen und mit Leben füllen.

Deshalb gebe ich im Formular dem Feld namens *antwort* die Werte 0 (für die erste Option), 1 (für die zweite Option) und 2 (für die dritte Option) mit auf den Weg!

Die Zeile `$umfrage[$antwort]++`; erhöht dann exakt den Wert um eins, der mit der Wahl des Benutzers korrespondiert! Zum Schluss packe ich das erhöhte und die nicht-erhöhten Teil-Elemente des Strings zu-

sammen, verfrachte sie wieder in die Variablen \$umfrage und lege alles zurück in die Datei ergebnis.txt. Eigentlich ganz simpel, oder?

Grafische Auswertung in der auswertung.php

Und was passiert im Auswerte-Skript? Also in der Datei, die der Besucher aufruft, wenn sie oder er die Umfrageergebnisse sehen möchte?

Das hier:

```
<html>
<head>
<title>Eine Web-Umfrage</title>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="../css/neu.css">
</head>
<body bgcolor="white">
<h1>Die Ergebnisse der Umfrage</h1>
<p>Folgende Umfrageergebnisse wurden ermittelt:</p>
<?php
$datei = "ergebnis.txt";
$zeiger = fopen($datei,"r");
$umfrage = fread($zeiger,filesize($datei));
fclose($zeiger);
// String zerlegen, Array entsteht
$umfrage = explode(",", $umfrage);
// Gesamtzahl aller Wahlvorgänge
$gesamt = $umfrage[0] + $umfrage[1] + $umfrage[2];
// Höchstlänge der Balken angeben
$laenge = 400;
// Anteil von Balken 1 (Indexwert 0!)
$laenge0 = $umfrage[0] * $laenge / $gesamt;
// Anteil von Balken 2 (Indexwert 1!)
$laenge1 = $umfrage[1] * $laenge / $gesamt;
// Anteil von Balken 3 (Indexwert 2!)
$laenge2 = $umfrage[2] * $laenge / $gesamt;
// Werte auf ganze Zahlen runden
$laenge0 = round($laenge0);
$laenge1 = round($laenge1);
$laenge2 = round($laenge2);
// Ausgabe nur für Testzwecke:
// echo "$laenge0 $laenge1 $laenge2";
?>
<p>Bisher haben <i><?php echo $gesamt;?> Teilnehmer</i> abgestimmt.<br>
So sieht das Ergebnis aus:</p>
<table border="0">
<tr>
<td><b>Option 1</b></td>
<td>&nbsp;</td><td width="<?php echo $laenge0; ?>" bgcolor="red">&nbsp;</td>
<td>&nbsp;<i><?php echo $umfrage[0];?></i></td>
</tr></table>
<table border="0">
<tr>
<td><b>Option 2</b></td>
```

```

<td>&nbsp;</td><td width="<?php echo $laenge1; ?>" bgcolor="green">&nbsp;</td>
<td>&nbsp;<i><?php echo $umfrage[1]; ?></i></td>
</tr></table>
<table border="0">
<tr>
<td><b>Option 3</b></td>
<td>&nbsp;</td><td width="<?php echo $laenge2; ?>" bgcolor="black">&nbsp;</td>
<td>&nbsp;<i><?php echo $umfrage[2]; ?></i></td>
</tr></table>
</body>
</html>

```

Das Skript im Überblick

Zuerst öffnen wir ganz gemütlich unsere Datei `ergebnis.txt`. Diesmal genügt der Nur-Lese-Modus, also schreiben wir `$zeiger = fopen($datei, "r");` Mit `fread()` lesen wir danach den Inhalt der gesamten Datei aus und speichern den String in der Variablen `$umfrage`. Mehr brauchen wir nicht, deshalb können wie die Datei gleich wieder mit `fclose()` schließen!

Nun wird unser neues Sternchen `explode()` aktiv und zerlegt unsere drei kommasetrennten Zahlen in ein Array. Zuerst werden alle drei Zahlen zusammengezählt, der Wert wird in der Variablen `$gesamt` abgelegt. Damit wissen wir schon einmal, wie viele Vote-Vorgänge es insgesamt gegeben hat. Als nächstes kümmere ich mich schon einmal um das Diagramm. Zumindest die Höchstlänge eines Balkens lege ich fest: Ich habe mich für 400 Pixel entschieden und speichere diesen Wert in der Variablen `$laenge`.

■ Wie lang sind die Balken? Verhältnisgleichung!

Nun besteht die Schwierigkeit nicht darin, Balken zu zeichnen. Das habe ich mit verschiedenen eingefärbten Tabellenzellen gelöst. Ich gebe dem `width`-Attribut der entsprechenden Zelle dynamisch die gewünschte Länge mit auf den Weg. Nicht gerade elegant, aber es funktioniert. Doch wie ermittle ich die Länge? Schließlich sollen die Balken nicht von 0 bis unendlich Pixel wachsen, dafür reicht irgendwann der Bildschirm nicht mehr aus. Die Länge der Balken soll lediglich im Verhältnis zueinander stimmen.

Ich arbeite mit einer Verhältnisgleichung, einem Dreisatz! Ich setze die Votes für jede einzelne Option zur Gesamtlänge (hier 400) ins Verhältnis.

So kann ich die Länge für den ersten Balken (Variable `$laenge0`) folgendermaßen berechnen:

```
$laenge0 = $umfrage[0] * $laenge / $gesamt;
```

Der daraus ermittelte Wert wird gerundet (Denn ich brauche Ganzzahlen für die Pixelwerte) und den Tabellenzellen als `width`-Angabe mit auf den Weg gegeben.

■ Der Trick mit der unsichtbaren Tabelle!

Wie erwähnt, die Ausgabe der Daten und das Zeichnen der Balken erfolgt innerhalb mehrerer unsichtbarer Tabellen. Interessant ist vor allem die vorletzte Zelle der jeweiligen Tabelle, schließlich wird hier der Balken erzeugt:

```
<td width="<?php echo $laenge0; ?>" bgcolor="red">&nbsp;</td>
```

Die Zelle selbst fülle ich nur mit einem geschützten Leerzeichen, doch die Länge wird dynamisch durch PHP erzeugt. Die letzte Zelle gibt dann den Absolutwert der jeweiligen Votes aus:

```
<td>&nbsp;<i><?php echo $umfrage[0]; ?></i></td>
```

Falls du hier lieber Prozentwerte magst, bitte sehr! Siehe auch Übung C6 auf der nächsten Seite. Selbstverständlich kannst du die Tabellen zur Darstellung der Balken noch viel raffinierter gestalten. Arbeite mit mehreren verschachtelten Tabellen oder nutze eigene Grafiken als Hintergrund.