

Herrnhuter Losungen auf eigener Internetseite (PHP)

Fassung 6. Januar 2016
für Losungsdateien im CSV-Format und PHP >=5.2.3

Autor: Wolfgang Pavel,
E-Post: post@wpavel.de - Hausseite: www.wpavel.de
Für die Apostelkirche Gerbrunn:
E-Post: pavel@apostelkirche-gerbrunn.de - www.apostelkirche-gerbrunn.de

Zugehörige CSV-Dateien gibt es bei www.losungen.de .

Bei Verwendung der Losungsdateien sind die Nutzungsbedingungen zu beachten, die man bei www.losungen.de findet.

Schnellanweisung

Führen Sie die im folgenden **eingerahmten** Schritte aus.

Schnellstanweisung:

Führen Sie die Anweisungen im [Kasten 1](#) aus, und dann die [Beispiele](#).

Inhalt

[Zweck](#)
[PHP](#)
[Dateien](#)
[Vorbereitung Ihrer HTML-Datei](#)
[Einsetzen der Losungstexte](#)
[Kodierung](#)
[Spezielles](#)
[Fehlende oder fehlerhafte Losungsdateien](#)
[Beispiele](#)

Zweck

Dieses „Programm“ dient der Anzeige der Losungen der Herrnhuter Brüdergemeine auf einer eigenen Internetseite. Diese Losungen, weit verbreitet als „Losungsbüchlein“, enthalten für jeden Tag ein ausgelostes Bibelwort aus dem Alten Testament und ein passendes Bibelwort aus dem Neuen Testament.

Das vorliegende Programm wendet sich an Gestalter von Internetseiten, die die Losungen nicht in einem fertigen Layout, z.B. als Popup-Fenster, anzeigen wollen, sondern die Losungstexte in einer selbstgestalteten Weise in die eigenen Internetseiten integrieren wollen. Die beigefügten Beispiele können als erster Einstieg dienen, lassen dann aber den eigenen Ideen beliebigen Spielraum. Im Prinzip können die Losungstexte dann mit allen normalen HTML- und CSS-Möglichkeiten gestaltet werden.

PHP

Voraussetzung: PHP-Version mindestens 5.2.3 .

Dieses Programm ist in PHP geschrieben. Wenn ein Besucher (Client) eine damit gestaltete Internetseite abrufen, wird das Programm zuerst auf dem Server ausgeführt. Dann wird nur die Ausgabe des Programms über das Netz an den Besucher, d.h. dessen Browser, übertragen, und das ist normalerweise einfaches HTML. Im Gegensatz zu Javascript, bei dem das Programm übertragen und vom Browser dann ausgeführt werden muß, braucht hier der Browser keine Skriptfähigkeiten zu haben (es sei denn, die Seite enthält noch andere Elemente, die das verlangen). Sie müssen deshalb bei der Verwendung dieses Programms nicht auf die verschiedenen Browser, Versionen, Einstellungen bei Ihren Besuchern achten. Allerdings muß der Server, auf dem Ihre Internetseiten liegen, für die Ausführung von PHP-Programmen eingerichtet sein. Normalerweise ist dies bei den heutigen Providern der Fall.

Eine Bemerkung, wenn Sie noch nicht mit PHP gearbeitet haben: Vielleicht sind Sie gewöhnt, Ihre erstellten Internetseiten erst einmal lokal auf Ihrem Rechner auszuprobieren. Das geht sogar mit Javascript-Programmen gut, wenn Ihr eigener Browser das kann. Nicht so bei PHP. Denn für die Ausführung von PHP-Programmen ist ja nicht der Browser, sondern der Server zuständig. Wenn Ihre Internetseite PHP enthält (z.B. diese Lösungsanzeigen) und Sie rufen Ihre HTML-Datei lokal auf, werden Sie enttäuscht sein: die Seite zeigt stellenweise den Originaltext der PHP-Befehle; ausgeführt werden diese nicht. Wenn Sie dann PHP nicht immer über Ihren Server und das Netz ausprobieren wollen, müssen Sie lokal ein Serverprogramm installieren. Der bekannteste ist wohl derzeit APACHE. Damit können Sie Ihre erstellten Internetseiten wie scheinbar über das Netz mit der URL `http://localhost` also nicht als `file://`) abrufen und Ihre PHP-Teile werden ausgeführt. Näheres finden Sie in den Installationsanleitungen der genannten Serverprogramme.

Sie werden zur Verwendung dieses Programms kleinste PHP-Skripte in Ihre Seite setzen müssen. Für die Syntax eines PHP-Skripts innerhalb von HTML gibt es verschiedene gleichwertige Möglichkeiten. Die kürzeste, und hier immer verwendete, ist:
`<? hier kommt das Skript hin; ?>`

Dateien

Notwendig sind die Datei *deflos.php* („deflos“ steht für „Definiere Lösung“) und eine CSV-Datei mit den anzuzeigenden Lösungen. Diese Datei ist für jedes Jahr von der Internetseite www.losungen.de herunterzuladen; der Name enthält normalerweise die Jahreszahl und hat die Endung .csv . (Bitte Urheberrecht und Copyright beachten!)

1	Vorbemerkung: Wenn Sie bei Ihren ersten Versuchen „merkwürdige“ Zeichen in den Lösungstexten sehen, lesen Sie den Abschnitt Kodierung .
	Übertragen Sie die Datei <i>deflos.php</i> und die Lösungsdatei (z.B. <i>losungen2008.csv</i>) in ein Verzeichnis, das von der HTML-Datei erreichbar ist, mit der Sie die Lösungen anzeigen wollen. Fast immer ist es das einfachste, alle diese Dateien im selben Verzeichnis zu haben.
	Benennen Sie die HTML-Datei, mit der sie die Lösungen anzeigen wollen, von der Endung .HTML (oder .HTM) um in .PHP. (Der eigentliche Name kann bleiben). Sonst werden die PHP-Befehle in Ihrer Datei nicht vom Server ausgeführt. Vergessen Sie nicht, in allen vielleicht schon vorhandenen Links auf diese HTML-Seite das .html ebenfalls in .php zu ändern. Sollte es dennoch nicht funktionieren, kann es an Besonderheiten des Servers liegen. Lesen Sie unter Spezielles nach.
	Wenn Sie es jetzt ganz eilig haben, die erste Lösung - in Ihrer eigenen Internetseite - zu sehen, nehmen Sie eines der Beispiele .

Sie können sowohl die Datei *deflos.php* als auch die Lösungsdatei an eine beliebige Stelle, auch unter anderem Namen (aber nicht anderer Endung) kopieren, sofern diese von der HTML-Datei durch relative Pfadangabe erreichbar sind. Dies ist wichtig, wenn Sie die Lösungen in mehreren HTML-Dateien in verschiedenen Verzeichnissen erreichen wollen ohne die Dateien mehrmals zu speichern.

Es ist auch möglich, mehrere Lösungsdateien für verschiedene Jahre zu speichern und das Programm die entsprechende automatisch auswählen zu lassen. (siehe [Spezielles](#))

Vorbereitung Ihrer HTML-Datei

Um die täglich veränderten Losungen (enthalten in der Losungsdatei) in Ihre HTML-Datei einzubauen, müssen zunächst folgende Schritte getan werden:

2	<p>Fügen Sie anfangs irgendwo im Headteil, Bodyteil oder dazwischen, aber vor dem ersten Abrufen eines Lösungstextes, die Zeile ein:</p> <pre><? include("pfad/deflos.php"); deflos("pfad/losungsdatei.csv"); ?></pre> <p>Der Pfad der Dateien ist relativ zur bearbeiteten HTML-Datei. Achten Sie auf Groß-Kleinschreibung und darauf, daß Sie Schrägstriche und nicht Backslashes verwenden. Sind alle diese Dateien im selben Verzeichnis, entfällt die Angabe <i>pfad/</i> . Für <i>losungsdatei</i> setzen Sie die Bezeichnung der von Ihnen verwendeten Losungsdatei ein.</p>
	<p>Wenn Sie die Losungsdatei „<i>losungen.csv</i>“ oder „<i>losungenjahr.csv</i>“ (mit der 4-stelligen Jahreszahl) genannt haben, können Sie die Dateiangabe auch ganz weglassen:</p> <pre><? include("pfad/deflos.php"); deflos(); ?></pre>
	<p><u>Optional</u> können Sie die Zeile auch schreiben:</p> <pre><? include("pfad/deflos.php"); deflos("pfad/losungsdatei.csv", datum); ?></pre> <p>dabei ist <i>datum</i></p> <ul style="list-style-type: none">- entweder ein Text in Anführungszeichen (String) mit dem Datum in deutscher Schreibweise "<i>tag.monat.jahr</i>" (4stellig), z.B. "25.3.2010", oder "<i>tag.monat</i>" für ein Datum im aktuellen Jahr, z.B. "23.9." oder "23.9" ;- oder ein „timestamp“, also eine ganze Zahl (integer) wie von der PHP-Funktion <i>time()</i> geliefert, ist. <p>Sie erhalten dann (mit den im folgenden erklärten Befehlen) die Losung des angegebenen Tages, nicht wie sonst die des aktuellen Datums.</p>

Mit Hilfe der letzten Möglichkeit können Sie in Ihrer Internetseite auch die Wahl verschiedener Losungen anbieten, z.B. die gestrige, heutige und morgige, evtl. unter zusätzlicher Verwendung weiterer PHP-Funktionen. Dies wird hier jedoch nicht mehr näher erläutert.

2a	<p>Optional können Sie statt <code>deflos(...)</code> auch schreiben:</p> <pre><code>\$losungsvariable=deflos(...)</code></pre> <p>Als <i>losungsvariable</i> setzen Sie einen von Ihnen gewählten PHP-Variablennamen ein. Diese Anweisung kann (mit verschiedenen Variablen) auch mehrmals gegeben werden. Auf diese Weise werden mehrere Losungen, z.B. zu verschiedenen Tagen, abgerufen und in den Variablen gespeichert. Die Verwendung der Variablen wird im folgenden erklärt. Im übrigen lesen Sie auch Spezielles.</p>
----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Einsetzen der Lösungstexte

Die Lösung steht in 4 Teilen zur Verfügung: Text des Alten Testaments, Verweis dazu, Text des Neuen Testaments, Verweis dazu. Hinzu kommt das Lösungsdatum in Textform.

	Gestalten Sie die Stellen, an denen die Lösungs-Textteile erscheinen sollen, wie ganz normale HTML-Textstellen, also als Absatz, als Tabellenzelle, mitten in anderen Texten, und so weiter. Verwenden Sie dafür alle Mittel aus HTML und CSS, die Sie auch sonst für Textgestaltung verwenden.
3	Wenn Sie die Lösung nicht in einer Variablen gespeichert haben (Normalfall, Kasten 2), schreiben Sie normalerweise an der Stelle, an der ein Teil des Lösungstextes stehen soll: <? AT(); ?> für den Text des Alten Testaments, <? AV(); ?> für den Verweis des Alten Testaments, <? NT(); ?> für den Text des Neuen Testaments, <? NV(); ?> für den Text des Alten Testaments, <? LD(); ?> für das Lösungsdatum als Text

Haben Sie, wie in [weiter oben](#) beschrieben, eine Lösung in einer PHP-Variablen gespeichert, können Sie ihre einzelnen Textteile ausgeben durch:

```
<? teil($losungsvariable); ?> oder
```

```
<? teil("losungsvariable"); ?>
```

wobei für teil wieder AT, AV, NT, NV oder LD zu setzen ist,

und für *losungsvariable* die PHP-Variable aus der Zuweisung nach [Kasten 2a](#).

Kodierung

Die Kodierung der Schriftzeichen sowohl der „Eingabe“, also der jeweiligen CSV-Lösungsdatei, als auch der Ausgabe, meist die Ergebnisse der Funktionen *AT()*, *NT()*, usw. , beeinflusst das Aussehen der Schriftzeichen in Ihrer Internetseite. Wenn Sie dort „merkwürdige“ Zeichen, insbesondere statt der deutschen Umlaute, sehen, müssen Sie das Folgende beachten.

Eingabe-Kodierung

Die CSV-Dateien der Brüdergemeine lagen bis zum Jahr 2014 im Zeichencode ISO-8859-1, vor, 2015 im Zeichencode UTF-16LE. Das Skript *deflos.php* erkennt automatisch diese beiden, sodass Sie hierzu keine eigenen Maßnahmen brauchen. Auch die Zeichencodes UTF-8 und UTF-16BE werden automatisch erkannt, falls die Lösungsdateien in diesen Codes zur Verfügung gestellt werden sollten. Die ebenfalls zur Verfügung gestellte CSV-Datei für MAC kann auch verwendet werden und wird vom Skript *deflos.php* automatisch erkannt.

Ausgabe-Kodierung

Ohne besondere Maßnahmen werden die Texte mit diesem Skript in der Zeichen-Kodierung UTF-8 ausgegeben. Dies ist wohl derzeit die in Internetseiten meist verwendete und dafür empfehlenswerteste Kodierung. Falls durch Ihren Server nicht automatisch eingestellt, erreichen Sie diese Kodierung, indem Sie im Header (zwischen `<head>` und `</head>`) Ihrer Internetseite schreiben:

```
<meta http-equiv="content-type" content="text/html; charset="UTF-8" />
```

Das ist natürlich nur sinnvoll, wenn alle Schriftzeichen Ihrer HTML-Datei in UTF-8 kodiert sind, nicht nur die Lösungen; andernfalls sehen zum Beispiel die deutschen Umlaute auf Ihrer Seite außerhalb der Lösungen „merkwürdig“ aus.

Sollte Ihre Internetseite automatisch oder durch die Anweisung im Header:

```
<meta http-equiv="content-type" content="text/html; charset="ISO-8859-1" />
```

auf die Kodierung ISO-8859-1 eingestellt sein (besonders früher in unseren Breiten meist gebraucht), können Sie die Lösungen auch in dieser Kodierung ausgeben. Dazu geben Sie im anfänglichen Aufruf von *deflos.php* einen zusätzlichen Parameter an:

```
deflos("pfad/losungsdatei.csv", datum, isoangabe)
```

wobei *isoangabe* entweder 1 oder 2 ist, was bedeutet:

1 : Die Ausgabe erfolgt in ISO-8859-1, die deutschen Umlaute werden auch in dieser Kodierung ausgegeben.

2: Die Ausgabe erfolgt in ISO-8859-1, Sonderzeichen wie die deutschen Umlaute werden durch HTML-Entities ersetzt, z.B. ä durch *ä* oder Ü durch *Ü* , was in HTML auch dann korrekt dargestellt wird, wenn die HTML-Seite nicht die charset-Angabe ISO-8859-1 hat.

Soll der 2. Parameter eigentlich fehlen (aktuelles Datum), schreiben Sie entweder dort einen leeren Text, also *deflos("pfad/losungsdatei.csv", "", isoangabe)* oder lassen Sie den zweiten Parameter einfach aus: *deflos("pfad/losungsdatei.csv", isoangabe)* .

Diese Einstellung gilt für alle Ausgaben, die mit den Funktionen *AT()*, *AV()*, *NT()*, *NV()* oder *LD*, bzw. *AT_()*, *AV_()*, *NT_()*, *NV_()* oder *LD_()* abgerufen werden.

Spezielles

Mehrere Lösungsdateien

Sie können das Programm aus mehreren Lösungsdateien die zu einem bestimmten Jahr passende auswählen lassen:

- Speichern Sie alle diese Lösungsdateien in demselben Verzeichnis.
- Geben Sie allen diesen Lösungsdateien fast denselben Namen; unterscheiden Sie nur durch eine 1- bis 4-stellige Zahlenangabe, z.B. *losungen2007.csv*, *losungen2008.csv*, usw. Achten Sie auf einheitliche Groß-Kleinschreibung.
- Die unterscheidende Zahl muss aus allen oder den letzten Ziffern des betreffenden Jahres bestehen.
- Schreiben Sie in `deflos("lösungsdatei")` (siehe [Kasten 2](#)) an Stelle der erwähnten Ziffern im Dateinamen #-Zeichen, also z.B.: `deflos("lösungen####.csv")`

Wenn Sie in `deflos(..., datum)` ein Datum angeben (siehe [Kasten 2](#)), wird die Lösungsdatei zum Jahr dieses Datums gesucht, andernfalls zum aktuellen Jahr.

Diese Jahresabhängigkeit funktioniert für jeden Dateinamen an Stelle von „lösungen“. Wenn Ihre Lösungsdatei aber tatsächlich „lösungen“, evtl. mit angehängter Jahreszahl, heißt, ist es besonders einfach: Der Anfangsaufruf kann dann einfach `deflos()` lauten; das Programm sucht zuerst nach einer Datei *lösungenjahr.csv* mit der aktuellen Jahreszahl, dann nach einer Datei *lösungen.csv* ohne Jahreszahl.

Losungstexte zum Weiterverarbeiten

Mit den Funktionsaufrufen `AT_()`, `AV_()`, `NT_()`, `NV_()` oder `LD_()` erhält man die Texte in derselben Weise wie in den vorher beschriebenen Aufrufen `AT()`, `AV()`, `NT()`, `NV()` bzw. `LD()`, aber ohne dass die Texte ausgegeben werden. Man kann diese Texte, vom PHP-Typ `string`, mit PHP-Befehlen weiter verarbeiten und z.B. letztlich mit dem PHP-Befehl `echo` ausgeben.

Für die Parameter in den Aufrufen gilt dasselbe wie für die vorher beschriebenen Ausgabeaufrufe. Auch hier werden Kodierungseinstellungen (siehe [Kodierung](#)) berücksichtigt.

Losungs-Standardvariable

Nach dem Abrufen einer Lösung mit `deflos(...)` ist die Lösung immer in der PHP-Variablen `$LOSUNG` gespeichert, egal, ob man sie, wie im [Kasten 2a](#) beschrieben, auch einer eigenen Variablen zugewiesen hat oder nicht.

Ruft man `deflos(...)` mehrmals auf, enthält `$LOSUNG` die jeweils zuletzt abgerufene Lösung.

Struktur der Lösungs-Variablen

Wer es ganz genau wissen will:

Jede Lösungsvariable, also `$LOSUNG` und die, wie im [Kasten 2a](#) beschrieben, erzeugten Variablen, ist ein assoziatives Array mit den 5 Elementen:

`$variable["AT"]`, `$variable["AV"]`, `$variable["NT"]`, `$variable["NV"]` und `$variable["LD"]`.

Jedes dieser 5 Elemente ist seinerseits ein Array, wobei jedem Element eine Textzeile entspricht. (Die Arrays zu `AV`, `NV` und `LD` werden wohl immer nur aus einem Element bestehen.)

Zum Beispiel erhält man die zweite Zeile des Textes aus dem Neuen Testament (sofern der Text mindestens 2 Zeilen hat) mit: `$LOSUNG["NT"][1]`. (Array-Zählung ab Null!) Statt `$LOSUNG` ist ggf. auch eine der eigenen Lösungsvariablen möglich.

[Kodierungen](#) und Text hervorhebungen sind in diesen Arrays noch nicht berücksichtigt; alle Texte in diesen Elementen sind UTF-8-kodiert. Deshalb rufen Sie die Texte normalerweise nur mit den dafür vorgesehenen Funktionen `AT()`, `AV()`, `NT()`, `NV()` oder `LD()`, bzw. `AT_()`, `AV_()`, `NT_()`, `NV_()` oder `LD_()` ab, oder Sie bearbeiten die Arrayelemente selbst bezüglich Kodierung und Hervorhebungen.

PHP funktioniert nicht

Wenn Sie wie in [Kasten 1](#) beschrieben die notwendigen Dateien auf den Server übertragen haben, dann beim Aufruf Ihrer Seite aber die PHP-Befehle im Original sehen und nicht die Wirkung ihrer Ausführung, kann dies an folgendem liegen:

- Der Server ist überhaupt nicht für PHP eingerichtet.
In dem hier wohl seltenen Fall, dass Sie Ihren eigenen Server betreiben, müssen Sie eben PHP auf Ihrem Server einrichten.
In dem hier wohl meist vorliegenden Fall, dass Sie Ihrer Internetseiten auf einem bei einem "Provider" gemieteten "Webspace" liegen haben, können Sie selbst gar nichts installieren. Sie müssen Ihren Provider bitten, PHP zu installieren oder Sie müssen Ihren Provider wechseln.
- Der von Ihnen benutzte Server will die Endung .PHP nicht, sondern die Mitangabe der Version.
Benennen Sie Ihre nach [Kasten 1](#) bereits in die Endung .PHP umbenannte HTML-Datei erneut um in die Endung .PHP5. Sie müssen dann aber auch die Programmdatei in *deflos.php5* umbenennen.

INDEX.PHP statt INDEX.HTML ?

Bekanntlich wird beim Aufruf einer URL ohne Angabe einer Datei die Datei *index.html* im URL-Stammverzeichnis aufgerufen. Wenn Sie nun gerade auf dieser Startseite die Lösungen anzeigen wollen, müssen Sie sie, wie vorhin erklärt, in *index.php*, bzw. *index.php5* umbenennen. Bei einem gut eingerichteten Server sollte das kein Problem sein. Dort wird eine so benannte Datei ebenfalls als Startdatei genommen, wenn es keine *index.html* gibt.

Wenn es nicht funktioniert, können Sie:

- Eine *index.html* schaffen, die nichts anderes tut, als zu *index.php* weiterzuleiten.
- Ihren Provider bitten, den Server entsprechend einzurichten.
- Wenn als Server APACHE verwendet wird, eine zusätzliche Datei mit dem Namen *.htaccess* (den Punkt nicht vergessen) in Ihr URL-Stammverzeichnis (in dem auch *index.php* liegt) zu stellen. In diese Datei schreiben Sie - als lesbaren Text mit einem Editor - die Zeile:

```
DirectoryIndex index.html index.php
```

Sie können noch weitere Dateibezeichnungen angeben, die beim Aufruf der Reihe nach gesucht werden; die erste gefundene wird als Startdatei angezeigt.

Vergessen Sie aber nicht, auch alle Links auf Ihre *index.html* auf *index.php* zu ändern, denn die werden nicht durch *.htaccess* umgeleitet.

Fehlende oder fehlerhafte Lösungsdateien

Die Lösung kann natürlich nicht angezeigt werden, wenn die zum Tag gehörende Lösungsdatei nicht auf Ihrem Server gefunden wird (vergessen, die neuesten zu überspielen?) oder fehlerhaft ist. In dem Programm wurde auf Ausgabe einer Standardfehlermeldung in diesen Fällen verzichtet, weil dies meist störend und irritierend auf der Internetseite aussieht. Alle Funktionen (deflos, AT, AV, NT, NV, LD, AT_, AV_, NT_, NV_, LD_) liefern false zurück, wenn die Lösung insgesamt oder einer der Teile fehlerhaft ist. In der Ausgabe erscheint deshalb im Fehlerfall einfach nichts.

Nach (jedem) Ausführen der Funktion deflos(. . .) wird in der Variablen \$DEFLOS ein Text gespeichert: war die Ausführung fehlerfrei, steht dort "OK", andernfalls eine kurze Fehlerbeschreibung. Mit z.B. echo \$DEFLOS oder – auf den Fehlerfall beschränkt – mit if (deflos(...)==false) echo \$DEFLOS

können Sie diesen Text für Testzwecke in ihrer HTML-Seite ausgeben.

Aber vielleicht wollen Sie eigene Varianten für Fehlerfälle programmieren, z.B. das Anzeigen der Jahreslösung oder das Weglassen der gesamten Umgebung für die Lösung.

Dazu fragen Sie mit eigenen PHP-Befehlen in Ihrer HTML-Datei die Rückgabewerte der Funktionen ab. Hier (nochmal) im einzelnen:

deflos liefert Lösungsarray (siehe [Spezielles](#)), wenn ok; liefert false wenn Fehler; (setzt zusätzlich die Variable \$DEFLOS).

AT , AV , NT , NV , LD liefern true, wenn ok; liefern false, wenn Fehler.

AT_ , AV_ , NT_ , NV_ , LD_ liefern Lösungstextteil, wenn ok; liefern false, wenn Fehler.

Ist die Lösung als ganzes fehlerhaft, liefert also bereits deflos false, so liefern auch alle anderen Funktionen false.

Damit können Sie in Ihrer HTML-Datei fehlerhaft oder richtig unterscheiden, z.B.:

```
if (!deflos(...)) { Skript wenn Lösung ganz fehlerhaft ; }
if (!AT(...)) { Skript wenn Text aus Altem Testament fehlerhaft ; }
if (NT(...)) { Skript wenn Text aus Neuem Testament richtig ; }
               else { Skript wenn Text aus Neuem Testament fehlerhaft ; }
```

Beispiel für die Behandlung eines Fehlerfalls:

- Datei *bsp4.php*: enthält absichtlich einen Fehler (ungültiger Lösungsdateiname); dadurch wird der gesamte Lösungstext nicht angezeigt; es erscheint hier eine entsprechende Meldung. Nach „Korrektur“ des Fehlers erscheint die Lösungsdatei wieder.

Wenn eine Lösung zwar ohne Fehlermeldung angezeigt wird, aber „merkwürdig“ fehlerhafte Schriftzeichen enthält, lesen Sie den Abschnitt [Kodierung](#).

Beispiele

Die folgenden Beispiele können als Vorlage direkt verwendet werden, aber auch kopiert, ausprobiert und dann zum Experimentieren abgeändert werden.

Speichern Sie alle Beispiel-Dateien, die Sie ausprobieren wollen, zusammen mit *deflos.php* und der Lösungsdatei gemeinsam in dasselbe Verzeichnis; benennen Sie die Lösungsdatei in *losungen.csv* um.

(Eine Lösungsdatei ist aus lizenzrechtlichen Gründen hier nicht enthalten; Sie müssen sie, wie anfangs beschrieben, bei der Brüdergemeine herunterladen.)

Beachten Sie evtl. die Hinweise unter [Spezielles](#), um die Endungen .PHP der Dateien und alle Links auf diese Dateien notfalls zu ändern.

1. Beispiel:

Dateien: *bsp1.php*, *bsp1a.html*

- *bsp1.php* ist eine selbständige Seite: Rufen Sie die Datei *bsp1.php* in Ihren Browser auf; die Lösung wird dann angezeigt.
- *bsp1a.html* enthält einen Link: Rufen Sie die Datei in Ihren Browser auf und klicken Sie auf „hier“; damit wird die Datei *bsp1.php* aufgerufen, die Lösung wird angezeigt.
- Oder erstellen Sie einen Link wie in *bsp1a.html* in Ihrer eigenen HTML-Datei.

2. Beispiel:

Datei: *bsp2.php*

Damit fügen Sie die Lösung in Ihre eigene Seite als Textabsatz ein. Öffnen Sie Ihre eigene HTML-Datei in einem Editor; öffnen Sie ebenso die Datei *bsp2.php* in einem Editor. Kopieren Sie die in *bsp2.php* gekennzeichneten Zeilen an eine entsprechende Stelle in Ihrer HTML-Datei. Benennen Sie Ihre HTML-Datei mit der Endung .PHP . Falls es Problem mit der Darstellung von Schriftzeichen gibt, lesen Sie den Abschnitt [Kodierung](#).

Zum reinen Ausprobieren können Sie auch zunächst nur die Datei *bsp2.php* in Ihrem Browser aufrufen. Kopieren Sie dazu, falls nicht schon geschehen, die Datei *bsp2.php* in dasselbe Verzeichnis wie *deflos.php* und die Lösungsdatei.

3. Beispiel:

Datei: *bsp3.php*

Damit fügen Sie die Lösung in Ihre eigene Seite in einem Tabellenrahmen ein. Öffnen Sie Ihre eigene HTML-Datei in einem Editor; öffnen Sie ebenso die Datei *bsp3.php* in einem Editor. Kopieren Sie die in *bsp3.php* gekennzeichneten Zeilen an eine entsprechende Stelle in Ihrer HTML-Datei. Falls es Problem mit der Darstellung von Schriftzeichen gibt, lesen Sie den Abschnitt [Kodierung](#).

Zum reinen Ausprobieren können Sie auch zunächst nur die Datei *bsp3.php* in Ihrem Browser aufrufen. Kopieren Sie dazu, falls nicht schon geschehen, die Datei *bsp3.php* in dasselbe Verzeichnis wie *deflos.php* und die Lösungsdatei.

Und dann experimentieren Sie mit den Attributen, bis es so aussieht, wie es Ihnen gefällt.

4. Beispiel:

Datei *bsp4.php*

Dieses Beispiel enthält absichtlich einen Fehler (ungültiger Lösungsdateiname); dadurch wird der gesamte Lösungstext nicht angezeigt; es erscheint hier eine entsprechende Meldung.

Nach „Korrektur“ des Fehlers erscheint die Lösungsdatei wieder.