

---

# PHP für Einsteiger - Programmbeispiele

---

Nachfolgend finden Sie die Beschreibung der Programmbeispiele zum Buch. Die einzelnen Skripte finden Sie mit Beschreibung für die entsprechenden Kapitel:

## Kapitel 2

**Beispiel2\_1.php** (HTML in PHP-Datei) – Diese Datei hat die Endung `.php` und beinhaltet HTML-Code. Beim Aufruf wird wegen der Dateiendung auf PHP-Code untersucht, es wird keiner gefunden, daher wird es dann an den Browser übermittelt, der interpretiert dann den HTML-Code.

**Beispiel2\_2.php** (nur PHP-Code in Tags) – Diese Datei beinhaltet nur PHP, mittels `echo`-Anweisung wird etwas auf dem Bildschirm ausgegeben. Der Code wird vom PHP-Interpreter interpretiert, die `echo`-Anweisung wird ausgeführt und der Text `"hello world"` wird ausgegeben. Wenn Sie sich den Quelltext Ihres Browsers ansehen, dann findet man in dem Dokument nur diesen Text.

**Beispiel2\_3.php** (HTML- und PHP-Code in einem Dokument) – Diese Datei beinhaltet HTML- und PHP-Code. Die Auswertung ist wie folgt: Da das Dokument die Endung `.php` hat wird es geparsed und auf PHP-Inhalte untersucht und der gefundene Code ausgewertet. Dann wird das Dokument also die HTML-Tags an den Browser übertragen und an der Stelle, an der sich die `echo`-Anweisung befindet, wird der Text `"hello world mit PHP"` eingebaut und dann der Rest der HTML-Tags. Sehen Sie sich zum besseren Verständnis im Browser den Quellcode an.

**Beispiel2\_4.php** (Zeilenkomentar) – In Beispiel2\_3.php wurde PHP-Code, der sich innerhalb der Tags befindet, ausgegeben, einfach zwei Schrägstriche davor kommentieren diesen Code aus. Starten Sie Beispiel2\_4.php, die `echo`-Anweisung ist dort als Kommentar gekennzeichnet und wird daher nicht ausgeführt, es erfolgt also keine Ausgabe.

**Beispiel2\_5.php** (Include) – Dieses Beispiel zeigt, wie mit `include` eine PHP-Datei eingebunden werden kann. Es wird im HTML-Code einfach mittels PHP-Tags der Befehl verwendet.

## Kapitel 3

**Beispiel3\_1.php** (`echo` mit Parametern) – Die `echo`-Anweisung wird in diesem Beispiel mit mehreren Parametern aufgerufen. Auf diese Weise kann man auch numerische und String-Werte prima miteinander verwenden.

**Beispiel3\_2.php** (`echo` mit dem Punkt-Operator) – Die `echo`-Anweisung wird in diesem Beispiel mit dem Punkt-Operator aufgerufen. Der dient dazu, um Strings miteinander zu verknüpfen. Es ist allerdings nicht möglich, auch numerische Werte zu verbinden.

**Beispiel3\_3.php** (`echo` mit dem Punkt-Operator) – Die `echo`-Anweisung wird in diesem Beispiel mit mehreren Parametern aufgerufen. Auf diese Weise kann man auch numerische und String-Werte prima miteinander verwenden. In Zeile 12 dieses Programmes wird ein Fehler verursacht, weil ein numerischer Wert mit dem Punkt-Operator verwendet wird.

**Beispiel3\_4.php** (erzeugen einer Tabelle) – Jetzt wird es dynamisch (fast!), in diesem kleinen Beispiel wird per `echo`-Anweisung eine Tabelle erzeugt. Wenn jetzt hier auch noch die `for`-Anweisung (siehe Kapitel 8) mitspielen würde, wäre wirklich eine dynamisch erzeugte Tabelle möglich. Aber ein erster Ansatz ist schon erkennbar.

**Beispiel3\_5.php** (erzeugen einer Tabelle) – Einen großen Unterschied zu Beispiel3\_4.php gibt es nicht, allerdings wenn man sich den Quelltext beider Programme ansieht, dann wird die Wirkung der ESC-Sequenz offensichtlich. Mit der ESC-Sequenz `\n` wird ein Zeilenvorschub eingefügt, allerdings nur im Quellcode.

## Lektion 4

**Beispiel4\_1.php** (Definieren und Verwenden von Konstanten) – Hier wird gezeigt, wie Konstante definiert werden und anschließend im Programm verwendet.

**Beispiel4\_2.php** (Definieren und Verwenden von Literalkonstanten) – Hier wird gezeigt, wie Literalkonstanten verwendet werden.

**Beispiel4\_3.php** (Verwenden vordefinierter Konstanten) – Hier wird gezeigt, wie vordefinierte Konstanten verwendet werden. Diese vordefinierten Konstanten sind bereits im Core von PHP definiert.

**Beispiel4\_4.php** (Verwenden magischer Konstanten) – In diesem Beispiel wird die magische Konstante `__LINE__` verwendet. Obwohl es sich hier quasi um eine Konstante handelt (lässt sich während des Programmlaufs nicht verändern) ist der Wert der sog. magischen Konstanten von den Umständen abhängig. Auch diese magischen Konstanten sind bereits im Core von PHP definiert.

## Lektion 5

**Beispiel5\_1.php** (Definition von Variablen) – In diesem kleinen Programm werden Variablen verschiedenen Typs definiert. Der guten Praxis halber sollten Variablen und auch Konstanten immer am Anfang eines Skripts definiert werden.

**Beispiel5\_2.php** (Variablenausgabe in Hochkommata) – Dieses Beispiel zeigt den Unterschied, wenn Variablen zwischen einfachen Hochkommata und Gänsefüßchen ausgegeben werden. Zwischen doppelten Hochkommata (Gänsefüßchen) werden Bezeichner von Variablen immer ausgewertet.

**Beispiel5\_3.php** (spezielle Definition von Variablen) – In diesem kleinen Programm werden verschiedene Variablen definiert. Möglicherweise sind Sie etwas verwundert, weshalb für `$a` der Wert 40 ausgegeben wird, dies ist deshalb so, da `$a` in Zeile 10 als Zeiger auf den Wert von `$b[1]` gesetzt wurde. Wenn Sie den Wert von `$b[1]` ändern, dann ändert sich ebenfalls der Wert von `$a`. Für die String-Variable `$s` wird 0 ausgegeben, weil durch die Anweisung `(int)` vor dem Wert die Variable auf den Typ Integer festgelegt wurde. Entfernen Sie mal diese Typfestlegung, dann wird **Peter** ausgegeben.

**Beispiel5\_4.php** (Definition und Verwendung eines Arrays) – Dieses Beispiel zeigt, wie ein Array definiert und wie dessen Werte ausgelesen werden können bzw. wie mittels Index darauf zugegriffen wird. Die Zählung des Index beginnt immer bei 0, also der erste Wert hat den Index 0 der zweite den Index 1 usw.

**Beispiel5\_5.php** (Definition und Verwendung eines assoziativen Arrays) – In diesem Beispiel wird ebenfalls ein Array definiert, hier handelt es sich aber um ein sog. assoziatives Array. Dies bedeutet, dass der Zugriff nicht wie in Beispiel5\_4 per Index, sondern mittels Namen (Key) erfolgt.

**Beispiel5\_6.php** (Definition und Verwendung variabler Variablen) – Eine ganz besondere Art von Variablen sind sogenannte variable Variablen. Bei diesen Variablen wird der Wert einer Variablen als Name einer anderen Variablen verwendet. In diesem Beispiel wird gezeigt wie das funktioniert, variable Variablen werden definiert und deren Wert per `echo` ausgegeben.

**Beispiel5\_7.php** (Verwendung externer Variablen) – In diesem Beispiel wird in einem Formular ein Wert eingegeben, dann wird dieser Wert an sich selbst (Beispiel5\_7.php) übergeben und per `echo` ausgegeben.

HINWEIS: Beim ersten Aufruf des Skriptes gibt es eine Fehlermeldung, weil der Wert `$_POST["Inhalt"]` beim ersten Aufruf nicht existiert. Um diese Fehlermeldung zu vermeiden, ist der Einsatz von `if` erforderlich, in Beispiel7\_1.php finden Sie eine dahingehend verbesserte Version.

**Beispiel5\_8.php** (Ausgabe von Variableninhalten mit `var_dump()`) – Dieses Beispiel zeigt, wie man sich den Inhalt und Typ einer Variablen detailliert mit einem Befehl ausgeben lassen kann.

## Lektion 6

**Beispiel6\_1.php** (Typecasting mit Variablen) – Dieses Beispiel zeigt, wie man sich den Typ (und manchmal auch den Inhalt) einer Variablen mittels Typecasting bei der Ausgabe manipulieren kann.

**Beispiel6\_2.php** (Typecasting mit Variablen) – Dieses Beispiel zeigt, wie man sich den Typ (und manchmal auch den Inhalt) einer Variablen bereits bei der Definition mittels Typecasting manipulieren kann.

## Lektion 7

**Beispiel7\_1.php** (Verwendung externer Variablen und der `if`-Anweisung) – In diesem Beispiel finden Sie eine verbesserte Version von Beispiel5\_7.php. Es wird in einem Formular ein Wert eingegeben, dann wird dieser Wert durch den Aufruf der eigenen Datei an sich selbst übergeben und per `echo` ausgegeben. Die Fehlermeldung, die es beim ersten Aufruf des Skriptes (Beispiel5\_7.php) gab, wird nun mittels `if`-Abfrage vermieden.

**Beispiel7\_2.php** (Verwendung externer Variablen und `if/else`-Anweisung) – Dieses Beispiel ist eine Erweiterung von Beispiel7\_1.php. Es wird in einem Formular ein Wert eingegeben, dann wird dieser Wert durch den Aufruf der eigenen Datei an sich selbst übergeben und per `echo`-Befehl auf den Bildschirm geschrieben. Beim ersten Start, wenn noch kein Wert übergeben wurde, wird eine Fehlermeldung mittels `if`-Abfrage vermieden und zusätzlich mittels `else` ein alternativer Text ausgegeben.

**Beispiel7\_3.php** (Verwendung von `if` und `elseif`) – In diesem Beispiel wird die aktuelle Systemzeit ausgelesen und entsprechend der Tageszeit wird der Benutzer mit **guten Morgen, guten Abend** etc. begrüßt und damit die Verwendung der `if`, der `else` und der `elseif`-Anweisung gezeigt. Die `elseif`-Anweisung kann beliebig verschachtelt werden.

**Beispiel7\_4.php** (Die `switch`-Anweisung) – In diesem kleinen Skript wird der Einsatz der `switch`-Anweisung verdeutlicht. Die Funktion `date("w")` gibt den aktuellen Wochentag als numerischen Wert zurück. Das Ergebnis wird mit `switch` ausgewertet und der Variablen `$tag` der aktuelle Tag als String zugewiesen.

**Beispiel7\_5.php** (**AUFGABE:** Fahrkostenberechnung mit `if` und `else`) – In diesem Skript ist die Programmieraufgabe des Kapitels gelöst. Verändern Sie spaßhalber der Werte der Variablen, um zu sehen, ob das Programm funktioniert.

## Kapitel 8

**Beispiel8\_1.php** (Eine einfache `for`-Schleife) – In diesem Skript wird eine einfache `for`-Schleife gezeigt, in der mit einer `echo`-Anweisung der Schleifenzähler ausgegeben wird.

**Beispiel8\_2.php** (Eine einfache `foreach`-Schleife) – In diesem Skript wird eine einfache `foreach`-Schleife gezeigt, in der mit einer `echo`-Anweisung der Wert des aktuellen Elements ausgegeben wird.

**Beispiel8\_3.php** (Auslesen eines Arrays mit `for`-Schleife) – In diesem Skript wird der Unterschied zwischen `foreach`-Schleife und `for`-Schleife aufgezeigt. Die `for`-Schleife erfordert deutlich höheren Aufwand.

**Beispiel8\_4.php** (Eine einfache `foreach`-Schleife) – In diesem Skript wird eine `foreach`-Schleife verwendet. Im Gegensatz zu Beispiel8\_2.php wird hier ein assoziatives Array genutzt. Mit der mit einer `echo`-Anweisung wird der Schlüssel (Key) und der Wert des aktuellen Elements ausgegeben.

**Beispiel8\_5.php** (Eine einfache `while`-Schleife) – In diesem Skript wird eine einfache `while`-Schleife gezeigt, in der mit einer `echo`-Anweisung der Schleifenzähler ausgegeben wird. Damit die Schleife auch irgendwann einmal beendet wird, muss der Schleifenzähler (als Abbruchbedingung) hochgezählt werden.

**Beispiel8\_6.php** (Eine einfache `while-do`-Schleife) – In diesem Skript wird eine einfache `do-while`-Schleife gezeigt, in der mit einer `echo`-Anweisung der Schleifenzähler ausgegeben wird. Damit die Schleife auch irgendwann einmal beendet wird, muss der Schleifenzähler (als Abbruchbedingung) hochgezählt werden.

**Beispiel8\_7.php** (Eine `for`-Schleife und die Anweisung `continue`) – In diesem Skript wird eine einfache `for`-Schleife gezeigt, in der mit einer `echo`-Anweisung der Schleifenzähler ausgegeben wird.

**Beispiel8\_8.php** (Ein Beispiel für die Anweisung `goto`) – In diesem Skript wird die einfache Anwendung von `goto` gezeigt. Die Schleife wird nicht ausgeführt, es wird aufgrund der `goto`-Anweisung die Schleife übersprungen.

**Beispiel8\_9.php (AUFGABE: Fahrkostenberechnung (erweiterte Version)-** Die Programmieraufgabe des Kapitels mit `if` und `else`) – In diesem Skript ist die Programmieraufgabe des Kapitels gelöst. Mit einer Schleife wird die Anzahl der Personen festgelegt.

**Beispiel8\_10.php (AUFGABE: Die Programmieraufgabe Quadratzahlenberechnung)** – In diesem Skript ist die Programmieraufgabe des Kapitels gelöst.

**Beispiel8\_11.php (AUFGABE: Die Programmieraufgabe Zahlenreihen)** – In diesem Skript ist die Programmieraufgabe des Kapitels gelöst. Bei dieser Aufgabe müssen Sie ein bisschen über den Tellerrand hinaussehen und kreativ werden. Die erste Überlegung gilt der Frage, was haben die Zahlenreihen gemeinsam und was nicht. Der beste Ansatz ist Array, wie in der Lösung gezeigt. Der Trick ist quasi, dass die Anzahl der Durchläufe und anderer Variablen im Array gespeichert ist.

## Lektion 9

**Beispiel9\_1.php** (Definition und Aufruf einer einfachen Funktion) - Dieses Beispiel zeigt die Definition einer Funktion und dann deren Aufruf aus dem Skript.

**Beispiel9\_2.php** (Definition und Aufruf einer Funktion mit Parametern) - Dieses Beispiel zeigt die Definition einer Funktion mit Parametern und dann deren Aufruf und Parameterübergabe aus dem Skript.

**Beispiel9\_3.php** (Definition und Aufruf einer Funktion mit einer variablen Anzahl an Parametern) - Dieses Beispiel zeigt die Definition einer Funktion mit einer variablen Anzahl an Parametern und dann deren Aufruf und Parameterübergabe aus dem Skript.

**Beispiel9\_4.php** (Definition und Aufruf einer Funktion mit Parametern-Verweis) - Dieses Beispiel zeigt die Definition einer Funktion mit Parametern als Verweis und dann deren Aufruf und Parameterübergabe aus dem Skript.

**Beispiel9\_5.php (AUFGABE: Zinseszinsrechnung mit Endbetrag)** Dieses kleine Programm berechnet aus einem eingegebenen Startkapital und einem Zins, wie lange es dauert, bis ein bestimmter Endbetrag erreicht ist.

**Beispiel9\_6.php (AUFGABE: kleinstes gemeinsames Vielfaches)** Dieses Programm soll das sog. kgV errechnen, das kleinste gemeinsame Vielfache aus 2 Zahlen.

**Beispiel9\_7.php (AUFGABE: Multiplikation nur durch Addition)** Dieses Beispiel zeigt deutlich, dass die Überprüfung einer korrekten Eingabe durch den Benutzer oftmals aufwändiger ist als der Programmteil, in dem die Berechnungen durchgeführt werden.

Zusätzlich zu den Beispielprogrammen in den einzelnen Kapiteln finden Sie im Verzeichnis Extras noch ein paar Beispiele. Die Codeschnipsel sind so simpel wie möglich gehalten und kommentiert, damit Sie diese für eigene Zwecke bequem anpassen können.

## Extra-Dateien

**Download.php** Dieses Beispiel zeigt, wie nach Eingabe von Benutzername und Passwort ein Download vom Server veranlasst wird.

**Upload.php** In diesem Beispiel wird gezeigt, wie der Benutzer eine Datei vom eigenen PC auswählen kann und auf den Server hochlädt.

**Rechner.php** Dieses Beispiel ist ein kleiner Rechner für die vier Grundrechenarten.

**Pwkontrolle.php** In diesem Beispiel muss der Benutzer Name und Passwort eingeben und bei korrekter Eingabe wird eine neue Seite dynamisch generiert.