

# Übungen PHP-Grundlagen (aus älterem WE-Script)

## Lektion - HTML und PHP (entspricht Lektion 04 'Hallo Welt')

### Übung 1:

Legen Sie einen Unterordner lektion03 im Ordner phpschulung an. Speichern Sie dort ein HTML-Dokument namens spass.php, das im `<body>`-Tag den Text PHP macht Spaß! stehen hat.

**Hinweis:** Legen Sie fortan für alle Übungen einer Lektion zunächst einen entsprechenden Unterordner an.

### Übung 2:

Verändern Sie das Dokument. Der Text PHP macht Spaß! soll nun mit der PHP-Anweisung `echo` ausgegeben werden.

### Übung 3:

Öffnen Sie nun die Datei spass.html direkt im Firefox-Browser über **Datei** → **öffnen** und beobachten Sie, was passiert.

**Hinweis:** Seit Version 4 ist im Mozilla Firefox die Menüleiste standardmäßig deaktiviert. Sie kann jedoch durch Drücken der (linken) Alt-Taste eingeblendet werden. Alternativ lässt sich die Datei auch einfach auf den Anzeigebereich des Browsers ziehen, wodurch sie im Browser geöffnet wird.

### Übung 4:

Öffnen Sie die Datei nun über den Webserver, also über `http://localhost/phpschulung/lektion03/spass.php`, und beobachten Sie, was passiert.

### Übung 5:

Benennen Sie die Datei in spass.html um, öffnen Sie sie über `http://localhost/phpschulung/lektion03/spass.html` und beobachten Sie den Unterschied in der Darstellung.

## Optionale Übungen

### Übung 6:

Schreiben Sie ein HTML-Dokument mit dem Namen *formular.php*, das ein Formular mit den folgenden Feldern enthält:

- `<input type="text" name="benutzername" value="" />`,
- `<input type="password" name="passwort1" value="" />`,
- `<input type="password" name="passwort2" value="" />` und
- `<input type="submit" value="registrieren" />`.

### Übung 7:

In dem Feld `benutzername` soll nun standardmäßig "Mr. X" stehen, was über das `value`-Attribut möglich ist. Der Text soll dort mittels einer PHP-`echo`-Anweisung eingetragen werden.

### Übung 8:

In den beiden `password`-Feldern sollen jeweils fünf Sterne (\*\*\*\*\*) stehen. Auch diese sollen dort mit `echo` eingetragen werden. Wie werden die Sterne im Browser dargestellt?

## Lektion Variablen und Operatoren (entspricht Lektion 05 'variablen Code schreiben — mit Variablen')

### Übung 9:

Schreiben Sie ein PHP-Skript *kommentare.php*, das beide Kommentar-Typen verwendet.

Anmerkung: wir kennen sogar drei Arten

### Übung 10:

Schreiben Sie ein PHP-Skript *variablen.php*, das die Variablen `$vorname`, `$nachname` und `$alter` definiert. Weisen Sie den Variablen Ihren Vornamen, Nachnamen und Alter oder auch fiktive Werte zu.

### Übung 11:

Erweitern Sie Ihre *variablen.php*, indem Sie folgenden Text ausgeben lassen:

Mein Name ist **vorname nachname** und ich bin **alter** Jahre alt.

Ersetzen Sie die Platzhalter durch die in der vorherigen Übung definierten Variablen, um Ihren Namen und Ihr Alter auszugeben.

## Optionale Übungen

### Übung 12:

Schreiben Sie ein PHP-Skript `wochentage.php`, das die verschiedenen Wochentage in eine einzelne Variable `$wochentage` schreibt. Beginnen Sie mit `$wochentage = 'Montag';` und erweitern Sie die Variable in jeder Code-Zeile um den nächsten Tag, getrennt durch ein Komma. Verwenden Sie den verkürzten Verknüpfungs-Operator (`.=`), um die Wochentage anzuhängen. Am Ende soll folgendes Ergebnis mittels `echo` ausgegeben werden:

Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, Sonntag

### Übung 13:

Schreiben Sie ein PHP-Skript `berechnungen.php`, das eine Variable `$zahl` mit dem Wert 5 belegt. Danach führen Sie mit dieser Variable schrittweise folgende Operationen durch:

- 3 addieren,
- durch 2 teilen,
- 1 subtrahieren,
- mit 7 multiplizieren,
- Modulo 8 durchführen,
- mit 2 potenzieren und geben zum Schluss den berechneten Wert mittels `echo` aus.

### Übung 14:

Schreiben Sie ein PHP-Skript `modulo.php`, das alle Ergebnisse der Berechnungen `13 % $zahl` ausgibt, wobei `$zahl` die Werte zwischen 1 und 13 haben soll.

Das bedeutet, Sie müssen 13 Modulo-Berechnungen durchführen, das jeweilige Ergebnis mit `echo` ausgeben und danach `$zahl` für die nächste Berechnung um eins erhöhen. Teilen Sie vor jedem Ergebnis mit, welcher Modulo hier berechnet wurde.

## Lektion - PHP-Funktionen (entspricht Anfang von Lektion 09 'Funktionen')

### Übung 15:

Finden Sie heraus, welchen Datentyp die Funktion `intval()` zurückliefert.

### Übung 16:

Finden Sie heraus, welchen Datentyp der zweite Parameter von `substr()` haben muss.

### Übung 17:

Überprüfen Sie, wie viele Parameter die Funktion `explode()` hat. Wie viele davon sind optionale Parameter?

## Optionale Übungen

### Übung 18:

Versuchen Sie anhand der Funktionsbeschreibung auf <http://php.net> zu verstehen, welchen Zweck die Funktion `trim()` erfüllt. Denken Sie sich eine praktische Anwendung aus, wo Sie `trim()` in einer Webseite einsetzen könnten. Eine Umsetzung als Code ist jedoch nicht nötig.

### Übung 19:

Versuchen Sie anhand der Funktionsbeschreibung auf <http://php.net> zu verstehen, welchen Zweck die Funktion `htmlspecialchars()` erfüllt. Überlegen Sie sich eine praktische Anwendung für `htmlspecialchars()` in einer Webseite. Eine Umsetzung als Code ist jedoch nicht nötig.

### Übung 20:

Verwenden Sie die Funktion `rand()`, um einen Glückszahlen-Generator zu programmieren. Das Skript soll die Ausgabe **Die heutige Glückszahl ist zahl !** ausgeben, wobei »zahl« eine zufällig erzeugte Zahl zwischen 1 und 50 ist.

## Lektion - PHP-Funktionen (entspricht Anfang von Lektion 09 'Funktionen')

### Übung 21:

Schreiben Sie ein PHP-Skript `variablen_test.php`, das folgende Variablen auf ihren Datentyp prüft:

- `$test1 = 'Hallo';`
- `$test2 = 45 * 3.7;`
- `$test3 = 7 . 5;`

Das Skript soll für jede Variable den Satz ausgeben:

**Der Wert der Variable ist \$testX und dieser hat den Datentyp datentyp !**

### Übung 22:

Schreiben Sie ein PHP-Skript `laengen_test.php`, das ausgibt, wie viele Zeichen ein String enthält, der in der Variablen `$test` hinterlegt wurde. Befüllen Sie die Variable mit einem beliebigen (nicht leeren) String. Das Skript soll ausgeben:

**Der Text in \$test ist XX Zeichen lang.**

## Optionale Übungen

### Übung 23:

Achten Sie nun bei der Ausgabe von `_variablentest.php` auf valides HTML. Fügen Sie alle nötigen HTML-Elemente (`Doctype`, `head`, `body` usw.) hinzu und verlagern Sie jede der drei Ausgaben in einen eigenen `<p>`-Tag.

### Übung 24:

Achten Sie außerdem darauf, dass Sie bei `variablen_test.php` **keinen** statischen Text mit der PHP-Anweisung `echo` ausgeben. Lediglich die Ausgabe des Datentyps darf mit `echo` erfolgen. Beachten Sie dabei die Regeln aus Abschnitt 3.5.1.

### Übung 25:

Führen Sie nun die Verbesserungen auch für das Skript `_laengentest.php` durch.

### Übung 26:

Verbessern Sie das Skript `variablen_test.php`, indem beim ausgegebenen Datentyp nun der erste Buchstabe groß geschrieben wird, also `Integer` statt bisher `integer`. Recherchieren Sie zu diesem Zweck auf <http://php.net> die Syntax der Funktion `ucfirst()`.

## Lektion - Arrays (entspricht Anfang von Lektion 11 'Arrays')

### Übung 27:

Gegeben sei das nachfolgende Array:

```
$sprachen = ['Java', 'Perl', 'Ruby', 'Python', 'PHP', 'JavaScript'];
```

Schreiben Sie ein PHP-Skript `lieblings_sprache.php`, das den Satz ausgibt:

```
Meine liebste Programmiersprache ist xxxx !
```

Den Namen Ihrer Lieblingssprache soll sich das Skript aus dem Array `$sprachen` holen.

Suchen Sie sich Ihre liebste Programmiersprache aus.

### Übung 28:

Überprüfen Sie, ob das Skript `_lieblings_sprache.php` `_valides` HTML erzeugt und keinen statischen Text mit `echo` ausgibt. Von nun an gelten die Regeln aus Abschnitt 3.5 für alle zukünftigen Aufgaben und werden nicht mehr gesondert erwähnt.

### Übung 29:

Erzeugen Sie in der Datei `ich.php` ein assoziatives Array mit einigen persönlichen Daten. Legen Sie Schlüssel für Ihren Vornamen, Nachnamen, Alter und E-Mail-Adresse an.

### Übung 30:

Geben Sie Ihre persönlichen Daten in `ich.php` nun formatiert in einer HTML-Seite aus. Verwenden Sie eine Tabelle zur Ausgabe.

## Optionale Übungen

### Übung 31:

Legen Sie in *ich.php* einen zweiten Datensatz für eine andere Person an. Geben Sie diesen ebenfalls formatiert in der Tabelle aus. Bauen Sie die Tabelle gegebenenfalls um, sofern sich die bisherige Aufteilung für zwei Personen nicht wirklich eignet.

### Übung 32:

Die Tabelle in *ich.php* soll nun (wenn noch nicht vorhanden) für jede Spalte einen `<th>`-Tag besitzen, der die Spalte beschriftet. Den Inhalt der Tags gewinnen Sie aus den Schlüsseln des Arrays für die erste Person. Verwenden Sie die Funktion `array_keys()`, um an sie heranzukommen.

### Übung 33:

Die ersten Buchstaben der Spalten-Beschriftungen in *ich.php* sollen großgeschrieben sein. Wenn Sie also den Array-Schlüssel `vorname` haben, soll die Beschriftung `Vorname` lauten. Verwenden Sie hierfür die Funktion `ucfirst()`.

## Lektion - Formulare und Links (entspricht Anfang von Lektion 12 'Formulare')

### Übung 34:

Schreiben Sie ein Skript *hallo.php*, dem Sie einen URL-Parameter `name` übergeben können, der dann wie folgt auf der Seite ausgegeben wird:

Hallo name . Schön, dass Sie wieder da sind.

### Übung 35:

Ändern Sie *hallo.php*, dass Sie nun die Parameter `vorname` und `nachname` getrennt übergeben können und der volle Name als Begrüßung ausgegeben wird.

### Übung 36:

Schreiben Sie ein Skript *eintragen.php*, das ein HTML-Formular mit folgenden Elementen anzeigt:

- Textfeld: `name`
- Textfeld: `email`
- Textarea: `inhalt` Das Formular soll per POST an die (noch nicht existierende) Seite *auslesen.php* versendet werden.

### Übung 37:

Schreiben Sie das Skript *auslesen.php*, das die Eingaben aus *eintragen.php* formatiert anzeigt (z.B. als Tabelle).

## Optionale Übungen

### Übung 38:

Der Besucher von *eintragen.php* soll auch seine Lieblingsfarbe auswählen können. Programmieren Sie eine Auswahl aus 10 Farben mit Radiobuttons. Vergessen Sie nicht, das Ergebnis auch auf *auslesen.php* anzuzeigen.

### Übung 39:

Ihre Besucher haben sich beschwert, dass Sie nur eine Lieblingsfarbe auswählen dürfen (Stichwort Mehrfachauswahl). Modifizieren Sie *eintragen.php* so, dass dort nun Checkboxes statt Radiobuttons verwendet werden.

**Hinweis:** Nach der Änderung passt der Code im Skript *auslesen.php* nicht mehr zum Formular. Ignorieren Sie dies bitte zunächst.

### Übung 40:

Die Lieblingsfarben sollen auf *auslesen.php* als eine kommaseparierte Liste angezeigt werden, also zum Beispiel *rot, blau, grün*. Verwenden Sie die Funktion *implode()*, um das Problem elegant zu lösen.

## Lektion - Funktionsentwurf (entspricht Lektion 09 'Funktionen')

### Übung 41:

Schreiben Sie eine Funktion `zeigeDaten()`, die einen Namen und ein Geburtsdatum ausgibt. Rufen Sie die Funktion zweimal auf.

### Übung 42:

Schreiben Sie eine Funktion namens `count()`, die die Zahlen von 10 bis 0 rückwärts ausgibt.

### Übung 43:

Rufen Sie Ihre neue Funktion auf und beobachten Sie, was passiert.

**Hinweis:** Seit Version 4 ist im Mozilla Firefox die Menüleiste standardmäßig deaktiviert. Sie kann jedoch durch Drücken der (linken) Alt-Taste eingeblendet werden. Alternativ lässt sich die Datei auch einfach auf den Anzeigebereich des Browsers ziehen, wodurch sie im Browser geöffnet wird.

### Übung 44:

Interpretieren Sie die Fehlermeldung und sorgen Sie dafür, dass die Funktion korrekt aufgerufen werden kann.

## Lektion - Strukturierung von PHP-Skripten (entspricht Lektion 13 'Strukturierung von PHP-Skripten')

### Übung 45:

Prüfen Sie alle bisher von Ihnen erstellten Lösungen, ob Sie den eben genannten Coderichtlinien entsprechen. Achten Sie ebenfalls darauf, von Ihnen geschriebene Funktionen in eigene Dateien auszulagern und diese per `require_once` einzubinden.

**Hinweis:** Achten Sie darauf, dass Sie auch bei allen zukünftigen Aufgaben die vorgegebenen Coderichtlinien einhalten.

### Optionale Übungen

### Übung 46:

Führen Sie dieselben Verbesserungen auch für die optionalen Übungen durch.



## Lektion Verzweigungen (entspricht Lektion 10 'Conditionals und Logik — wir verzweigen unseren Code')

### Übung 47:

Nehmen Sie das Skript `hallo.php` aus den Aufgaben in Lektion 8 als Basis für die folgenden Überarbeitungen. Ändern Sie das Skript `hallo.php` so ab, dass nun der Text `Wer sind Sie?` ausgegeben wird, wenn einer der beiden oder beide URL-Parameter nicht übergeben wurden. Verwenden Sie dafür eine `if-else`-Anweisung und `isset()`.

### Übung 48:

Verbessern Sie das Skript `hallo.php`, sodass es sich gezielt darüber beschwert, welcher Parameter fehlt. Wenn beide fehlen, soll der Text `Vor- und Nachname fehlt!` ausgegeben werden, fehlt nur einer, wird der Text `Vorname fehlt!` oder `Nachname fehlt!` ausgegeben. Verwenden Sie zu diesem Zweck eine `if-elseif-else`-Anweisung.

Prüfen Sie jeden Fall der `if`-Anweisung, indem Sie im Browser die Parameter angeben oder weglassen.

### Übung 49:

Schreiben Sie ein Skript `bewertung.php`, das ähnlich funktioniert wie das aus Abschnitt 11.4 (mit `switch`-Anweisung und `default`-Fall). Dieses Mal sollen allerdings die Noten von 1 bis 6 als Zahlen die verschiedenen Fälle der `switch`-Anweisung bilden.

Sichern Sie Ihren Code zudem mittels Null-Coalesce-Operator gegen das Fehlen des URL-Parameters `note` ab.

## Lektion Schleifen (entspricht Lektion 14 'Schleifen')

### Übung 50:

Schreiben Sie ein PHP-Skript `variablen.php`, das die Variablen `$vorname`, `$nachname` und `$alter` definiert. Weisen Sie den Variablen Ihren Vornamen, Nachnamen und Alter oder auch fiktive Werte zu.

### Übung 51:

Erzeugen Sie in einem Skript `ich.php` ein assoziatives Array `$ich`, das einige Daten, wie `vorname`, `nachname`, `alter`, `geschlecht` über Sie enthält. Geben Sie dieses formatiert in einer zweispaltigen HTML-Tabelle aus, wobei die Schlüssel in der linken, die Werte in der rechten Spalte stehen.

### Übung 52:

Ändern Sie das Skript `sprachen.php`. Anstatt der `foreach`-Schleife soll es nun eine `for`-Schleife verwenden.

**Übung 53:**

Schreiben Sie eine weitere Version von der *sprachen.php*, welche eine `while`-Schleife verwendet.

**Optionale Übungen****Übung 54:**

Verbessern Sie die *ich.php*. Die linke Spalte der Tabelle soll nun `<th>` anstatt `<td>`-Tags verwenden. Wenn Sie alles richtig gemacht haben, müssen Sie genau zwei Zeichen in der *ich.php* ändern.

**Übung 55:**

Momentan werden die Schlüssel in der linken Spalte kleingeschrieben ausgegeben, also z.B. `vorname` statt `Vorname`. Verwenden Sie die Funktion `ucfirst()`, um das Problem zu lösen.

**Übung 56:**

Schreiben Sie eine Version von *ich.php*, die statt der `foreach`-Schleife eine `for`-Schleife verwendet. Verwenden Sie hierfür die bereits bekannten Funktionen `array_keys()` und `array_values()`.

**Übung 57:**

Schreiben Sie eine weitere Version von *ich.php*, die eine `while`-Schleife verwendet.

## Lektion Funktionen mit Parametern und Rückgabewerten (entspricht Lektion tw. 09 'Funktionen' und tw 16 'anonyme Funktionen')

**Übung 58:**

Schreiben Sie eine Funktion `druckeMehrfach()`, die einen String mehrfach mit `echo` ausgibt. Der erste Parameter `$string` ist der Text, der ausgegeben werden soll, der zweite `$anzahl` bestimmt, wie oft der Text ausgegeben wird.

**Übung 59:**

Rufen Sie die Funktion `druckeMehrfach()` mit verschiedenen Parameterkombinationen auf.

**Übung 60:**

Der zweite Parameter von `druckeMehrfach()` soll optional werden. Wenn er weggelassen wird, soll der Text dreimal ausgegeben werden.

**Übung 61:**

Ändern Sie `druckeMehrfach()` so ab, dass die Funktion den resultierenden Text nun nicht mehr direkt ausgibt, sondern als String zurückgibt. Dieser Rückgabewert soll natürlich ausgegeben werden.

## Optionale Übungen

### Übung 62:

Schreiben Sie eine Funktion `zaehleWoerter()`, die die Anzahl von Wörtern in einem String zählt und als Integer zurückgibt. Zur Vereinfachung nehmen wir an, dass Wörter immer durch ein Leerzeichen getrennt sind. Verwenden Sie die Funktion `explode()`, um den String an den Leerzeichen zu trennen und `count()`, um die Ergebnisse zu zählen.

### Übung 63:

Schreiben Sie eine Variante von `zaehleWoerter()` namens `teileInWoerter()`, die nicht die Anzahl der Wörter, sondern die einzelnen Wörter als Rückgabewert hat. Beachten Sie, dass Sie nun mehrere Werte als Rückgabewert haben. Geben Sie die Wörter aus dem Rückgabewert kommasepariert aus.

## Lektion Mehrdimensionale Arrays (entspricht Lektion 11.2 'Mehrdimensionale Arrays')

### Übung 64:

Schreiben Sie ein Skript `filme.php`, das eine Variable `$filme` enthält. `$filme` soll ein zweidimensionales Array sein. Die erste Ebene ist ein numerisches Array. Jeder Wert dieses Arrays ist ein assoziatives Array, das einen Film darstellt. Das innere Array hat die Schlüssel `titel`, `dauer`, `genre` und `regisseur`. Suchen Sie sich drei beliebige Filme aus und tragen Sie diese in das Array `$filme` ein.

### Übung 65:

Geben Sie das Array `$filme` als HTML-Tabelle aus. Achten Sie auf valides HTML.

### Übung 66:

Das Feld `regisseur` ist bisher als ein einfacher String realisiert. Nun soll es ein assoziatives Array mit den Schlüsseln `vorname`, `nachname` und `geschlecht` sein. Ändern Sie die Datenstruktur dementsprechend und passen Sie Ihre Filme an.

**Hinweis:** Ignorieren Sie bitte zunächst die derzeit fehlerhafte Datenausgabe.

### Übung 67:

Derzeit erfolgt in der HTML-Tabelle im Feld `regisseur` keine korrekte Datenausgabe mehr. Passen Sie deswegen auch diese Ausgabe an die neue Datenstruktur an, wobei alle drei Werte ausgegeben werden sollen.

## Optionale Übungen

### Übung 68:

Jeder Film erhält nun ein neues Feld `darsteller`. Dies ist ein numerisches Array, das die Namen der Darsteller des jeweiligen Films enthält. Beschränken Sie sich pro Film auf zwei bis drei Darsteller.

**Übung 69:**

Fügen Sie auch der HTML-Tabelle eine neue Spalte **Darsteller** hinzu, wo die einzelnen Werte kommasepariert ausgegeben werden.