

↓  
Informatik 8bi Schuljahr 2018/2019 als PDF exportieren

# Informatik 8. Klasse - Schuljahr 2018/19

## Lehrinhalte

- [Lehrplaninhalte](#)

[Remote-Zugriff auf Schulserver](#)

## Kapitel

- [1\) PHP](#)

## Leistungsbeurteilung

- **Schularbeiten (SA)**
  - 2x SA (2h) pro Semester
- **Mitarbeit (MA)**
  - Aktive Mitarbeit im Unterricht (aMA)
  - Mündliche Stundenwiederholungen (mMA)
  - Schriftliche Stundenwiederholungen (sMA)
- **Praktische Arbeiten (PA)**
  - 1x praktischer Arbeitsauftrag pro Woche
- [Aktueller Leistungsstand](#)

**Stoff für die 2. Schularbeit in Informatik - 8BI - ???.???.????**

**Stoff für die 1. Schularbeit in Informatik - 8BI - ???.???.????**

From:  
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819)

Last update: **2018/09/11 13:40**



# Was wird in der 8. Klasse gemacht?

## 7. Semester

8. Klasse (3 Stunden, eine 2- oder 3-stündige Schularbeit)

### Sicherung der Nachhaltigkeit

- Notwendiges Vorwissens für die Kompetenzbereiche dieses Moduls wiederholen und aktivieren
- Grundlagen für die Kompetenzbereiche dieses Moduls ergänzen und bereitstellen

### Gesellschaftliche Aspekte der Informationstechnologie

#### Berufliche Perspektiven

- Informatikberufe und Einsatzmöglichkeiten der Informatik in verschiedenen Berufsfeldern benennen und einschätzen können.

#### Verantwortung, Datenschutz und Datensicherheit

- Die Entwicklung der Informatik beschreiben und bewerten können.
- Die Bedeutung von Informatik in der Gesellschaft beschreiben, die Auswirkungen auf die Einzelnen und die Gesellschaft einschätzen und Vor- und Nachteile an konkreten Beispielen abwägen können.
- Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennen und anwenden können.

### Informatiksysteme - Hardware, Betriebssysteme und Vernetzung

#### Technische Grundlagen und Funktionsweisen (Hardware)

- Aktualisierungen im Zusammenhang mit der Hardware kennen

#### Betriebssysteme (Windows, Linux, MacOS, iOS, Android)

- Aktualisierungen im Zusammenhang mit Betriebssystemen kennen

## **Mensch-Maschine-Schnittstelle**

- Maßnahmen für einen barrierefreien zu Zugang Informatik-Systemen angeben können

## **Algorithmik und Programmierung**

### **Algorithmen und Datenstrukturen**

- Algorithmen erklären, entwerfen, darstellen können.
- Datenstruktur Bäume kennen und einsetzen können
- Rekursionen kennen und einsetzen können
- Dynamische Programmierung kennen
- Algorithmen mit Bäumen erstellen können
- Algorithmen mit Rekursionen erstellen können

### **Programmierung (Objektorientierte visuelle Programmiersprache)**

- Algorithmen in einer Programmiersprache implementieren können
- Datenbank Anwendungen programmieren können
- Programme mit Bäume erstellen können
- Rekursive Algorithmen erstellen können

## **Angewandte Informatik, Datenbanksysteme und Internet**

### **Datenmodelle und Datenbanksysteme**

- Einen Webserver konfigurieren können
- Internetdienste (Mail-Server, Web-Server, FTP-Server) in ihrer Funktionsweise verstehen und einsetzen können

### **Web-Techniken (Content-Management-Systeme)**

- Content-Management-Systeme installieren können
- Rechte bei Content-Management-Systemen vergeben können
- Oberfläche bei Content-Management-Systemen einstellen und anpassen können
- Die Funktionsweise durch Plugins und Module erweitern können

## 8. Semester

8. Klasse (3 Stunden, eine 3- oder 4-stündige Schularbeit)

### Sicherung der Nachhaltigkeit

**Wiederholen, Vertiefen von Fähigkeiten und Vernetzen von Inhalten, um einen umfassenden Überblick über die Zusammenhänge unterschiedlicher informatischer Teilgebiete zu gewinnen.**

### Inhalt und Umfang der Klausurarbeit im Prüfungsgebiet Informatik

(1) Im Rahmen der Klausurarbeit im Prüfungsgebiet „Informatik“ ist den Prüfungskandidatinnen und Prüfungskandidaten eine Aufgabenstellung mit drei bis fünf voneinander unabhängigen Aufgaben, die in Teilaufgaben gegliedert sein können, aus unterschiedlichen Kompetenzbereichen – Gesellschaftliche Aspekte der Informationstechnologie, Informatiksysteme, Algorithmik und Programmieren sowie Angewandte Informatik, Datenbank und Internet - mit ausgewogenen Anforderungen schriftlich vorzulegen. Mindestens eine Aufgabe hat anwendungsorientierten Charakter zu haben. Für die Bearbeitung zumindest einer Aufgabe muss Computertechnologie eingesetzt werden. (2) Die Arbeitszeit hat 270 Minuten zu betragen.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:0\\_lehrplaninhalte](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:0_lehrplaninhalte)

Last update: **2018/09/10 14:48**



# PHP

## Was ist PHP?

- [Was ist PHP](#)
- [How to use PHP](#)

## Grundlegende Sprachelemente

- [Kommentare](#)
- [Ausgabe](#)
- [ÜBUNG 1](#)

## Variablen und Operatoren

- [Variablen und Operatoren](#)

## Interaktive Webseiten

- [Formulare](#)
- [ÜBUNG 2](#)

## Anführungszeichen

- [Verwenden von Anführungszeichen](#)

## Kontrollstrukturen

- [Vergleichsoperatoren](#)
- [if-Anweisung](#)
- [switch-Anweisung](#)
- [ÜBUNG 3-7](#)

## Schleifen

- [for-Schleife](#)
- [while-Schleife](#)
- [do-while-Schleife](#)
- [ÜBUNG 8-11](#)

## Felder

- [Grundlagen zu Felder](#)
- [Indizierte Felder](#)
- [Assoziative Felder](#)
- [Mehrdimensionale Felder](#)
- [Weitere Informationen zu Feldern](#)
- [ÜBUNG 12-19](#)
- [KONTROLLFRAGEN](#)

## Formulare

- [Hidden-Feld](#)
- [Textarea](#)
- [Checkbox](#)
- [Radio-Button](#)
- [Auswahllisten](#)
- [ÜBUNG 20](#)

## Externe Dateien

- [Externe Datei nutzen](#)
- [Dateien öffnen,lesen und schließen](#)
- [Aus Dateien lesen](#)
- [In Dateien schreiben](#)
- [ÜBUNG 21 Gästebuch](#)
- [ÜBUNG 22 Besucher zählen](#)
- [ÜBUNG 23-26](#)

## Datum und Zeit

- [Datum und Zeit](#)

## Sessions

- [Grundlagen zu Sessions](#)
- [Einschalten von Sessions](#)
- [Sessiondaten speichern](#)
- [Sessiondaten abrufen](#)
- [Sessiondaten löschen](#)
- [ÜBUNG 27](#)

## Cookies

- Grundlagen
- ÜBUNG 28-33

## Funktionen

- Funktionen erstellen und aufrufen
- Funktionen verwenden
- Gültigkeitsbereich von Variablen
- PHP-Dateien einbinden
- Andere Dateitypen einbinden
- ÜBUNG 34-35

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1)

Last update: **2018/09/11 14:33**



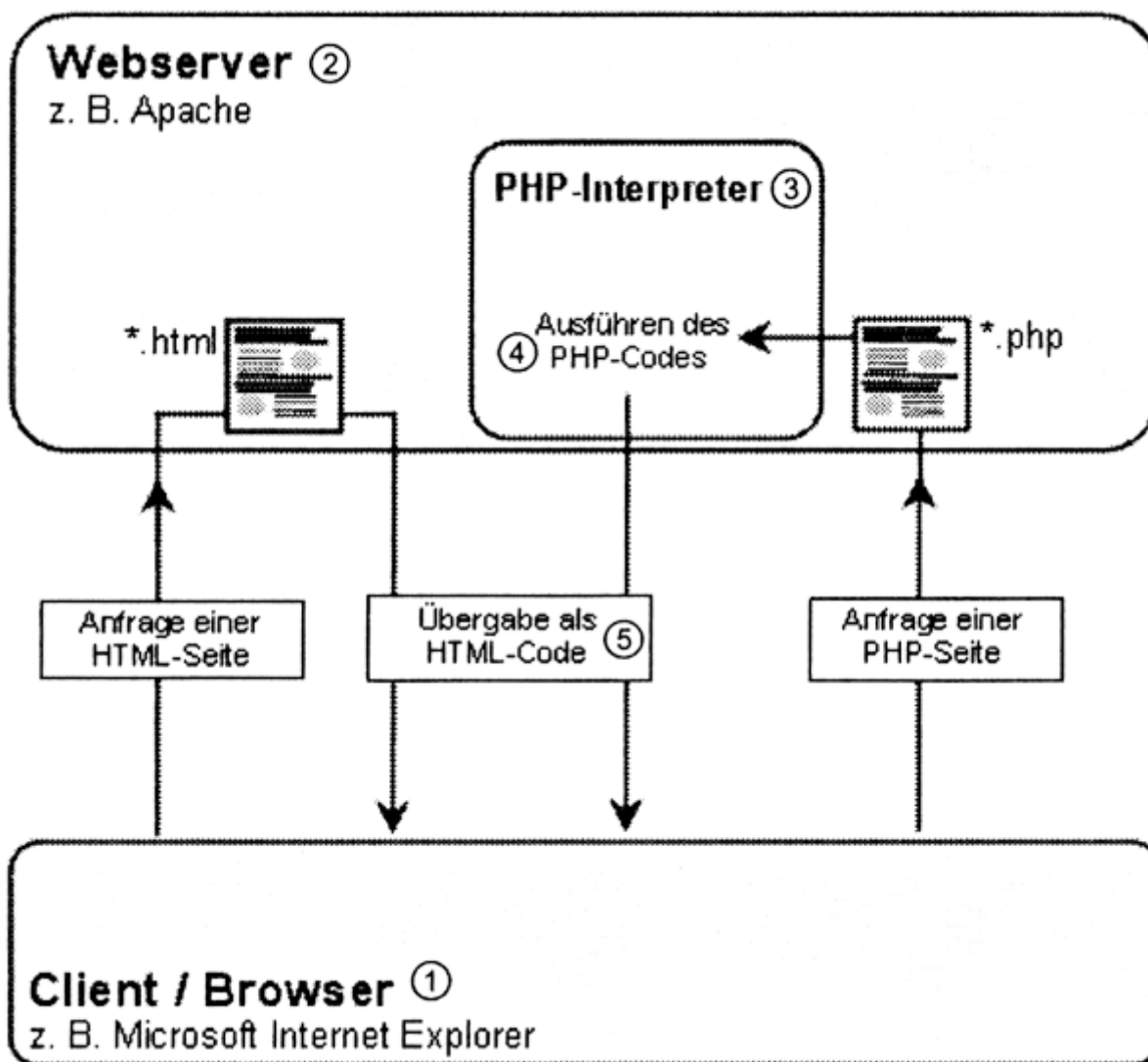
## Was ist PHP?

**PHP (= PHP Hypertext Preprocessor) ... Skriptsprache** für Internet, wird am **Webserver** ausgeführt.

PHP, die Kurzform von Personal HomePage oder auch PHP Hypertext Preprocessor, ist eine Skriptsprache, die speziell für den **Einsatz im Internet** entwickelt wurde. Die auszuführenden Anweisungen können **direkt in den HTML-Code integriert** werden.

Bei einer mit **HTML erstellten Webseite ist keine Dynamik** möglich. Mithilfe von **PHP** können auf einer Webseite **Interaktionen** eingebaut oder die Seite in irgendeiner Form, beispielsweise durch **Reaktionen auf Eingaben**, angepasst werden.

Öffnet der Betrachter in seinem Browser (1) eine solche Webseite über einen **PHP-fähigen Webserver** (2), werden die Anweisungen von PHP interpretiert (3), ausgeführt (4) und das Ergebnis als HTML-Code zurückgesendet (5). Ein klarer Vorteil dabei ist, dass der PHP-Quellcode dem Betrachter nicht zugänglich ist. Er sieht im Browser nur den zurückgelieferten HTML-Code.

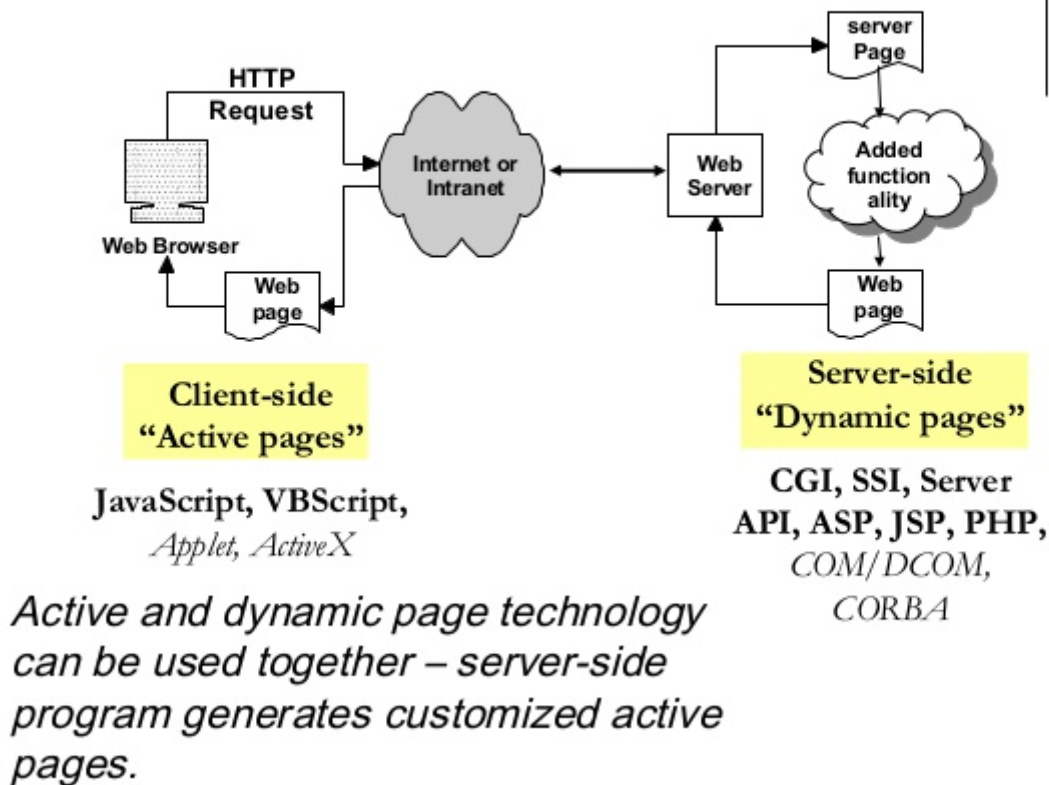


Um dynamische Webseiten zu erstellen und später auf Datenbanken zugreifen zu können, benötigt man neben PHP einen Webserver und einen Datenbankserver. Unter Windows kann man einen

lokalen Webserver installieren. Als Entwicklungssoftware hat sich XAMPP sehr bewährt.

## Unterschied zu Javascript

# Where does PHP fit ?



## Download von XAMPP

- <http://www.apachefriends.org/de/xampp.html>
- localhost greift auf den Ordner .../htdocs zu. In diesen Ordner müssen die html- bzw. php-Dateien gespeichert werden.

From:  
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_01](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_01)

Last update: **2018/09/11 14:10**



# How to use PHP

## PHP-Befehle einfügen

PHP-Code wird direkt in den HTML-Code eingebettet. PHP-Blöcke können vollständig im Dokumentkopf `<head>` oder im Dokumenttrumpf `<body>` stehen. Weiterhin hat man die Möglichkeit, innerhalb einer Datei mehrmals zwischen PHP- und HTML-Code zu wechseln.

## Dateinamenerweiterungen mit PHP-Code

Damit der Webserver erkennt, dass es sich um eine Datei mit einem PHP-Aufruf handelt, werden die Dokumente mit der Dateinamenerweiterung `.php` oder `.php5` gespeichert.

## XML-Schreibweise

Der auszuführende PHP-Code wird zwischen dem öffnenden Tag `<?php` und dem schließenden Tag `?>` geschrieben

```
<?php
    hier steht ein (oder mehrere) PHP-Befehl(e)
?>
```

Diese Schreibweise ist am gebräuchlichsten. Alternativ dazu kann man auch einen Skript-Tag verwenden. Diese Schreibweise entspricht im Wesentlichen der JavaScript- bzw. CSS-Angabe:

```
<script language="php">
    hier steht ein (oder mehrere) PHP-Befehl(e)
</script>
```

## PHP-Anweisungen voneinander trennen

Jede PHP-Anweisung wird durch einen Strichpunkt beendet.

```
<?php
    PHP-Code;
    PHP-Code;
?>
```

Es empfiehlt sich, nach einer Anweisung in einer neuen Zeile weiterzuschreiben.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_02](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_02)



Last update: **2018/09/11 13:46**

# Kommentare einfügen

- Kommentare werden nicht vom Interpreter ausgeführt.
- Kommentare dienen zur näheren Beschreibung des Quellcodes

## Syntax und Bedeutung der Kommentare

```
<html>
<body>
<?php
    echo "Für den Anfang: "; //einzeiliger Kommentar am Ende eines PHP-Codes
    // einzeiliger K0mmentar

    /*
        mehrzeilige Kommentare sind ebenfalls möglich,
        um den Quellcode ausführlicher zu beschreiben
    */
    echo "Hallo Welt!";
?>
</body>
</html>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_03](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_03)

Last update: **2018/09/11 13:47**



# Daten im Browser ausgeben

## Befehl "echo"

Da PHP eine eingebettete Skriptsprache ist, die vom Server ausgeführt wird, bleibt der Quellcode im Gegensatz zu HTML für den Client unsichtbar. Zeichenketten und Daten werden über den Befehl echo auf dem Bildschirm ausgegeben.

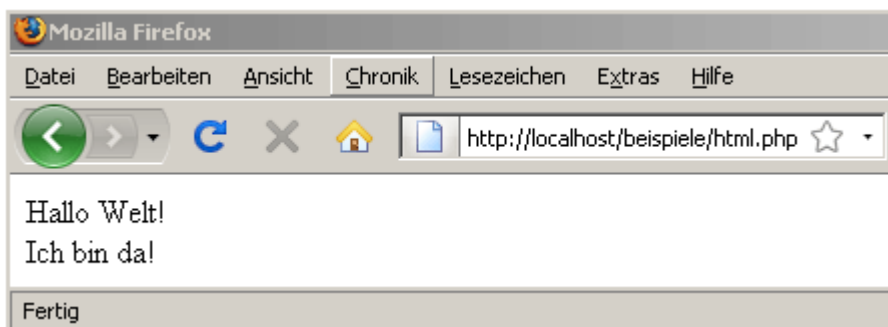
```
echo "Hallo Welt";
```

## Umgang mit HTML-Syntax bei der Ausgabe

Um Zeichenketten anzuzeigen, werden diese direkt an den Browser weitergegeben. Daher kann man innerhalb des Befehls echo auch jeden HTML-Tag (z.B. <br>) angeben. Dieser wird dann vom Browser als normaler HTML-Befehl interpretiert.

```
<html>
<body>
<?php
    echo "Hallo Welt! <br>"; //Der HTML-Tag <br> wird vom Browser korrekt
interpretiert.
    echo "Ich bin da!";
?>
</body>
</html>
```

## Screenshot Browser



Im Screenshot wird der HTML-Tag <br> richtig interpretiert. Der Browser fügt eine Leerzeile ein. Der Kommentar wird natürlich nicht angezeigt!

## Quelltext HTML

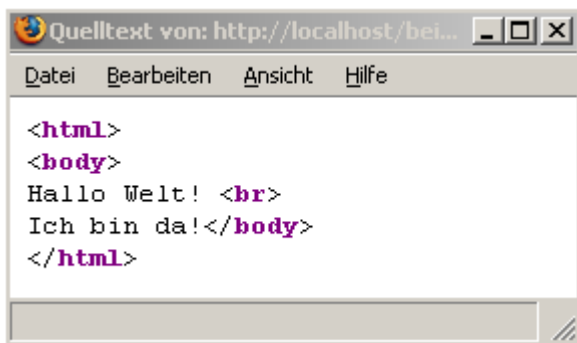


Wie man sieht, wird der PHP-Kommentar auch im HTML-Quelltext nicht angezeigt. Weiters erscheint der übersetzte HTML-Code in einer Zeile.

Der HTML-Quellcode sollte ebenfalls wie der PHP-Quellcode übersichtlich erscheinen. Mittels `\n` kann man in PHP erwirken, dass im HTML-Quelltext eine Leerzeile eingefügt wird. `\n` hat keinen Einfluss auf das Erscheinungsbild im Browser.

```
<html>
<body>
<?php
    echo "Hallo Welt! <br>\n"; // \n bewirkt eine Leerzeile im HTML-Quellcode!
    echo "Ich bin da!";
?>
</body>
</html>
```

Das Ergebnis im HTML-Quelltext sieht wie folgt aus:



## Sonderzeichen ausgeben

Wenn man Hochkommas oder einen Backslash ausgeben möchte, setzt man vor den Zeichen einen Backslash:

```
<?php
    echo "<p>Der \"Erfinder\" von PHP war Rasmus Lerdorf. </p>";
    echo "Das Programm wird installiert im Verzeichnis C:\\Programme";
?>
```

## Zeilenumbruch im Quelltext

Möchte man im Quelltext einen Zeilenumbruch einfügen, verwendet man \n

```
<?php
    echo "Das ist ein Text<br>\n";
    echo "Das ist noch ein Text";
?>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_04](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_04)



Last update: **2018/09/11 13:48**

# Aufgabe 1

Schreibe ein einfaches PHP-Skript, z.B. eine einfache Webseite mit Informationen über dich (Name, Wohnort, Hobbies, ...)

Folgende Elemente sollen dabei vorkommen (eingebunden in PHP-Code):

- Kommentare
- Sonderzeichen, wie z.B. Anführungsstriche
- Farbformatierungen
- Listen
- Tabellen
- ...

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_05](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_05)

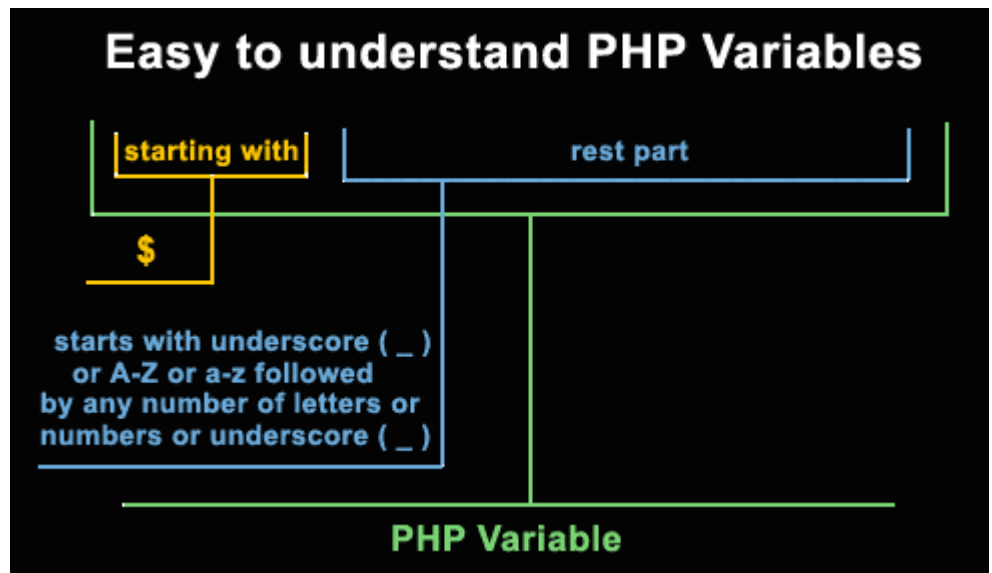
Last update: **2018/09/11 13:50**



# Variablen und Operatoren

## Variablen

Variablen müssen im Gegensatz zu vielen Programmiersprachen nicht extra deklariert werden. Sie werden mit einem vorangestellten Dollar- Zeichen gekennzeichnet und beim ersten Aufruf definiert.



Folgende Datentypen werden von PHP unterstützt:

Datentyp	Bezeichnung	Beispiel
<b>Ganze Zahlen</b>	<b>integer</b>	1911 oder -1911
<b>Dezimalzahl</b>	<b>double</b>	19.11 oder -19.11
<b>Zeichenketten</b>	<b>string</b>	„Kette von Zeichen“ oder 'Kette von Zeichen'
<b>Felder</b> (ein- oder mehrdimensional)	<b>array</b>	(„Frankfurt“, „Berlin“, „Zürich“)
<b>Objekte</b>	<b>object</b>	Verweis auf eine bestimmte Variable

### Namensgebung bei Variablen

Bei der Vergabe von Namen für die Variablen gibt es folgende Regeln. Der Name einer Variablen

- muss mit dem Dollarzeichen beginnen und darf kein Leerzeichen enthalten
- darf nur aus Buchstaben und Ziffern bestehen, aber keine Umlaute oder „ß“ enthalten
- muss als zweites Zeichen einen Buchstaben enthalten
- darf keine Sonderzeichen außer dem Unterstrich „\_“ enthalten
- kann Groß- oder Kleinbuchstaben enthalten, wobei zwischen Groß- und Kleinschreibung unterschieden wird (\$PrimZahl ist nicht gleich \$primzahl)
- darf nicht identisch sein mit einem so genannten reservierten Wort (and, break, case, class, continue, default, do, else, elseif, empty, endfor, endif)

# Variablen und Operatoren für Zahlen

## Numerische Datentypen

Die numerischen Datentypen werden in Ganzzahl-Datentypen `integer` und Fließkommazahl-Datentypen `double` unterteilt.

### Beispiel: `preis.php`

```
<?php
$preis_apfel = 2.50;
$menge = 4;
$gesamtpreis = $preis_apfel * $menge;
echo $gesamtpreis;
?>
```

## Arithmetische Operatoren

Operator	Name	Bedeutung
+	Addition	<code>\$a + \$b</code> ergibt die Summe
-	Subtraktion	<code>\$a - \$b</code> ergibt die Differenz
*	Multiplikation	<code>\$a * \$b</code> ergibt das Produkt
/	Division	<code>\$a / \$b</code> ergibt den Quotienten
%	Modulo	<code>\$a % \$b</code> ist der Rest der ganzzahligen Division von <code>\$a / \$b</code>
++	Prä-, bzw. Postinkrement	<code>\$a++</code> ( <code>++\$a</code> ) erhöht die Variable <code>a</code> um 1 nach (vor) der weiteren Verwendung
--	Prä-, bzw. Postdekrement	<code>\$a--</code> ( <code>--\$a</code> ) verringert die Variable <code>a</code> um 1 nach (vor) der weiteren Verwendung

### Beispiel: `berechnung.php`

```
<?php
$preis_apfel = 2.50;
$menge1 = 4;
$menge2 = 10;
$menge3 = 15;
$gesamtpreis = $preis_apfel * ($menge1 + $menge2 + $menge3);
echo $gesamtpreis;
?>
```

# Variablen und Operatoren für Zeichenketten

## Zeichenkettenoperator

Mittels Zeichenkettenoperator „.“ können mehrere Zeichenketten miteinander verknüpft werden:

```
<?php
$a = "Hallo ";
$b = "Welt!";
$c = $a.$b;
echo $c;
?>
```

Ergebnis: Hallo Welt!

## Typumwandlung

Der Datentyp einer Variablen wird durch die erste Wertzuweisung bestimmt.

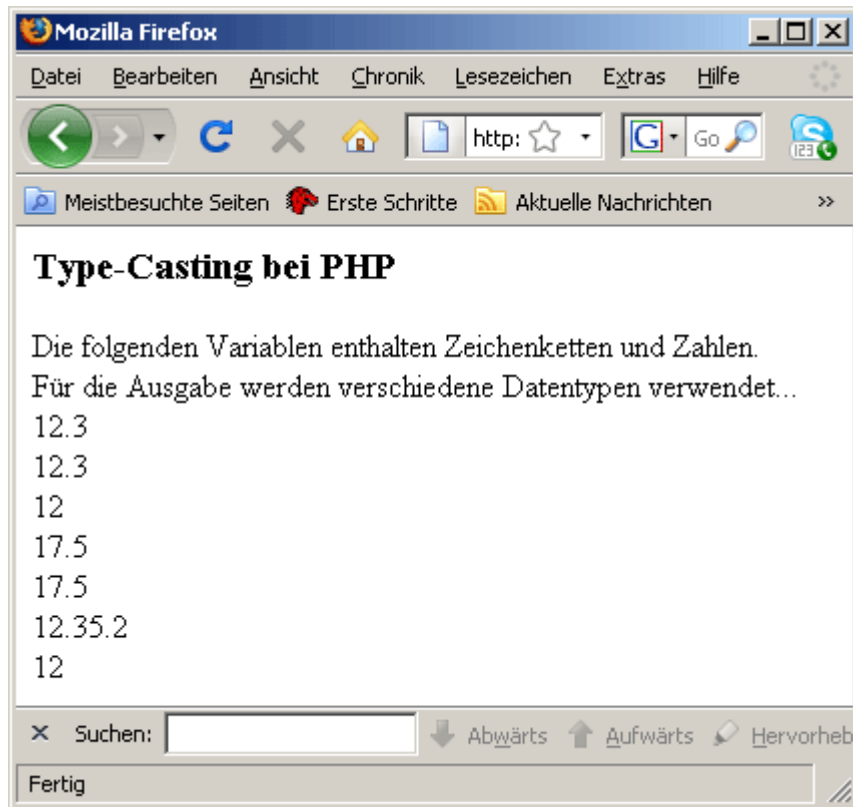
Durch eine vorangestellte Typendeklaration kann dieser Datentyp geändert werden:

<h3>Type-Casting bei PHP</h3>

Die folgenden Variablen enthalten Zeichenketten und Zahlen.<br>

F&uuml;r die Ausgabe werden verschiedene Datentypen verwendet...<br>

```
<?php
$a = 12.3;
$b = 5.2;
echo $a . "<br>\n\t";
echo (string) $a . "<br>";
echo (int) $a . "<br>";
echo (string) ($a + $b) . "<br>";
echo (string) $a + (string) $b . "<br>";
echo (string) $a . (string) $b . "<br>";
echo (int) ( (string) $a . (string) $b );
?>
```



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_06](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_06)



Last update: **2018/09/11 13:56**

# Formulare

Mit PHP kann man **interaktive Webseiten** erstellen, bei denen **Benutzereingaben aus Formularen durch PHP ausgewertet** werden. Die **Darstellung der Formulare** wird mit **HTML-Code** umgesetzt. Die Daten aus dem Formular werden dabei **per HTTP** mit den **Methoden POST oder GET** an das **auswertende PHP-Programm** übermittelt.



## Methoden der Datenübertragung

### POST

- Angabe von `method="POST"` im einleitenden HTML-Tag `<form>`
- Formulardaten werden direkt an die Webadresse versendet.
- Daten können nicht in der Adresszeile des Browsers manipuliert werden. Optionen sind nur über das Formular möglich.
- Daten werden nicht im Logfile des Servers gespeichert.
- Formulardaten sind nicht im Verlauf des Browsers sichtbar.
- Längere Texte oder Daten können in Formularen übermittelt werden, da keine Beschränkung der Größe besteht.

### GET

- Angabe von `method="GET"` im einleitenden HTML-Tag `<form>`
- Formulardaten werden in der URL übermittelt und werden dann durch den Server in einer speziellen Umgebungsvariablen zwischengespeichert.

- Daten sind in der Adresszeile des Browsers änderbar, ohne das Formular erneut ausfüllen zu müssen.
- Der Aufruf des Skripts mit Angabe der Daten kann als Favorit abgelegt werden.
- Daten sind auf ca. 2KB beschränkt.

Da bei der Methode GET die Daten über die URL an den Server weitergegeben werden, ist es möglich, die Daten zu manipulieren. Die sicherere und am häufigsten genutzte Methode ist daher die Methode POST.

## Formulare auswerten mit PHP

### Formulardaten eingeben

Zur Eingabe von Formulardaten, genügt eine HTML-Datei, in der ein Formular integriert ist:

#### formular\_eingabe.html

[formular\\_eingabe.html](#)

```
<html>
<body>
<p>Bitte füllen Sie die nachfolgenden Eingabefelder aus:</p>
<form action="formular_auswertung.php" method="POST">
<pre>
<p>
Vorname: <input type="Text" name="vorname"><br>
Nachname: <input type="Text" name="nachname"><br>
Wohnort: <input type="Text" name="ort"><br>
</p>
<input type="Submit" value="Abschicken">
<input type="Reset" value="Zurücksetzen">
</pre>
</form>
</body>
</html>
```

### Formulardaten übertragen

Klickt man auf den Button „Abschicken“, wird das in action angegebene Skript (formular\_auswertung.php) gestartet.

### Formulardaten auswerten

Die mit der Methode POST übermittelten Daten wird mit dem Befehl `$_POST[„Parameter“]` angesprochen. Der Name des Eingabefeldes im Formular (z.B. `name=„vorname“`) wird automatisch Element der Variablen `$_POST` (z.B. `$_POST[„vorname“]`) Der Eintrag des Eingabefeldes im Formular wird bei der Auswertung zum Wert der Variablen, z.B. `$_POST[„vorname“] = „Max“`.

## formular\_auswertung.php

[formular\\_auswertung.php](#)

```
<?php
    echo "<p>Folgende Daten wurden übermittelt:</p>\n";
    echo "Vorname: " . $_POST["vorname"] . "<br>\n";
    echo "Nachname: " . $_POST["nachname"] . "<br>\n";
    echo "Wohnort: " . $_POST["ort"] . "<br>\n";
?>
```

## Formular und Auswertung in einer Datei

Das Formular und die Auswertung können in einer Datei zusammengefasst werden. Mit einer if-Abfrage kann überprüft werden, ob der Submit-Button gedrückt wurde oder nicht.

## pizzabestellung.php

[pizzabestellung.php](#)

```
Pizzabestellung<br>
Pizza Margarita zum günstigen Preis von 5.50 Euro bestellen! <br>
<pre>
<form action="pizzabestellung.php" method="POST">
    Name: <input type="Text" name="name" />
    Lieferadresse: <input type="Text" name="adresse" />
    Anzahl der Pizzen: <input type="Text" name="anzahl" /><br><br>
    <input type="Submit" name="schicken" value="Bestellung
abschicken"><br>
    <input type="reset" value="Zurücksetzen">
</form>
</pre>
<hr>
<?php
    if ($_POST["schicken"]=="Bestellung abschicken")
    {
        echo "<i>Für folgende Bestellung wird gedankt: </i><br><br>";
        echo "<b>Name:</b> " . $_POST["name"] . "<br>";
        echo "<b>Lieferadresse:</b> " . $_POST["adresse"] . "<br>";
        if ($_POST["anzahl"]<>0)
```

```
{
    echo "Menge beträgt <b>".$_POST["anzahl"]." Stück</b> zum Preis von
    <b>".$_POST["anzahl"]*5.50." Euro</b>";
}
}

?>
```

## Überprüfen des mitgeschickten Arrays mittels POST

```
print_r($_POST);
var_dump($_POST);
```

`print_r()` Zeigt den Inhalt der Variablen in lesbarer Form an. `var_dump()` geht einen Schritt weiter, es können auch, durch Beistriche getrennt, mehrere Variablen angegeben werden. Die Informationen sind detaillierter. Sie enthalten auch Typ und Länge der Variablen.  
`var_dump(Variable 1[, Variable 2,...]);` Das gesamte mittels POST übermittelte Array wird ausgegeben.

### Beispiel Newsletter

- [Beispiel Newsletter](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_07](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_07)

Last update: **2018/09/11 14:13**



## Aufgabe 2

Erstelle eine kleine Website für eine Apfelbestellung:

- bioladen.html
  - Eingabe von Name, Adresse, Menge (in kg) 1. Sorte Golden Delicious (1,20 pro kg), 2. Sorte: Idared (1,50 pro kg)
- bioladen-auswertung.php
  - Ausgabe von Name, Adresse, bestellter Menge und Gesamtpreis
  - ab insg. 10 kg bestellter Menge: keine Versandkosten, sonst 5,00 Euro.

## Lösung

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_08](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_08)

Last update: **2018/09/11 14:17**



# Anführungszeichen

## Doppelte oder einfache Anführungszeichen

```
$foo = 'Heute';

echo "$foo ist ein schöner Tag!"; // Ausgabe: Heute ist ein schöner Tag!
echo '$foo ist ein schöner Tag!'; // Ausgabe: $foo ist ein schöner Tag!
```

Wenn ein PHP-Programmierer einen Link aus Variablen erstellt, kann man häufig folgenden Code sehen:

```
echo '<a href="' . $link . '" id="' . $id . '" class="' . $class . '">' .
$linktext . '</a>';
```

oder

```
?>
<a href="<?php echo $link; ?>" id="<?php echo $id; ?>"
  class="<?php echo $class; ?>"<?php echo $linktext; ?>"</a>
<?php
```

oder

```
echo "<a href=\"\$link\" id=\"\$id\" class=\"\$class\">\$linktext</a>";
```

Abgesehen vom persönlichen Stil des Programmierers haben die Code-Beispiele eines gemeinsam: sie sind schlecht zu lesen. „Klar!“, werden jetzt die Programmierer sagen, „In HTML müssen die Attribute in doppelten Anführungszeichen stehen und in PHP muss man eine der oben gezeigten Methoden verwenden“. Sicher? Müssen Attribute wirklich in doppelten Anführungszeichen stehen? Die einfache Antwort ist: Nein! Ein Blick in die Spezifikation bringt Folgendes an den Tag:

[...]Standardmäßig verlangt SGML, dass alle Attributwerte entweder von doppelten Anführungszeichen (ASCII dezimal 34) oder einfachen Anführungszeichen (ASCII dezimal 39) begrenzt werden. Einfache Anführungszeichen können im Attributwert enthalten sein, wenn der Wert durch doppelte Anführungszeichen begrenzt ist und umgekehrt.[...]

Doppelte oder einfache Anführungszeichen. Das macht die Sache doch viel übersichtlicher:

```
echo "<a href='$link' id='$id' class='$class'>\$linktext</a>";
```

```
echo 'Micro$oft'; // ergibt: Micro$oft
echo "Micro$oft"; // ergibt: Micro + Inhalt der Variable $oft
echo "Micro\$oft"; // ergibt: Micro$oft

echo 'c:\temp'; // ergibt: c:\temp
```

```
echo "c:\temp"; // ergibt: c: + Tabulator + emp
echo "c:\\temp"; // ergibt: c:\temp
echo 'c:\\temp'; // ergibt: c:\temp

echo 'Kein Hochkomma: \x27'; // ergibt: Kein Hochkomma: \x27
echo " Ein Hochkomma: \x27"; // ergibt: Ein Hochkomma: '
echo ' Ein Hochkomma: \''; // ergibt: Ein Hochkomma: '

echo "<input name='foo' value='$bar'>"; // gültiges HTML
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_09](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_09)



Last update: **2018/09/11 14:17**

# Vergleichsoperatoren

Eine Bedingung ist eine Möglichkeit, den Ablauf eines Skripts durch Entscheidungen zu beeinflussen. In einer Bedingung werden Ausdrücke verglichen. Das Ergebnis kann entweder „Ja“ (TRUE) oder „Nein“ (FALSE) sein.

Operator	Name	Bedeutung
==	Gleichheit	\$a == \$b ergibt TRUE, wenn \$a und \$b gleich sind.
===	Identisch	\$a === \$b ergibt TRUE, wenn \$a und \$b gleich und vom selben Datentyp sind.
!=	Ungleichheit	\$a != \$b ergibt TRUE, wenn \$a und \$b ungleich sind.
<	Kleiner	\$a < \$b ergibt TRUE, wenn \$a kleiner \$b ist.
>	Größer	\$a > \$b ergibt TRUE, wenn \$a größer \$b ist.
<=	Kleiner gleich	\$a <= \$b ergibt TRUE, wenn \$a kleiner oder gleich \$b ist.
>=	Größer gleich	\$a >= \$b ergibt TRUE, wenn \$a größer oder gleich \$b ist.

## Verknüpfung von Bedingungen

Operator	Name	Bedeutung
and bzw. &&	UND	\$a and \$b ergibt 1, wenn \$a und \$b 1 sind, ansonsten wird 0 zurückgegeben
or bzw.	ODER	\$a or \$b ergibt 1, wenn mindestens eine der beiden Variablen 1 ist.
xor	ausschließendes ODER	\$a xor \$b ergibt 1, wenn entweder \$a oder \$b 1 sind, aber nie beide gleichzeitig
!	NICHT	!\$a ergibt die Umkehrung des Wahrheitswertes

From:  
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_10](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_10)

Last update: **2018/09/11 14:18**



# Die einfache if-Anweisung

## Syntax

```
if (Bedingung)
{
    Anweisungen;
}
```

- Die Bedingung steht in runden Klammern.
- Die Anweisungen stehen in geschwungenen Klammern.
- Liefert die Bedingung TRUE zurück, werden die Anweisungen ausgeführt, ist die Bedingung FALSE, werden die Anweisungen ignoriert
- Wird nur eine Anweisung ausgeführt, können die geschwungenen Klammern entfallen.

## Beispiel: if1.php

```
<?php
$menge = 7;
if ($menge > 5)
    echo "Sie haben mehr als 5 Kilo bestellt.";
?>
```

- Der Variable \$menge wird der Wert 7 zugewiesen. Da \$menge größer als 5 ist, wird die Anweisung ausgeführt.

## Beispiel: if2.php

```
<?php
$menge = 7;
if ($menge > 5)
{
    echo "Sie haben mehr als 5 Kilo bestellt.<br>\n";
    echo "Der Versand ist deswegen kostenfrei.";
}
?>
```

# Die if-Anweisung mit else Zweig

## Syntax

```
if (Bedingung)
{
    Anweisungsblock1;
}
```

```
else
{
    Anweisungsblock2;
}
```

- Falls die Bedingung erfüllt ist, wird Anweisungsblock1 ausgeführt, sonst Anweisungsblock2.

### Beispiel: ifelse.php

```
<?php
$menge=4;
if($menge > 5)
{
    echo "Sie haben mehr als 5 Kilo bestellt.<br>\n";
    echo "Der Versand ist deswegen kostenfrei.";
}
else
{
    echo "Sie haben $menge Kilo bestellt.<br>\n";
    echo "Der Versand kostet pauschal 7,00 EUR";
}

?>
```

## Verschachtelte if-Anweisungen

### Syntax

```
if (Bedingung1)
{
    Anweisungsblock1;
    if (Bedingung2)
    {
        Anweisungsblock2;
    }
    else
    {
        Anweisungsblock3;
    }
}
else
{
    Anweisungsblock4;
}
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_11](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_11)



Last update: **2018/09/11 14:18**

# Fallauswahl mit switch-Anweisung

Wenn eine Variable mit verschiedenen Werten verglichen werden soll, kann die Programmierung mit einer if-Anweisung sehr aufwändig sein. Als Alternative dazu kann man in diesem Fall die switch-Anweisung einsetzen:

## Syntax

```
switch ($variable)
{
    case Wert1:
        Anweisungsblock1;
        break;
    case Wert2:
        Anweisungsblock2;
        break;
    default:
        Anweisungsblock3;
}
```

- Stimmt der Wert der Variable mit einem der angeführten Auswahlwerte überein, dann wird der Anweisungsblock unmittelbar bis zur Anweisung break ausgeführt. Die restlichen Blöcke werden nicht ausgeführt.
- Ist kein break am Ende eines case-Teils angegeben, so werden auch alle nachfolgenden case-Blöcke ausgeführt, bis eine break-Anweisung erfolgt.
- Stimmt der Wert der Variablen mit keinem der angegebenen Werte überein, wird der Anweisungsblock nach der default-Anweisung durchgeführt.

## Bsp: switch-case.php

```
<?php
$sorte = "Gala";
echo "Der Preis für 1kg $sorte beträgt <br>\n";
switch($sorte) //Preiszuordnung abhängig von Sorte
{
    case "Jonagold":
        echo "1,50 EUR.";
        break;
    case "Delicious":
        echo "1,60 EUR.";
        break;
    case "Gala":
        echo "1,65 EUR.";
        break;
    case "Elstar":
        echo "2,00 EUR.";
        break;
}
```

```
default:  
    echo "Diese Sorte haben wir leider nicht im Angebot."  
?>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_12](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_12)

Last update: **2018/09/11 14:18**



## Aufgabe 3

### Apfelbestellung - Versandkosten

Erweitere die [Aufgabe 2](#) folgendermaßen:

- Wenn mehr als 10 kg Äpfel bestellt werden, ist der Versand kostenfrei, ansonsten werden 7.00 Euro verrechnet.
- Ausgabe:
  - Rückmeldung, ob Versandkosten zu bezahlen sind oder nicht.
  - Gesamtpreis.

## Aufgabe 4

### Minimum - Maximum

- In einem Formular (zahlen\_eingabe.html) sollen drei Zahlen eingegeben werden (mittels drei Input-Feldern)
- Als Rückmeldung (minmax\_ausgabe.php) wird das Maximum sowie das Minimum der drei Zahlen ausgegeben.

## Aufgabe 5

### Kennwortabfrage mit if

Viele Webseiten arbeiten mit einer Anmeldung des Benutzers. Um Zugang zu weiteren Teilen der Webseite zu bekommen, muss sich der Benutzer einem „Loginverfahren“ unterziehen.

- Erstelle hierfür ein Formular (kennwort.html), in welchem der Benutzer ein Kennwort eingeben kann. Verwende dazu ein `<input>`-Feld, bei dem das Kennwort bei der Eingabe nicht sichtbar ist.
- Nach Klick auf den Button „Abschicken“ soll eine Rückmeldung (kennwort\_auswertung.php) erfolgen, ob das Passwort richtig eingegeben wurde.
- Baue das Formular so aus, dass vor dem Passwort auch der Nickname eingegeben werden muss. Eine positive Rückmeldung (z.B. „Zutritt erfolgreich...“) wird nur dann gegeben, wenn Nickname **und** Passwort korrekt eingegeben wurden.

## Aufgabe 6

### Verschachtelte if-Anweisungen

Baue die Aufgabe 3 folgendermaßen aus:

- Wenn mehr als 20 kg Äpfel bestellt werden, gibt es zur Bestellung zusätzlich ein kleines Präsent.
- Ausgabe:
  - Rückmeldung, ob Versandkosten zu bezahlen sind oder nicht und ob es ein Präsent gibt.
  - Gesamtpreis.

## Aufgabe 7

### switch-Anweisung

Erweitere die Aufgabe 5 folgendermaßen aus:

- Bei der Bestellung soll die bestellte Apfelart in einem `<input>`-Feld eingegeben werden.

Apfelsorte	Preis
Jonagold	1,50 EUR
Delicious	1,60 EUR
Gala	1,65 EUR
Elstar	2,00 EUR

- Ausgabe
  - Welche Äpfelsorte bestellt wurde (inkl. Preisangabe), bei falscher Eingabe soll eine entsprechende Meldung erfolgen und kein Gesamtpreis berechnet werden.
  - Gesamtpreis je nach bestellter Sorte.

From:  
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:  
[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_13](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_13)

Last update: **2018/09/11 14:18**



## for-Schleife

In der for-Schleife wird genau angegeben, wie oft die Schleife durchlaufen werden soll.

### Syntax

```
for (Initialisierung; Bedingung; Operation)
{
    Anweisungsblock;
}
```

### Beispiel: Ausgabe der ersten n ungeraden Zahlen und die Summe dieser Folge

```
<?php
    echo "<h2>Ausgabe der ersten n ungeraden Zahlen und die Summe dieser
Folge</h2>\n";
    $n=10;
    $summe=0;
    for ($i=1;$i<=$n;$i++)
    {
        $ung=$i*2-1;
        $summe=$summe+$ung;
        echo $i.". ungerade Zahl: ".$ung."<br>\n";
    }
    echo "Die Summe der ersten $n ungeraden Zahlen beträgt: ".$summe;
?>
```

### Beispiele für Operatoren in for-Schleifen

Bedingung in der for-Schleife	Werte von \$i
for (\$i=1; \$i<=5, \$i++)	1, 2, 3, 4, 5
for (\$i=1; \$i<5, \$i++)	1, 2, 3, 4
for (\$i=15; \$i>=10, \$i--)	15, 14, 13, 12, 11, 10
for (\$i=15; \$i>10, \$i--)	15, 14, 13, 12, 11
for (\$i=0; \$i<=100, \$i=\$i+10)	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
for (\$i=1; \$i<=12, \$i=\$i+1.2)	1, 2.4, 3.6, 4.8, 7.2, 8.4, 9.6, 10.8, 12

## Geschachtelte for-Schleifen

Mit geschachtelten for-Schleifen können zweidimensionale Strukturen abgearbeitet werden.

### Beispiel: Erstellen einer Tabelle mit 3 Zeilen und 4 Spalten mittels zweier for-Schleifen

```
<?php
    echo "<table border='1'>\n";
    for ($i=1;$i<=3;$i++)
    {
        echo "<tr>\n";
        for ($j=1;$j<=4;$j++)
        {
            echo "<td> $i. Zeile, $j. Spalte </td>\n";
        }
        echo "</tr>\n";
    }
    echo "</table>\n";
?>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_14](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_14)



Last update: **2018/09/11 14:21**

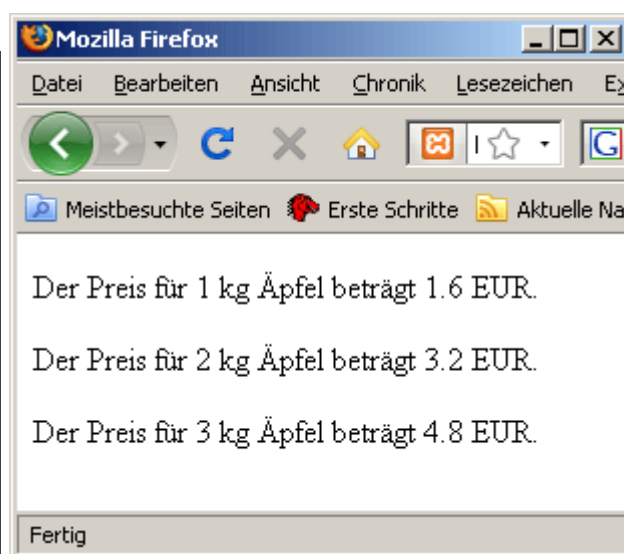
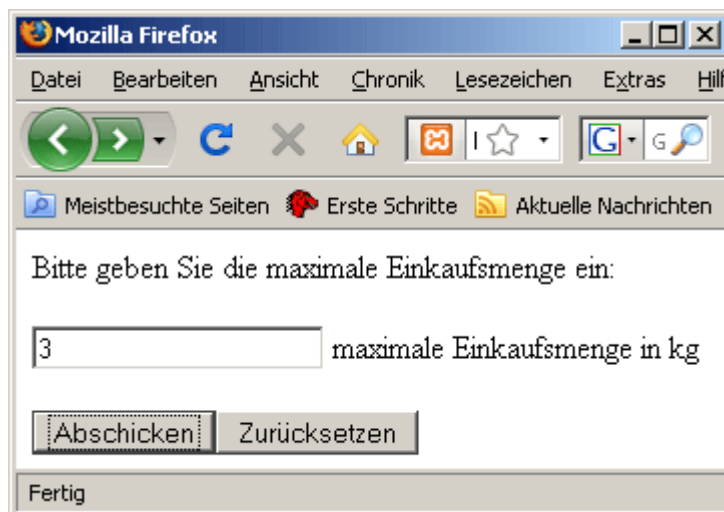
# while-Schleife

## Syntax

```
while(Bedingung)
{
    Anweisungsblock;
}
```

## Bsp: while.php

```
<?php
$menge = 1;
$preis = 1.6;
while ($menge <= $_POST["maxmenge"])
{
    $gesamtpreis = $preis * $menge;
    echo "<p>Der Preis für $menge kg Äpfel beträgt $gesamtpreis EUR.</p>\n";
    $menge++;
}
?>
```



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_15](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_15)

Last update: **2018/09/11 14:19**



# do-while-Schleife

Die do-while Schleife unterscheidet sich von der while-Schleife, dadurch, dass die Abbruchbedingung nicht am Anfang, sondern am Ende jedes Schleifendurchlaufs geprüft wird. Die Anweisung wird somit zumindest einmal durchlaufen.

## Syntax

```
do
{
    Anweisungsblock;
}
while (Bedingung);
```

## Bsp: dowhile.php

```
<?php
    $i=1;
    do
    {
        echo "<br>".$i*$i;
        $i++;
    }
    while ($i<=10);
?>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_16](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_16)



Last update: **2018/09/11 14:23**

## Aufgabe 8

Erstelle ein PHP-Skript, welches eine Folge von n Zahlen erzeugt, in der jede Zahl den doppelten Wert der vorhergehenden hat.

## Aufgabe 9

Erstelle ein PHP-Skript, das das „Ein-mal-eins“ für die Faktoren zwischen 11 und 20 ausgibt („großes Ein-mal-eins“). Die einzelnen Rechnungen sollen dabei in einer Tabelle aufscheinen.

## Aufgabe 10

Schreibe ein PHP-Skript, das die Fibonacci-Zahlen bis zu einer eingegebenen Höchstgrenze erzeugt und ausgibt.

Eine Fibonacci-Zahl wird als Summe der beiden vorhergehenden Fibonacci-Zahlen gebildet. Die erste und die zweite Fibonacci-Zahl sind gleich 1.

Beispiel: 1, 1, 2, 3, 5, 8, 13, ...

## Aufgabe 11

### Teiler einer Zahl

Erstelle ein HTML- und PHP-Skript, welches die Teiler von Zahlen ermittelt. Weiters soll zusätzlich bestimmt werden, ob es sich um eine Primzahl (Anzahl der Teiler: 2) handelt

a) Version 1: Es werden die Teiler der eingegebenen Zahl ermittelt:

Ausgabe z.B.: 27 hat die Teiler 1, 3, 9, 27

b) Version 2: Es werden die Teiler sämtlicher Zahlen bis zur eingegebenen Zahl ermittelt:

Ausgabe z.B.:

...

23 hat die Teiler 1, 23 (Primzahl)

24 hat die Teiler 1, 2, 3, 4, 6, 8, 12, 24

25 hat die Teiler 1, 5, 25

26 hat die Teiler 1, 2, 13, 26

27 hat die Teiler 1, 3, 9, 27

teiler.php

```
<meta name="author" content="user">
</head>
<body>
<h1>Teiler</h1>
berechnet Teiler einer geg. Zahl<br> <br>
<form action="Teiler.php" method="POST" target="">
Untergrenze:<input type="Text" name="untergrenze" value="" size=""
maxlength=""><br>
Obergrenze:<input type="Text" name="zahl" value="" size="6"
maxlength=""> <input type="Submit" name="send"
value="Berechnen"></form>
<?php
if (isset($_POST["send"])){
    $zahl=$_POST["zahl"];
    $ug=$_POST["untergrenze"]; //Startwert
    for($n=$ug;$n<=$zahl;$n++){
        echo "Teiler von ".$n."&nbsp;&nbsp;&nbsp;";
        echo "1";
        $teilerzahl=2;
        for($i=2;$i<=($n/2);$i++){
            if($n%$i==0){
                echo ", ".$i;
                $teilerzahl++;
            }
        }
        if($n!=1) echo ", ".$n;
        if($teilerzahl==2) echo " (Primzahl)";
        echo "<br>";
    }
}
?>
</body>
</html>
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

[http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi\\_201819:1:1\\_17](http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819:1:1_17)

Last update: **2018/09/11 14:24**

